AfIA

Association française
pour l'Intelligence Artificielle

RJCIA

24es Rencontres des Jeunes Chercheurs en Intelligence Artificielle

PFIA 2026

Table des matières

Éditorial

24es Rencontres des Jeunes Chercheurs en Intelligence Artificielle

Éditorial

Rencontres des Jeunes Chercheurs en Intelligence Artificielle

Les Rencontres des Jeunes Chercheurs en Intelligence Artificielle (RJCIA) sont destinées aux jeunes chercheurs en IA, doctorants ou titulaires d'un doctorat depuis moins de deux ans. À ce titre, l'objectif de cette manifestation est double :

1. permettre aux jeunes chercheurs préparant une thèse en Intelligence Artificielle, ou l'ayant soutenue depuis peu, de se rencontrer et de présenter leurs travaux, et d'ainsi nouer des contacts avec d'autres jeunes chercheurs et d'élargir leurs perspectives en échangeant avec des spécialistes d'autres domaines de l'IA ;
2. former les jeunes chercheurs à la préparation d'un article, à sa révision pour tenir compte des observations du comité de programme, et à sa présentation devant un auditoire de spécialistes, leur permettant ainsi d'obtenir des retours de chercheurs de leur domaine ou de domaines connexes.

Toute contribution relevant de l'Intelligence Artificielle est la bienvenue.

Portées par la Plate-Forme Intelligence Artificielle (PFIA), les RJCIA en sont à leur **24e édition**, organisée à Arras du 2 au 3 juillet 2026, au sein de la PFIA qui se déroule du 29 juin au 3 juillet 2026 sur le site du CRIL. Cette année, **22 articles ont été soumis**. Chaque article a été relu par trois membres du comité de programme. Les auteurs des articles publiés dans les présents actes ont eu à prendre en compte les remarques qui leur ont été adressées dans le cadre de cette relecture.

Les **20 articles acceptés** — 14 articles longs et 6 articles courts — ont été répartis en **cinq sessions thématiques** couvrant un large spectre de l'IA contemporaine : modèles de langage et raisonnement, apprentissage par renforcement et systèmes multi-agents, santé et environnement, perception et vision, détection d'anomalies et quantification de l'incertitude. La diversité des travaux présentés, qu'ils s'intéressent aux grands modèles de langage, à la fusion de capteurs, à la planification ou encore à l'audit de systèmes en boîte noire, témoigne de la vitalité et de l'étendue des recherches menées par la nouvelle génération de chercheurs en IA.

Le programme a été enrichi par une **conférence invitée** de **Solenne Gaucher**, Assistant Professor au département de mathématiques appliquées de l'École polytechnique et chercheuse au CREST (ENSAE). Ses travaux portent sur l'apprentissage séquentiel, l'analyse de réseaux et l'équité algorithmique ; elle s'attache en particulier à concevoir des modèles garantissant la neutralité des décisions prises par les algorithmes. Lauréate du Prix Jeunes Talents L'Oréal-UNESCO « Pour les Femmes et la Science » en 2024, elle apporte à cette édition un éclairage précieux sur les enjeux d'une intelligence artificielle plus juste et inclusive.

Deux interventions supplémentaires ont été ajoutées pour enrichir le programme : celles de **Alain Berger**, Directeur Général de Ardans, intitulée *"Ausculter, explorer, qualifier et exploiter une base de connaissance industrielle"*, et celle de **Mickaël Audegond**, Vice- Président de la Communauté Urbaine d'Arras en charge du Numérique.

Cette année, grâce à l'AFIA, un **Prix du Meilleur Article** sera décerné par le comité de programme à l'issue de la conférence, récompensant la qualité scientifique et la clarté de présentation d'un travail.

Je remercie chaleureusement les **membres du comité de programme** pour leur investissement dans l'évaluation des soumissions et leurs retours constructifs aux auteurs, ainsi que l'ensemble des **intervenantes et intervenants** pour leurs précieuses contributions. Mes remerciements vont également au **comité d'organisation de la PFIA 2026** pour l'accueil de cette édition à Arras.

Nicolas Verstaevel
IRIT, Université Toulouse Capitole
Président du comité de programme RJCIA 2026

Nicolas Verstaevel

Comité de programme

Présidence

— Nicolas Verstaevel (IRIT, Université Toulouse Capitole, France).

Membres

— Lylia Abrouk (LIB, Université de Bourgogne, France) ;
— Frédéric Amblard (IRIT, Université Toulouse Capitole, France) ;
— Sandra Bringay (LIRMM, Université Paul-Valéry Montpellier 3, CNRS, France) ;
— Francesca Bugiotti (LISN, CentraleSupélec, Université Paris-Saclay, France) ;
— Jean-Pierre George (IRIT, Université de Toulouse, France) ;
— Davide Guastella (LIS, Université d'Aix-Marseille, France) ;
— Alexis Guyot (LIS, Université d'Aix-Marseille, France) ;
— Nathalie Hernandez (IRIT, Université de Toulouse 2 Jean Jaurès, France) ;
— Liliana Ibanescu (MIA-Paris-Saclay, AgroParisTech, Université Paris-Saclay, INRAE, France) ;
— Pierre Larmande (UMR DIADE, IRD, France) ;
— Mélanie Munch (INRAE, France) ;
— Pierre-Henri Paris (LISN, Université Paris-Saclay, France) ;
— Nathalie Pernelle (LIPN, Université Sorbonne Paris Nord, France) ;
— Manon Prédhumeau (IRIT, Université Toulouse Capitole, France) ;
— Paul Saves (IRIT, Université Toulouse Capitole, France) ;
— Danai Symeonidou (MISTEA, INRAE, France) ;
— Cassia Trojahn (IRIT, Université de Toulouse 2 Jean Jaurès, France) ;
— Nadia Yacoubi Ayadi (LIRIS, Université Claude Bernard Lyon 1, France).

Session 1 : LLM, NLP et Raisonnement

Accelerating Large Language Model Inference with Dynamic Self-Speculative Decoding

Florian Valade[1]

[1] Université Gustave Eiffel, Fujitsu

florian.valade@email.com

Résumé

Cet article présente une approche modulaire pour accélérer l'inférence des modèles de langage de grande taille (LLMs) en ajoutant des têtes de sortie anticipée (« early exit heads ») aux couches intermédiaires du transformer. Chaque tête est entraînée de manière auto-supervisée pour imiter les prédictions du modèle principal, permettant ainsi d'interrompre le calcul dès qu'un seuil de confiance calibré est atteint. Nous évaluons plusieurs métriques de confiance et démontrons que l'entropie offre la séparation la plus fiable entre les prédictions correctes et incorrectes. Des expériences menées sur la suite de modèles Pythia (de 70M à 2,8B de paramètres) montrent que notre méthode réduit considérablement le coût d'inférence tout en maintenant la précision sur plusieurs benchmarks. Nous adaptons ensuite cette approche au décodage spéculatif en introduisant le « Dynamic Self-Speculative Decoding » (DSSD), qui atteint un taux d'acceptation de jetons 1,66× supérieur aux références LayerSkip ajustées manuellement, avec un réglage minimal des hyperparamètres.

Mots-clés

Early Exit, LLM Inference, Speculative Decoding, Acceleration

Abstract

This paper presents a modular approach to accelerate inference in large language models (LLMs) by adding early exit heads at intermediate transformer layers. Each head is trained in a self-supervised manner to mimic the main model's predictions, allowing computation to stop early when a calibrated confidence threshold is reached. We evaluate several confidence metrics and show that entropy provides the most reliable separation between correct and incorrect predictions. Experiments on the Pythia model suite (70M to 2.8B parameters) demonstrate that our method significantly reduces inference cost while maintaining accuracy across multiple benchmarks. We further adapt this approach to speculative decoding, introducing Dynamic Self-Speculative Decoding (DSSD), which achieves 1.66× higher token acceptance than manually-tuned LayerSkip baselines with minimal hyperparameter tuning.

Keywords

Early Exit, LLM Inference, Speculative Decoding, Acceleration

1 Introduction

Large language models (LLMs) have become central to advancing capabilities in natural language processing (NLP), delivering remarkable performance across a range of tasks. The trend towards scaling up these models correlates strongly with improved performance, understanding, and generality. This relationship has been formalized through empirical scaling laws, which demonstrate that model performance improves predictably with increased model size, dataset size, and compute budget [16, 13]. However, the computational cost associated with these larger models is substantial, often necessitating the use of powerful server infrastructure [22]. This not only limits local usability but also raises significant privacy concerns and requires considerable investment to scale in response to user demand. Solutions exist to reduce the computational demands of these models, but they often impact the model's performance by reducing its accuracy [41].

Despite their effectiveness, these models often operate inefficiently. The nature of language itself contributes to this inefficiency ; namely, not all tokens generated during the inference process contribute equally to the overall meaning or require the same level of computational resources. Some tokens are inherently simpler and can be predicted with high confidence early in the computation process, while others, contributing more significantly to the context or meaning, may require deeper processing.

In response to these challenges, we develop a method that can be easily integrated into existing pre-trained models to enhance their inference speed without extensive retraining. Our solution focuses on the strategic placement of early exit "heads" [23, 28] within the transformer layers of an LLM. These heads terminate the inference process when a calibrated confidence threshold is met, based on the complexity and predictability of the token being processed.

Our contributions are twofold :

1. We provide a detailed experimental study and modular framework for training and deploying early exit heads on top of LLMs. We analyze multiple training strategies, confidence metrics, and demonstrate scalability across model sizes from 70M to 2.8B parameters on the Pythia suite.

2. We adapt our early exit mechanism to speculative decoding, introducing **Dynamic Self-Speculative Decoding (DSSD)**, which achieves **1.66× higher token acceptance rates** than manually-tuned LayerSkip baselines while requiring minimal hyperparameter tuning—only a single accuracy threshold ϵ.

2 Related Work

Transformers [32] scaled into today's LLM families such as BERT, GPT-3, PaLM, LaMDA, LLaMA and OPT [7, 3, 6, 29, 30, 39], powering vision [8], speech [21] and multimodal models. Their billion-parameter footprints, however, make every token generation costly. Static compression—quantisation, pruning and distillation [25, 27, 38, 11, 26, 1]—slashes model size but still expends identical compute on easy and hard inputs. Early-exit methods address this imbalance by attaching lightweight classifiers to intermediate layers and halting computation once a confidence criterion is met.

In computer vision, BranchyNet [28] and MSDNet [14] or EERO methodology [31] established the two key components still used today : deeply supervised branches and an entropy-based exit rule. Transferring the idea to Transformers, DeeBERT and *The Right Tool* calibrated softmax confidence to save up to 5× latency with negligible loss [35, 24]. FastBERT distilled the final head into earlier exits [20], PABEE demanded k consecutive agreeing predictions instead of a threshold [40], and BERxiT learned an explicit when-to-exit module while alternating fine-tune schedules [36]. Skip/SmartBERT added trainable gates that may bypass whole layers, combining skipping with exiting for 2–4× cheaper inference [18, 5].

Early exit for sequence-to-sequence generation is newer. Depth-Adaptive Transformers first applied exits to neural MT [9] ; FREE and DEED refined token-level uncertainty estimates [34, 37]. Recent evidence shows exits remain effective inside 13 B-parameter LLaMA-2 and GPT-J [19], suggesting scalability to modern LLMs.

Speculative decoding [17, 4] accelerates LLM inference by using a smaller draft model to generate candidate tokens, which are then verified in parallel by the target model. LayerSkip [10] combines this with early exits, using intermediate layers as draft models. However, LayerSkip requires exhaustive hyperparameter search over both head selection and speculation length. Our DSSD method addresses this limitation by adaptively selecting exit layers based on calibrated confidence.

Our work inherits the plug-and-play nature of DeeBERT-style branches, borrows the self-supervised signal of FastBERT, and scales token-level exits to different architecture sizes. We train exit heads without extra data—using the model's own probabilities—then calibrate a single threshold on a held-out set à la conformal prediction [33]. Unlike layer-skipping approaches, the backbone weights remain frozen, guaranteeing monotonic accuracy with deeper computation.

3 Methodology

This section details our methodology for integrating and utilizing early exits within large language models (LLMs) to enhance computational efficiency during inference. The approach is designed to be generalizable and, while we demonstrate its application using the Pythia suite, it is applicable to any multi-layered transformer model. This adaptability ensures that our methodology can be leveraged across a broad spectrum of modern LLMs, enhancing their usability without requiring significant modifications to their underlying architectures.

3.1 Definitions and Notation

We introduce here the main notations used throughout the paper.

Vocabulary. Let $\mathcal{V}$ denote the vocabulary, a finite set of tokens :

$$\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\},$$

where $|\cdot|$ denotes cardinality of sets.

Dataset. We consider a dataset $\mathcal{D}$ of N examples, where each example is a sequence of tokens :

$$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}, \quad \mathbf{x}^{(i)} = (x_1^{(i)}, \dots, x_{L_i}^{(i)}), \quad x_j^{(i)} \in \mathcal{V}.$$

A subset $\mathcal{D}_{\text{cal}} \subset \mathcal{D}$ is reserved as a calibration set, the rest being used for training.

Each sequence $\mathbf{x}^{(i)}$ can have a variable length L_i. The total number of tokens in the dataset is $M = \sum_{i=1}^{N} L_i$, with $M > N$ in general. Similarly, the calibration set $\mathcal{D}_{\text{cal}}$ contains N_{cal} examples and a total of M_{cal} tokens.

Language model. A large language model (LLM) is a function f_θ parameterized by θ, mapping a sequence of tokens to a sequence of probability distributions over the vocabulary :

$$f_\theta : (x_1, \dots, x_L) \mapsto (\mathbf{p}_1, \dots, \mathbf{p}_L), \quad \mathbf{p}_t \in \Delta^{|\mathcal{V}|},$$

where $\Delta^{|\mathcal{V}|}$ is the probability simplex over $\mathcal{V}$.

Objective. Given an input sequence $\mathbf{x} = (x_1, \dots, x_L) \in \mathcal{V}^L$, the target sequence is $\mathbf{y} = (y_1, \dots, y_L) \in \mathcal{V}^L$ with $y_t = x_{t+1}$ (i.e., the next-token prediction task).

Early exit heads. We introduce K early exit heads, each defined as a function h_k (with its own parameters, but sharing the backbone with f_θ) that maps a sequence of L tokens to a sequence of probability distributions :

$$h_k : (x_1, \dots, x_L) \mapsto (\mathbf{p}_{k,1}, \dots, \mathbf{p}_{k,L}), \quad \mathbf{p}_{k,t} \in \Delta^{|\mathcal{V}|}.$$

All early exit heads share the same backbone, so most of their parameters are shared with the main model.

Confidence metric. A confidence metric is a function $c : \Delta^{|\mathcal{V}|} \to \mathbb{R}$ that assigns a real-valued confidence score to a probability vector (e.g., $c(\mathbf{p}) = \max_j p_j$).

Accuracy threshold ϵ. The parameter $\epsilon \in [0, 1]$ controls the minimal desired accuracy for early exit decisions. For each early exit head, a prediction is only output if its confidence metric exceeds a calibrated threshold corresponding to at least ϵ empirical accuracy on a held-out calibration set. Lowering ϵ increases speedup at the cost of potential accuracy loss, while higher ϵ enforces stricter correctness guarantees.

The above definitions give a precise, token-level view of the model outputs and their interaction with early-exit logic ; they will serve as the foundation for the training objectives, calibration techniques, and inference algorithms described in the following sections.

3.2 Implementation Details

To enhance the inference efficiency of large language models, we incorporate early exit "heads" into a pre-existing model, in this instance, the Pythia suite [2]. These heads are implemented at regular intervals along the network. Structurally, each head is a simple multi-layer perceptron (MLP) identical to the final classification head of the model. Each of these head takes as input hidden features from a transformer block inside the model.

Placement of early–exit heads. Let the transformer backbone consist of L stacked blocks, indexed $\mathcal{I} = \{1, 2, \ldots, L\}$, and let K denote the desired number of early–exit heads. To distribute the heads uniformly while keeping the final classification layer untouched, we attach head $k \in \{1, \ldots, K\}$ to the block whose index is

$$\ell_k = \left\lfloor \frac{k}{K+1} L \right\rfloor, \qquad k = 1, \ldots, K. \tag{1}$$

Equation (1) simply divides the layer indices into $K + 1$ equal segments and selects the endpoint of each segment (rounded down to the nearest integer) as a branch location. In our experiments we set $K = 4$; hence the heads are inserted at $\ell_k \in \left\{ \lfloor \frac{L}{5} \rfloor, \lfloor \frac{2L}{5} \rfloor, \lfloor \frac{3L}{5} \rfloor, \lfloor \frac{4L}{5} \rfloor \right\}$.

For the implementation of these heads, we experimented with two initialization strategies : initializing the heads from scratch and copying the final classification head in order to fine-tune it. The difference between the two are analyzed in Section 4. With the architectural choices established, we next describe the data and objectives used to train the early exit heads.

3.3 Training data and objectives

Corpus. All auxiliary heads are trained on MINIPILE [15], a 6-GB stratified subset of the original 825-GB THEPILE-DEDUPLICATED [12] dataset that was employed for pretraining the PYTHIA backbone [2]. Using the same data distribution avoids the distribution-shift issues that often appear when auxiliary classifiers are fitted post-hoc.

Losses. For the whole predicted sequence $\hat{\mathbf{y}}$, we can compute different losses depending on the objectives. We define the cross-entropy loss and the Kullback–Leibler divergence between the predicted probability distribution $\hat{y}$ and the ground-truth labels y as follows :

$$\mathcal{L}_{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{L} \sum_{t=1}^{L} \sum_{v \in \mathcal{V}} \mathbf{1}_{\{\mathbf{y}^{(t)} = v\}} \log(\hat{\mathbf{y}}_v^{(t)}),$$

$$\mathcal{L}_{\text{KL}}(\hat{\mathbf{y}} || \mathbf{y}) = \frac{1}{L} \sum_{t=1}^{L} \sum_{v \in \mathcal{V}} \hat{\mathbf{y}}_v \log \frac{\hat{\mathbf{y}}_v}{\mathbf{y}_v}.$$

In the context of self-supervised training, we consider the Kullback-Leibler divergence where the target $\mathbf{y}$ is replaced by the output of the main model f_θ. Thus, the loss $\mathcal{L}_{\text{KL}}$ becomes :

$$\mathcal{L}_{\text{KL}}(\hat{\mathbf{y}} || f_\theta) = \frac{1}{L} \sum_{t=1}^{L} \sum_{v \in \mathcal{V}} \hat{\mathbf{y}}_v \log \frac{\hat{\mathbf{y}}_v}{f_\theta(v)}$$

The cross-entropy loss $\mathcal{L}_{\text{CE}}$ is used when considering a supervised training objective. It matches how the main model is trained and can be used to train the heads as well. The Kullback–Leibler divergence $\mathcal{L}_{\text{KL}}$ is used when considering a self-supervised training objective. It encourages the early exit heads to mimic the full probability distribution of the main model, which can be useful for improving the performance of the early exits. From this, we consider three training objectives obtained from the above building blocks :

- **Supervised** ($\mathcal{L}_{\text{sup}}$). Purely next-token cross-entropy against ground-truth labels, that is, $\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{CE}}$;
- **Self-supervised** ($\mathcal{L}_{\text{self}}$). KL divergence that encourages each head to mimic the teacher's full probability distribution, that is, $\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{KL}}$;
- **Hybrid** ($\mathcal{L}_{\text{hyb}}$). The sum of the two losses above, weighted by coefficients $\alpha \in (0, 1)$, that is, $\mathcal{L}_{\text{hyb}} = \alpha \mathcal{L}_{\text{CE}} + (1 - \alpha) \mathcal{L}_{\text{KL}}$.

As reported in Section 4, the self-supervised objective ($\mathcal{L}_{\text{self}}$) yields the most faithful approximation of the teacher's behavior while preserving calibration, and therefore constitutes our default choice for all subsequent experiments.

3.4 Calibration and Inference

After training the early exit heads, the next crucial step involves calibrating and using these heads during model inference. This process is divided into two main stages : calibration of the confidence thresholds and the application of these thresholds during inference.

3.4.1 Calibration of Confidence Thresholds

In our approach, we evaluated three different confidence metrics for calibrating early exit thresholds : (1) maximum probability, (2) entropy of the predicted distribution, and (3) the difference between the top two probabilities ("breaking ties"). To determine which metric best separates correct from incorrect predictions, we analyzed their ability to discriminate between right and wrong outputs using ROC (Receiver Operating Characteristic) curves (Figure 1).

After training, we computed the ROC curves for each metric by plotting the true positive rate against the false positive rate as the threshold varies, using the predictions from

the early exit heads. This analysis was performed across six Pythia models, ranging from 70M to 2.8B parameters. The results are summarized in Figure 1, which displays the ROC curves for all metrics and models.
Our findings consistently show that entropy outperforms the other metrics in every case, achieving a higher area under the curve (AUC) regardless of model size. This indicates that entropy provides a more reliable separation between correct and incorrect predictions, making it the most effective metric for threshold calibration in our early exit framework. Furthermore, we observe that the AUC of the entropy metric tends to increase with model size, suggesting that early exit mechanisms become even more effective as the underlying model grows larger.

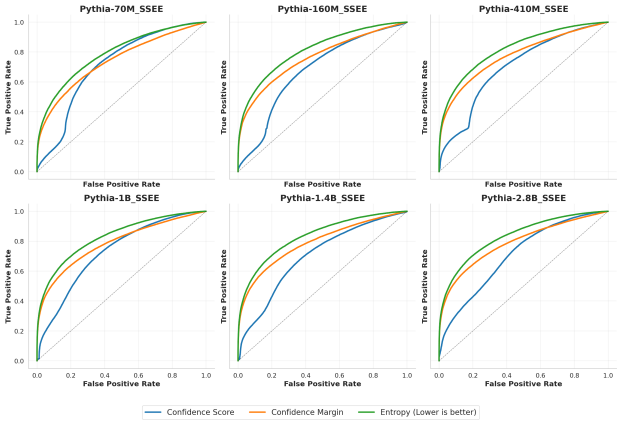


FIGURE 1 – ROC curves for three confidence metrics across six Pythia models (70M to 2.8B). Entropy consistently achieves the highest AUC.

To calibrate the confidence thresholds, we perform a full epoch over our calibration dataset. For each example in the calibration set and for each early exit head, we compute the head output for every token in the sequence, then apply the confidence metric c to each token output, and store all these scores in a global vector $\mathbf{c}_k \in \mathbb{R}^{M_{\text{cal}}}$, where M_{cal} is the total number of tokens in the calibration dataset.
Correspondingly, let $\mathbf{t}_k$ be a binary vector of length M_{cal} where each element corresponds to the correctness of the prediction associated with the respective element in $\mathbf{c}_k$. Specifically, $t_{k,j}$ is 1 if the j^{th} prediction at head k matches the j^{th} prediction of the underlying model $f_\theta(\mathbf{x})$, otherwise 0 :

$$\begin{aligned} t_{k,j} &= \mathbb{1}_{\{\arg\max \mathbf{p}_{k,j} = \arg\max \mathbf{p}_{\theta,j}\}} \\ &= \begin{cases} 1 & \text{if } \arg\max(\mathbf{p}_{k,j}) = \arg\max(\mathbf{p}_{\theta,j}) \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

To determine the confidence threshold for each early exit head, we utilize the calibration set to empirically estimate the relationship between the confidence metric and prediction correctness. Specifically, for each head k, we sort the calibration metric values in ascending order. Given a user-specified confidence level $\epsilon \in [0, 1]$, which represents the minimum desired proportion of correct predictions above the threshold, we identify the smallest metric value such that the proportion of correct predictions among all samples with higher (or equal) metric values is at least ϵ. This procedure ensures that, during inference, predictions made with a confidence metric exceeding the threshold are correct with probability at least ϵ, thereby providing a principled trade-off between computational efficiency and predictive accuracy. Formally, the threshold for each head is defined as follows.
Recall that $\mathbf{c}_k$ and $\mathbf{t}_k$ denote, respectively, the vectors of confidence metric values and correctness indicators for head k, as defined in equation (2). To proceed, we jointly sort these vectors in ascending order according to the values in $\mathbf{c}_k$, resulting in ordered sequences $\mathbf{c}_k = (c_{k,1}, c_{k,2}, \ldots, c_{k,M_{\text{cal}}})$ and $\mathbf{t}_k = (t_{k,1}, t_{k,2}, \ldots, t_{k,M_{\text{cal}}})$ such that $c_{k,1} \leq c_{k,2} \leq \ldots \leq c_{k,M_{\text{cal}}}$.
We then define $\hat{j}$ as the smallest index for which the proportion of correct predictions among all samples with confidence at least $c_{k,\hat{j}}$ meets or exceeds the target confidence level ϵ :

$$\frac{\sum_{i=\hat{j}}^{M_{\text{cal}}} t_{k,i}}{M_{\text{cal}} - \hat{j} + 1} \geq \epsilon.$$

Then, let the threshold τ_k be defined as :

$$\tau_k = c_{k,\hat{j}}.$$

In other words, τ_k is the value of the confidence metric at the index $\hat{j}$, where $\hat{j}$ is the smallest index such that the proportion of correct prediction in the remaining samples is at least ϵ.

3.4.2 Inference Process

After establishing the confidence thresholds for each early exit head through the calibration process, the model is then ready to utilize these thresholds during the inference phase to efficiently process new inputs.
During inference, each input $\mathbf{x}$ is sequentially processed through the model's layers, with the possibility of early termination at any of the early exit heads h_k. For each head $k \in \{1, 2, \ldots, K\}$, we compute :

$$\begin{aligned} \mathbf{p}_k &= h_k(\mathbf{x}), \\ c_k &= c(\mathbf{p}_k). \end{aligned}$$

We then return $\mathbf{p}_k$ as output from the model if :

$$c_k \geq \tau_k.$$

If no head satisfies this inequality, we return $\mathbf{p}_\theta$.
This calibrated and threshold-driven early exit mechanism allows the model to balance efficiency with accuracy, ensuring that resource-intensive computations are only performed when necessary.

4 Experiments and Results

We organize our experimental evaluation into two complementary parts [1]. First, we evaluate calibrated early exits on

1. All computations are run on a server with an Intel(R) Xeon(R) Gold 5120 CPU and a Tesla V100 GPU with 32GB of Vram and 64GB of RAM. Code used in experiments to train and evaluate can be found under : https ://anonymous.4open.science/r/BranchyLLM-B870

standard inference tasks using the Pythia suite (70M–2.8B parameters). We describe the training process for early exit heads and analyze how accuracy-speedup trade-offs scale across model sizes on benchmark tasks. Second, we extend our method to speculative decoding, introducing Dynamic Self-Speculative Decoding (DSSD), which eliminates the need for manual hyperparameter tuning. Throughout our experiments, we use Pythia as a representative example that allows us to scale according to hardware capabilities, though our approach is theoretically applicable to any language model architecture.

4.1 Training

In the training phase, we systematically compared the three loss functions described in Section 3.2 : supervised (cross-entropy), self-supervised (KL divergence to the main model's output), and a hybrid of both. Our primary goal was to select a training objective that enables early exit heads to best approximate the main model's predictions while also providing meaningful uncertainty estimates.

We found that the self-supervised loss, which encourages each early exit head to mimic the output distribution of the main model, consistently led to the best alignment with the main model's predictions. This loss not only matches the output probabilities but also preserves the calibration properties necessary for reliable early exits.

Additionally, we experimented with two initialization strategies for the early exit heads : (1) copying the weights from the main model's final classification head (lm_head), and (2) random initialization. When the early exit heads are initialized by copying the lm_head, they start with a lower loss and initially perform better. However, as training progresses, the randomly initialized heads quickly overtake the copied ones in both loss and performance.

A key observation is that heads copied from the lm_head tend to be overconfident, even when making incorrect predictions. This results in low entropy outputs and a lack of meaningful uncertainty, which is detrimental for early exit decisions. In contrast, randomly initialized heads, when trained with the self-supervised loss, develop a better notion of uncertainty, producing higher entropy outputs when unsure. This property is crucial for effective early exit mechanisms, as it allows the confidence metric to reliably distinguish between correct and incorrect predictions.

For our training experiments, we launch training with the self-supervised loss described above on all Pythia models from 70M to 2.8B parameters. We use a learning rate of 5×10^{-5} and train on 500,000 examples sampled from the MiniPile dataset. Four early exit heads are added at the locations described in the Methodology section.

4.2 Inference

For the inference evaluation, each model is evaluated with different values of the confidence threshold parameter ϵ, ranging from 0.5 to 1.0 in increments of 0.05. This systematic sweep allows us to analyze the trade-off between computational savings and predictive accuracy.

We follow the same set of benchmarks as the original Pythia paper : WSC, Winogrande, SciQ, PIQA, LogiQA, LAMBADA_OpenAI, ARC Easy, and ARC Challenge. For each benchmark and each value of ϵ, we report both the benchmark score and the mean output layer, translated to a speedup percentage.

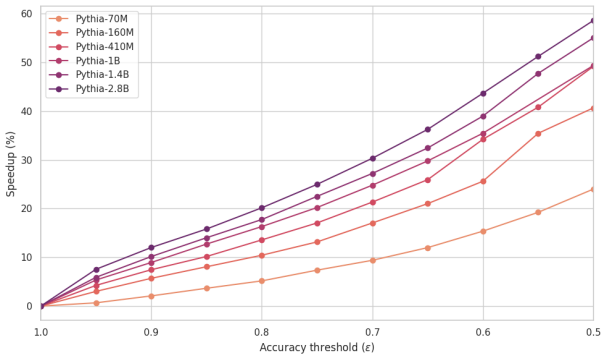


FIGURE 2 – Speedup vs. confidence threshold ϵ for different Pythia model sizes.

TABLE 1 – Benchmarks results for Pythia-2.8B. Each line corresponds to a value of ϵ ranging from 0.5 to 1.0 in steps of 0.05 (top to bottom), with the bottom line being the baseline ($\epsilon = 1$). More benchmarks available in the appendix.

ϵ	Speedup	winogrande	piqa
0.50	58.6%	0.541 ± 0.014	0.615 ± 0.011
0.55	51.2%	0.564 ± 0.014	0.635 ± 0.011
0.60	43.7%	0.565 ± 0.014	0.643 ± 0.011
0.65	36.2%	0.565 ± 0.014	0.668 ± 0.011
0.70	30.3%	0.586 ± 0.014	0.681 ± 0.011
0.75	25.0%	0.579 ± 0.014	0.690 ± 0.011
0.80	20.2%	0.574 ± 0.014	0.707 ± 0.011
0.85	15.8%	0.579 ± 0.014	0.719 ± 0.010
0.90	12.0%	0.571 ± 0.014	0.730 ± 0.010
0.95	7.6%	0.572 ± 0.014	0.733 ± 0.010
1.00	0.0%	0.571 ± 0.014	0.742 ± 0.010

The results of the benchmark on Pythia-2.8B are shown in Table 1. The results for the other model sizes are provided in the appendix.

Our results show that for some benchmarks, such as Winogrande and WSC, the reduction in computation has little to no effect on accuracy, even with aggressive early exiting. However, some benchmarks experience a drop in performance when using very aggressive thresholds (lower ϵ). Despite this, for moderate values of ϵ, the benchmark scores remain close to those of the main model, while achieving speedups between 10% and 20% depending on the model size.

Interestingly, we observe that smaller models in the Pythia suite, such as the 70M and 160M parameter variants, can even show improvements on certain benchmarks when early exits are used aggressively. This suggests that early exit mechanisms may help regularize smaller models or mitigate overfitting in some cases.

Beyond benchmark results, we also observe that the speedup achieved by early exit increases with the size of the

model. Larger models benefit more from early exit, with greater computational savings for the same threshold values. This effect may be explained by several factors : (1) the relative size of the early exit heads becomes less significant as model size increases, and (2) the intermediate features in larger models may provide better representations, enabling more accurate early predictions. This trend is illustrated in Figure 2, which shows the speedup as a function of the threshold for all model sizes.

All detailed tables of benchmark scores and speedup for each model and threshold are provided in the appendix. While our results demonstrate the effectiveness of early exits, it is important to consider the limitations and broader impacts of this approach.

To address the accuracy drops observed with aggressive early exiting while maintaining computational efficiency, we adapt our calibrated early exit mechanism to speculative decoding. This approach allows us to achieve significant speedups without compromising the final output quality, as the full model verifies all predictions.

4.3 Dynamic Early Exit for Speculative Decoding

Having established the effectiveness of calibrated early exits for single-token prediction, we now extend our method to *speculative decoding*—a technique where a faster draft model proposes multiple tokens that are then verified in parallel by the full model. This approach, exemplified by LayerSkip [10], can significantly accelerate autoregressive generation. However, LayerSkip requires practitioners to manually tune two coupled hyperparameters : (i) which intermediate head/layer to use for drafting, and (ii) how many tokens to speculate per round. Finding optimal settings often requires exhaustive grid search across prompts and tasks, with 24 configurations tested in our experiments.

Our key contribution is to replace these two discrete knobs with a *single continuous accuracy threshold* $\epsilon \in [0, 1]$, leveraging the same calibrated entropy-based exit mechanism from Section 3.4.1. The key insight is that per-token confidence naturally determines both *where* to exit (head selection) and *when* to stop drafting (adaptive speculation length). This unified approach, which we call **Dynamic Self-Speculative Decoding (DSSD)**, eliminates manual tuning while achieving superior acceptance rates : on Pythia-2.8B, DSSD reaches **88.8% acceptance at** $\epsilon = 0.9$ compared to LayerSkip's best of 53.6%, a **1.66× improvement**.

4.3.1 DSSD Algorithm

DSSD extends the inference procedure from Section 3.4.2 to multi-token drafting :

Drafting phase. Starting from the current context, we evaluate early-exit heads sequentially from shallowest to deepest. For each position, we select the first head k whose entropy $H(\mathbf{p}_k) < \tau_k(\epsilon)$ falls below its calibrated threshold. We append the predicted token to a draft buffer and continue until : (i) the buffer reaches a preset limit (e.g., 32 tokens), (ii) an end-of-sequence token is generated, or (iii) no head meets the confidence threshold, triggering a fallback to the full model. This produces a variable-length draft sequence whose length adapts to local token difficulty.

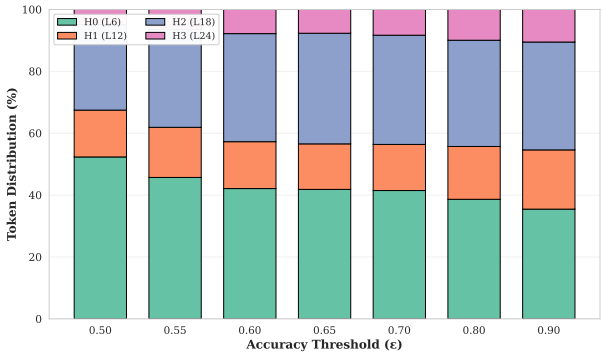


FIGURE 3 – Adaptive head selection : token distribution across heads H0(L6), H1(L12), H2(L18), H3(L24) at different accuracy thresholds ϵ, where notation $H_i(L_j)$ indicates head i at layer j. As ϵ decreases, the system shifts from primarily using H0 to a balanced mix across all heads.

Verification phase. We pass the entire draft buffer through the full model in a single forward pass, obtaining target predictions for all positions simultaneously. We then compare each drafted token with the corresponding full-model prediction : accepted tokens are committed to the output, while the first mismatch triggers an immediate rewrite with the full-model token. This produces an *atomic commit* of the accepted prefix. The mean number of accepted tokens per verification round acts as an implicit, adaptive speculation budget.

Comparison to LayerSkip. In LayerSkip, practitioners fix a single head (e.g., layer 6) and a speculation length (e.g., 10 tokens) for all contexts. Our method dynamically chooses the head on a per-token basis and automatically adjusts the effective speculation length through the acceptance mechanism, adapting to context difficulty without manual intervention.

4.3.2 Experimental Design

We evaluate DSSD on Pythia-2.8B with 200-token greedy decoding, comparing :

- **DSSD (ours)** : Accuracy sweep $\epsilon \in \{0.5, 0.55, 0.6, 0.65, 0.7, 0.8, 0.9\}$ (7 levels).
- **LayerSkip (fixed)** : Exhaustive grid search over 4 heads × 6 speculation lengths (24 configurations).

Experiments use prompts sampled from our dataset to test the method across diverse topics.

4.3.3 Results and Analysis

Superior acceptance with lower overhead. DSSD at $\epsilon = 0.9$ achieves **88.8% acceptance rate**—1.66× better than LayerSkip's best (53.6%). This improvement translates directly to reduced wasted computation : DSSD discards only 8 tokens versus LayerSkip's 128 tokens, a **14× reduction** in waste. Figure 3 illustrates this adaptive behavior.

Understanding layer concentration. An important observation from Figure 3b is that mean exit layers concentrate

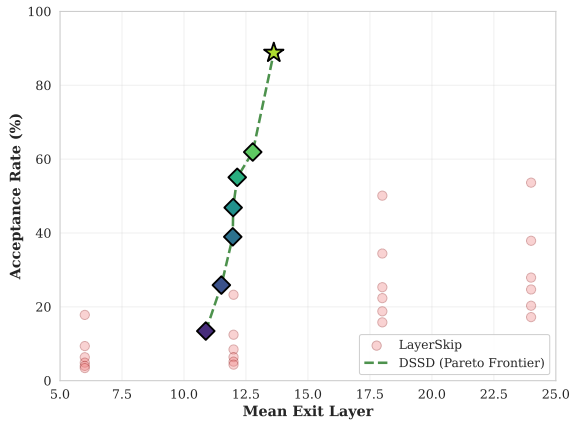


FIGURE 4 – Acceptance rate vs mean exit layer during drafting across all configurations. The dashed green line traces DSSD's frontier (diamonds), while LayerSkip configurations (circles) are dominated. The star highlights DSSD at $\epsilon = 0.9$, achieving 88% acceptance with only 13.6 mean exit layers.

in a relatively narrow range (10.9–13.6 across all ϵ values), spanning only 2.74 layers despite having 4 discrete exit points at [6, 12, 18, 24]. This concentration is **expected and optimal** rather than anomalous : (i) only 4 discrete exit points naturally limit the range, (ii) calibrated thresholds steer tokens toward heads H1-H2 (layers 12-18) as these offer the best efficiency-quality trade-off, and (iii) the narrow span represents successful optimization—the system has identified that most tokens benefit from moderate depth.

4.3.4 Summary

Our dynamic early-exit controller unifies head selection and speculation length into a single accuracy threshold ϵ, eliminating exhaustive hyperparameter tuning. Across 31 configurations tested on Pythia-2.8B, our method demonstrates :

- **Superior acceptance** : 1.66× higher than best LayerSkip (88.8% vs 53.6%)
- **Reduced waste** : 14× fewer discarded tokens (9 vs 128)
- **Earlier exit** : 10.4 layers shallower on average (13.6 vs 24.0)
- **Zero tuning** : Single threshold vs 2D grid search

As model size increases and exhaustive search becomes prohibitive, the automatic adaptation to token difficulty positions DSSD as a scalable, user-friendly alternative to fixed speculative strategies.

5 Limitations and Potential Impacts

While our early exit approach enables acceleration of large language models, the speedup gains without DSSD can remain modest when aiming to preserve the original LLM's accuracy. However, we observe that for certain benchmarks, early exits do not lead to any noticeable degradation in accuracy, even with significant acceleration.

Another important trend highlighted by our experiments is that the speedup obtained through early exits tends to increase with model size. Larger models appear to benefit more from this mechanism, both in terms of computational savings and in the stability of their predictions under early exit. Nevertheless, due to our limited computational resources, we were unable to train and evaluate early exit models on the largest architectures available. Confirming and further exploring this trend would require access to greater computing power.

6 Conclusion

In this work, we introduced a modular early exit mechanism for large language models, allowing inference to terminate early based on calibrated confidence metrics. Our approach is easy to integrate into existing transformer architectures and does not require retraining the backbone model. Through extensive experiments on the Pythia suite, we demonstrated that early exits can provide significant inference speedups, especially for larger models, while maintaining high accuracy on several benchmarks.

Furthermore, we introduced Dynamic Self-Speculative Decoding (DSSD), which integrates calibrated heads into a speculative decoder to achieve 1.66× higher token acceptance rates compared to manually-tuned LayerSkip baselines. The zero-tuning property and automatic adaptation to token difficulty make DSSD particularly attractive for practical deployment.

Overall, our findings suggest that exploiting the natural variability in token difficulty is a promising direction for accelerating large language models. We encourage the community to build upon this work, exploring new strategies and applications to enable efficient and scalable deployment of LLMs in real-world, resource-constrained environments.

Références

[1] Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. BinaryBERT : Pushing the Limit of BERT Quantization, July 2021. arXiv :2012.15701 [cs].

[2] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia : A suite for analyzing large language models across training and scaling, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information*

Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[4] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023.

[5] Kehan Chen, Jinchao Wang, Xun Liu, et al. SmartBERT : Dynamic layer skipping and early exiting for efficient transformer inference. In *Proc. of IJCAI*, 2023.

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM : Scaling Language Modeling with Pathways, October 2022. arXiv :2204.02311 [cs].

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale. *arXiv :2010.11929 [cs]*, October 2020. arXiv : 2010.11929.

[9] Maha Elbayad, Laurent Besacier, and Jakob Verbeek. Depth-adaptive transformer. In *Proc. of ICLR*, 2020.

[10] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. Layerskip : Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, page 12622–12642. Association for Computational Linguistics, 2024.

[11] Angela Fan, Edouard Grave, and Armand Joulin. Reducing Transformer Depth on Demand with Structured Dropout, September 2019. arXiv :1909.11556 [cs, stat].

[12] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The Pile : An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv :2101.00027*, 2020.

[13] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, and et al. Training compute-optimal large language models. *arXiv preprint arXiv :2203.15556*, 2022.

[14] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification, 2018.

[15] Jean Kaddour. The minipile challenge for data-efficient language models. *arXiv preprint arXiv :2304.08442*, 2023.

[16] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv :2001.08361*, 2020.

[17] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023.

[18] Zhengxin Li, Shiyang Shen, Yichong Xu, et al. SkipBERT : Skipping layers for efficient BERT inference. In *Proc. of ACL*, 2022.

[19] Tianyi Liu, Shuhe Ren, Hao Zhou, et al. Early exit is a natural capability of transformer-based language models. *arXiv preprint arXiv :2402.00000*, 2024.

[20] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. Fastbert : a self-distilling bert with adaptive inference time, 2020.

[21] Alec Radford, Jong Wook Jeong, Jack Clark, et al. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv :2302.04200*, 2023.

[22] Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From Words to Watts : Benchmarking the Energy Costs of Large Language Model Inference, October 2023. arXiv :2310.03003 [cs].

[23] Simone Scardapane, M. Scarpiniti, E. Baccarelli, and A. Uncini. Why should we add early exits to neural

networks ? *Cognitive Computation*, 12 :954 – 966, 2020.

[24] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. The right tool for the job : Matching model complexity to instance difficulty. In *Proc. of ACL*, 2020.

[25] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. Q-BERT : Hessian Based Ultra Low Precision Quantization of BERT, September 2019. arXiv :1909.05840 [cs].

[26] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient Knowledge Distillation for BERT Model Compression, August 2019. arXiv :1908.09355 [cs].

[27] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. MobileBERT : a Compact Task-Agnostic BERT for Resource-Limited Devices, April 2020. arXiv :2004.02984 [cs].

[28] Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. Branchynet : Fast inference via early exiting from deep neural networks, 2017.

[29] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. LaMDA : Language Models for Dialog Applications, February 2022. arXiv :2201.08239 [cs].

[30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA : Open and Efficient Foundation Language Models, February 2023. arXiv :2302.13971 [cs].

[31] Florian Valade, Mohamed Hebiri, and Paul Gay. Eero : Early exit with reject option for efficient classification with limited budget, 2024.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[33] V. Vovk, A. Gammerman, and C. Saunders. Machine-learning applications of algorithmic randomness. In *In Proceedings of the Sixteenth International Conference on Machine Learning*, pages 444–453. Morgan Kaufmann, 1999.

[34] Jing Wang, Hao Chen, Renqian He, et al. FREE : Fast, reliable early-exit transformers for efficient autoregressive generation. In *Proc. of EMNLP*, 2023.

[35] Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT : Dynamic Early Exiting for Accelerating BERT Inference, April 2020. arXiv :2004.12993 [cs].

[36] Ji Xin, Raphael Tang, and Jimmy Lin. BERxiT : Early exiting strategies for BERT. In *Proc. of EACL*, 2021.

[37] Yiming Yang, Yi Zheng, Xiaobo Li, et al. DEED : Dynamic early exiting for efficient decoding of large language models. In *Findings of NAACL*, 2024.

[38] Z. Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant : Efficient and affordable post-training quantization for large-scale transformers. *Neural Information Processing Systems*, 2022.

[39] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT : Open Pre-trained Transformer Language Models, June 2022. arXiv :2205.01068 [cs].

[40] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. BERT Loses Patience : Fast and Robust Inference with Early Exit, October 2020. arXiv :2006.04152 [cs].

[41] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A Survey on Model Compression for Large Language Models, September 2023. arXiv :2308.07633 [cs].

The consequences of a perfect LLM fingerprinting function

Rossana Cometa[1], Erwan Le Merrer[1], Gilles Tredan[2]

[1] Inria de l'Université de Rennes 1
[2] LAAS/CNRS

rossana.cometa@inria.fr

Abstract

LLM fingerprinting (FP) consists in identifying a remote model using only regular query/answer information. Identifying good FP strategies is an active research area. In this short paper, rather than finding good FP strategies, we assume the existence of a perfect one and model its cost as a lower bound of the cost of any FP approach. We then explore on GPT2 possible relaxations of this arguably simplistic model. This opens the discussion on the realistic cost/accuracy trade-off for future schemes.

Keywords

LLMs, Fingerprint, Operational cost

1 Introduction

Fingerprinting, watermarking, change detection : many contemporary problems revolve around the identification of a target LLM using only (black-box) interaction traces. Indeed, in these challenging regulatory and auditing settings, identifying the target model (either through a fingerprint, or indirectly through the absence of detected change) appears as a fundamental building block. Intuitively, assessing the safety of a model today is of little help if the auditor cannot properly detect tomorrow that its behaviour differs from the assessed one.

State-of-the-art fingerprinting schemes [1, 2] for large language models (LLMs) operate under the work assumption that variants from a given model architecture *must* be identified as a whole (i.e., the same) identical entity. This is because variants from an expensive-to-train architecture are equally valuable. Under this work assumption, these schemes track the best possible identification accuracy.

In this short paper, we take a step to examine the logical consequences of such a design choice. We start by modeling the fingerprinting process using an ideal function, which permits to reason about operational cost in Section 2, before we conjecture practical implications when this cost is not reachable in practice. Finally, we illustrate in Section 3 the results of our experiments with GPT2 to showcase that this modelization makes sense in the wild.

2 The Topology of Fingerprints

Definition 1 (Model Space)**.** *Let $\Theta \subseteq \mathbb{R}^d$ represent the space of LLM parameters, where d is the dimension (e.g., $d = 70 \times 10^9$). To further simplify the approach, we assume models exhibit deterministic behaviour (temperature set to zero), and no context. LLM behaviour space is here the space of all possible (prompt/answer) couples.*

These hypotheses represent an ideal analytical situation, at the expense of realism. Nevertheless, they all concur to simplifying the fingerprinting : any realistic fingerprinting approach will be strictly harder.

Definition 2 (Perfect Fingerprint)**.** *Given the set $\mathcal{F}$ of observable features used for identification, a **perfect fingerprint** is a function $FPP : \Theta \to \mathcal{F}$ such that :*

$$\theta_1 \neq \theta_2 \implies FPP(\theta_1) \neq FPP(\theta_2)$$

Assuming the availability of such a FPP is a strong hypothesis. It is arguably the implicit function sought by any fingerprinting method. Intuitively, an ideal FPP would be equivalent to any LLM truthfully responding to the query "what are your parameters ?". In this study, we focus on models all sharing the same architecture, so that it's possible for us to compare their parameters. Allowing multiple architectures would extend the parameter space, and thus make the problem even harder.

2.1 The Cost of Perfect Fingerprinting

Proposition 1 (Fingerprinting Cost)**.** *Let FPP be a perfect fingerprint. By the pigeonhole principle :*

$$|\mathcal{F}| \geq |\Theta|$$

If each of the d parameters is represented with k quantization levels (e.g., $k = 2^{16}$ for fp16), then :

$$|\Theta| \approx k^d$$

*The information-theoretic cost to distinguish **all** fingerprints with a FPP is :*

$$\begin{aligned} \text{cost}(\text{FPP}) &= \log_2(|\mathcal{F}|) \geq \log_2(|\Theta|) \\ &= d \cdot \log_2(k) \textit{ bits} = O(d) \textit{ bits} \end{aligned}$$

Example 1 (Numerical illustration)**.** *Consider GPT-2 with* $d = 124 \times 10^6$ *parameters quantized to 16 levels (fp16)* ($k = 2^{16}$) *:*
Perfect fingerprint : $\text{cost}(\text{FPP}) = d \cdot \log_2(k) \approx 124 \times 10^6 \times 16 = 1984 \times 10^6$ *bits (248 MB).*

2.2 Coarse Perfect Fingerprints

In practice, perfect fingerprinting (distinguishing **all** $\theta \in \Theta$) is thus prohibitively expensive. State-of-the-art fingerprinting methods [1, 2] do not aim for global uniqueness, but instead assign the **same fingerprint** to a model θ_0 and all models "similar" to it. We can model this as follows :

Definition 3 (Coarse Perfect Fingerprint)**.** *Let* $\sim$ *be an equivalence relation on* Θ*. A* ***coarse perfect fingerprint*** *is a function* $\text{FPP}_\text{c} : \Theta \to \mathcal{F}$ *such that :*

$$\theta \sim \theta_0 \implies \text{FPP}_\text{c}(\theta) = \text{FPP}_\text{c}(\theta_0)$$

That is, FPP_c *is constant on each equivalence class* $[\theta_0]_\sim$ *over* Θ*.*

Proposition 2 (Cost Reduction via Coarse FPP)**.** *If* Θ *is partitioned into equivalence classes* $[\theta]$ *of size* T*, the information-theoretic cost becomes :*

$$\begin{aligned}\text{cost}(\text{FPP}_c) = \log_2(|\mathcal{F}|) &\geq \log_2\left(\frac{|\Theta|}{T}\right) \\ &= \text{cost}(\text{FPP}) - \log_2(T)\end{aligned}$$

State-of-the-art fingerprinting schemes operate at a radically lower scale : [2] distinguishes models using 300 samples from TruthfulQA, while [1] requires only 8 queries to distinguish among 42 LLMs. To match the cost of state-of-the-art schemes, each equivalence class would need to contain an astronomically large number of models, far beyond any realistic notion of identical behavior. This gap is partly explained by a difference in goals : since most of existing schemes focus on intellectual property protection, they're designed to consider two models identical if they share the same base model. In contrast, our setting requires detecting behavioral changes since a model's last audit : under this stricter notion of identity, our analysis confirms that no fingerprint of practical size can avoid false positives, even in our deterministic and favorable setting. This gap will only widen in more complex, realistic scenarios.
This section showed that even in a restricted, favorable setup, a perfect fingerprinting scheme requires an impractical query size : any small sized fingerprint will cause errors in identification (false positives) due to collision, by the pigeonhole principle. We now reach the same conclusion in Section 3 under additional geometric assumptions on Θ.

3 A Geometrical Illustration

For the purposes of this illustration, we make two additional assumptions : that Θ is uniformly populated by equivalence classes, and that they correspond to l_2-balls of radius l. We selected 10 different models, all sharing the GPT-2 architecture, from HuggingFace. For each base model, we generated variants by fine-tuning on 10, 15, or 20 examples from the WikiText-2 dataset, for 1 or 2 epochs, with learning rate 10^{-6} or 5×10^{-6}. We kept only the variants whose perplexity on 128 examples from WikiText-2 differed from the base model by less than 5%. When more than 5 such variants were available, we kept the 5 with the largest l_2 distance from their base model. This process leads to the examination of the pairwise l_2 distances on 58 models across all families, split in two classes : "identical" and "different", the logic being that the variants ("identical") should be closer to their original model than to other specialized GPT2 models ("different"). Figure 1 shows a clear separation between the two classes at distance around 1.0.

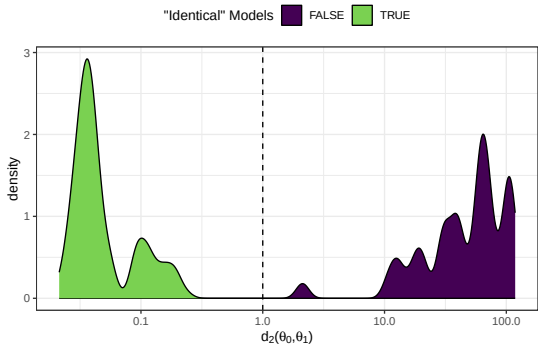


FIGURE 1 – PDF of the distances between models in the two "identical" or "different" classes.

Let's consider a back-of-the-envelope upper bound : fp16 uses 5 exponent bits, 10 fraction bits, and 1 sign bit. Considering a maximal distance of 1.0 between identical models (arbitrary dashed cutoff in Figure 1), assume two models are identical if and only if all their exponent bits are identical. We therefore allow 11/16 bits per dimension to change between identical models. As a consequence, a coarse fingerprinting scheme must determine the 5 remaining bits per dimension : $\text{cost}(\text{FPP}_c)$ is thus $5/16 \approx 31\%$ of $\text{cost}(\text{FPP})$.
In conclusion, having equivalence classes in practical settings does not significantly change the argument from Section 2 (same order of magnitude), and that a critical cost/accuracy trade-off is **present by design**, due to the fingerprint sizes in the state-of-the-art.

Références

[1] Pasquini, Kornaropoulos, and Ateniese. LLMmap : Fingerprinting for large language models. In *USENIX Sec.*, 2025.

[2] Zhang, Li, Qian, Zhang, Liu, Qiao, and Shao. Reef : Representation encoding fingerprints for large language models. *ICLR*, 2025.

Can LLMs help lawyers? Argument analysis in legal texts

Karla Salas-Jimenez

Université Toulouse Capitole, IRIT, Toulouse, France

Karla-Denia.Salas-Jimenez@irit.fr

Résumé

Le raisonnement juridique dépend fondamentalement de la capacité des professionnels du droit à identifier, formaliser et évaluer les arguments dans les textes juridiques. Cependant, la recherche actuelle repose sur un ensemble limité de ressources annotées et intègre rarement la logique formelle dans l'ensemble de la chaîne d'analyse. Ce travail propose un cadre modulaire qui combine l'extraction d'arguments, la traduction NL-logique et l'évaluation de la force des arguments, en utilisant des stratégies de prompting et des schémas argumentatifs. L'objectif est de faire progresser le raisonnement juridique computationnel en proposant des outils qui améliorent la transparence et la cohérence dans la prise de décision juridique, en créant de nouveaux corpus annotés variés pour soutenir des modèles plus robustes, et en contribuant à des systèmes d'IA qui aident les juristes à construire et à interpréter des arguments juridiques bien fondés grâce à une évaluation basée sur la logique.

Mots-clés

Extraction d'arguments, Raisonnement juridique, Logique déontique défaisable, Programmation par ensembles réponses.

Abstract

Legal reasoning fundamentally depends on the the ability of the legal professionals to identify, formalize, and evaluate arguments in legal texts. However, current research relies on a limited set of annotated resources and rarely integrates formal logic into the full analysis pipeline. This work proposes a modular framework that combines argument extraction, natural language to logic (NL-Logic) translation, and argument strength evaluation, using prompting strategies and argument schemes. The goal is to advance computational legal reasoning by offering tools that improve transparency and consistency in legal decision-making, creating new and diverse annotated corpora to support stronger models, and contributing to AI systems that help lawyers construct and interpret well-founded legal arguments through Logic-based evaluation.

Keywords

Argument Mining, Legal Reasoning, Defeasible Deontic Logic, Answer Set Programming

1 Introduction

Argumentation plays a crucial role in legal reasoning. Lawyers must construct coherent and logically structured arguments to support their clients' positions, while judges are responsible for evaluating their validity, identifying implicit assumptions, and addressing potential exceptions. Since these tasks are complex and cognitively demanding, automating parts of the argumentative analysis can help improve consistency, transparency, and efficiency in legal decision-making.

Over the past decades, research has explored the intersection between law and computational argumentation. Early work by Mochales and Moens [40] pioneered the automatic identification and classification of arguments in legal texts. Subsequent approaches have incorporated formal and explainable models of legal reasoning. For instance, Collenette et al. [18] proposed an explainable AI framework for legal reasoning applied to Article 6 of the European Court of Human Rights (ECHR), formalizing legal factors using the ADF for kNowledGe Encapsulation of Legal Information for Cases (ANGELIC) methodology within Abstract Dialectical Frameworks (ADFs) [15] and evaluating them in PROLOG to predict ECHR decisions. More recently, the emergence of Large Language Models (LLMs) has opened new possibilities. Trajano et al. [51] employed LLMs to translate natural language arguments into computational representations, while Abdullah et al. [3] analyzed the performance of LLMs in legal argument mining tasks, showing that these models can assist in identifying argumentative components.

However, most existing approaches focus on isolated sub-tasks and are typically evaluated on a single dataset or closely related corpora, often centered on ECHR decisions [44]. Moreover, many proposals either rely exclusively on symbolic methods under controlled settings or use LLMs without integrating them into a structured logical framework. As a result, there remains a gap between natural language argument extraction and formal, logic-based evaluation of argumentative structures.

To address these limitations this research proposes a system that extracts arguments from legal documents, translates them into a logic representation, and evaluates their internal structure and potential contradictions.

The present work describes a research plan, with preliminary results, that aims at achieving the following

contributions:
1) the creation of new corpora for argument mining and NL-Logic translation in the legal domain, and 2) a practical pipeline combining LLM-based extraction, structured prompting, and formal reasoning. These contributions will provide a methodology for future studies, enabling specialized datasets and models for fine-grained legal argument analysis, achieving consistency and fairness in judicial reasoning.
The paper is organized as follows. Section 2 reviews related work on legal argument mining, the formalization of arguments into logic, and the evaluation of argument strength. Section 3 presents the methodology underlying each module of the proposed pipeline. Section 4 reports and analyzes our preliminary results. Finally, the paper is concluded with a discussion of the findings and directions for future work.

2 Related Work

This section is divided into three parts: *Argument Mining*, *Translation of Arguments to Logic*, and *Argument Strength*. The first two are emphasized, as they constitute the main focus of this study. The last is for future work in order to give the completed idea of the work.

2.1 Argument Mining

Argument mining aims to automatically identify and extract argumentative components and their relations from natural language texts in order to generate structured, machine-processable representations for computational argumentation models [16].
Most works rely on argument schemes [39], which "represent stereotypical patterns of reasoning that capture the inferential relationships between premises and conclusions" [52]. In particular, in the legal domain, Atkinson et al. [7] conducted a study on the impact of Walton's conception of argumentation schemes on AI and Law research. Their work shows that incorporating argumentation schemes helps address normative aspects of legal reasoning by operating over a structured knowledge base, thereby providing richer and more accurate representations.
In the legal domain, research has mainly focused on cases from the European Court of Human Rights (ECHR). Poudyal et al. [44] analyzed 20 Decisions and 22 Judgments of the ECHR, noting that Decisions are more concise (≈3,500 words) than Judgments (≈10,000 words) and contain, on average, 18 arguments, while Mumford et al. [41] compiled legal cases under Article 6. Additional datasets from related domains have also been introduced, for example: EthiX [52], built from ethical debates in 22 ethical topics on the Kialo[1] platform, contains 686 arguments categorized into eight ethical classes based on argument schemes. NLAS-multi [49] is a large corpus of 3,810 arguments spanning 20 argumentation schemes and 50 topics, including several legal-related issues, all generated using GPT-4 and Araucaria [46], developed at the Arg-tech Centre, annotates arguments at the level of claims and premises. It includes texts from newspapers, parliamentary records, court reports, magazines, and online sources, covering multiple countries and topics such as human rights and climate change.
Approaches to legal argument extraction range from traditional NLP and machine learning methods [38] to a growing set of LLM-based techniques. Recent work has explored how different prompting strategies shape LLM performance, including studies on clause classification [55], argument identification [22], relation detection [30], and automated debate construction [25]. In parallel, symbolic systems such as ANGELIC [9], applied to ECHR outcome prediction [18], have demonstrated strong performance relative to purely data-driven approaches. More recently Kong et al. [37] proposed to use the LLMs as annotators, using a third LLM as a mediator.
This study also builds on the modular LLM-based annotation pipeline introduced by Berghegger et al. [12], that we describe in Section 3.1.

2.2 Translation of Arguments to Logic

Translating parts of the human thought process, particularly arguments, into formal representations is a highly challenging task. Multiple logical formalisms are available, and a natural question is which of them is most suitable for a given application. One of the most widely used formalisms is First-Order Logic (FOL).
Early efforts in Natural Language-First Order Logic (NL-FOL) translation were rule-based and difficult to scale [1, 14], whereas modern approaches use LLMs, recent work has advanced NL-FOL translation [53, 45] through in-context learning and prompting strategies.
However, FOL presents important limitations in the legal domain [2]. In particular, it does not naturally capture normative notions such as obligations, permissions, and exceptions. Additionally, reasoning in FOL may raise computational complexity concerns in practical derivations, since it is undecidable in general. For these reasons, alternative approaches have emerged.
Trajano et al. [51] explore the use of LLMs to translate natural language arguments into structured computational representations based on argumentation schemes. Their approach combines scheme classification with retrieval-augmented generation to guide the translation process. Results show that LLMs perform well on simple schemes, while more complex argumentative structures require additional contextual guidance. The study demonstrates the feasibility of leveraging LLMs to connect natural language argumentation with formal reasoning frameworks.
Gupta et al. [34] analyze the formalization of human argumentative reasoning by mapping informal logic into Answer Set Programming (ASP), showing that ASP better supports default reasoning and the management of exceptions. Building on this direction, Governatori [31, 33]

[1] https://www.kialo.com/

[2] During the initial stage of this PhD research, the NL→FOL translation was examined, and this limitations were identified.

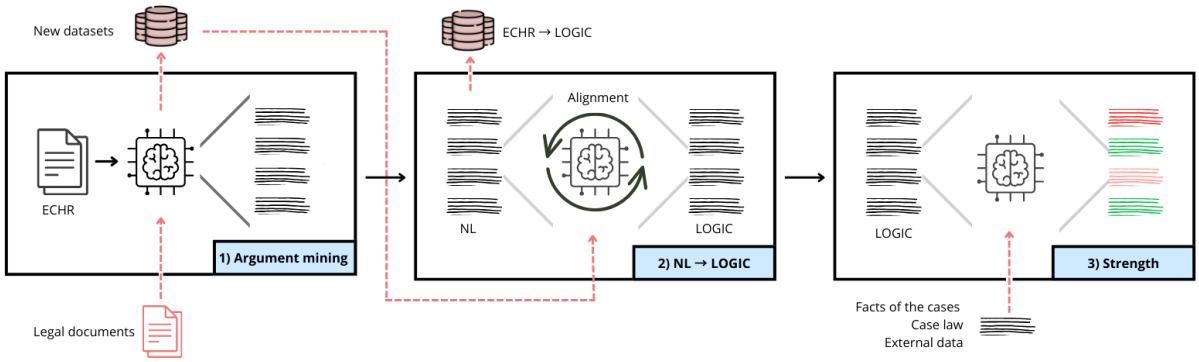


Figure 1: Pipeline from legal text to logical translation and evaluation of arguments strength: (1) argument mining, (2) NL-Logic translation and alignment, and (3) strength evaluation.

first proposed a logic designed to capture key features of the legal domain, the Defeasible Deontic Logic (DDL) which extends Defeasible Logic by incorporating deontic operators.
A DDL consists of three main components:

1. **Facts** - indisputable evidence or conditions of a case.
2. **Rules** - representing legal norms, which can be of three types:
   (a) Constitutive rules: define terms in legal documents.
   (b) Prescriptive rules: assert obligations or prohibitions, potentially organized in compensation chains where the violation of one obligation triggers the next.
   (c) Permissive rules: assert permissions.
3. **Superiority relation** - resolves conflicts between rules.

DDL employs three deontic operators: O (obligation), F (prohibition), and P (permission), which function as modal operators qualifying the truth of propositions. Obligations require a bearer to perform an act or achieve a state, with non-compliance resulting in a violation; prohibitions forbid an act or state, with violations similarly penalized. Permissions hold when no obligation or prohibition forbids an action, with weak permission indicating the absence of restrictions and strong permission representing an explicit exception.
More recently, an implementation of DDL using Answer Set Programming (ASP) as a meta-programming framework has been presented [32], enabling formal reasoning over complex normative structures, employed LLMs to translate arguments into ASP-DDL representations, demonstrating that LLMs can effectively support this task [36].
Given the variety of formalisms available for representing legal norms, Robaldo et al. [47] provide a comparative study of several reasoning frameworks, including ASP, SHACL, DLV, Arg2P, PROLEG, and SPINdle. Their results show that ASP-based reasoners achieve the best computational performance, while also offering a favorable balance between expressivity and ease of representation. Although ASP reasoning is exponential in the worst case [24], modern solvers [27] are efficient in practice. In particular, with bounded arities and restricted domains, we may reduce the problem to propositional ASP where reasoning reduces to NP-complete problem for non-disjunctive programs [20].

2.3 Argument Strength

Argument strength has received much attention in the formal argumentation community [43, 35]. In the field of abstract argumentation [23], arguments strength is usually associated with the acceptability of arguments that can be computed via extension-based semantics (as in Dung's original approach for assessing collective acceptability [23]) or through ranking-based [4, 13] or gradual semantics [17] (when the focus is on individual acceptability). These approaches mainly focus on the "resulting" strength of arguments, *i.e.* how strong they are after facing some conflicts. Some works introduce also a notion of "intrinsic" strength of arguments, using weights to represent how strong are arguments *before* facing the conflicts. In these cases, semantics (*e.g.* extension-based in [48] and gradual in [5]) are adapted to take into account the initial arguments strength for determining their acceptability, *i.e.* their final strength.
In Value-based Argumentation [8], arguments are associated with a value (*i.e.* an abstract label representing a moral or social value), and a preference relation over values allows to assign different strengths to the corresponding argument. The link between Value-based Argumentation and normative reasoning has been emphasized in [11]. However, before using any such approaches for legal reasoning, two challenges must be solved: 1) determining, among the many different approaches for defining the semantics of formal argumentation frameworks, which ones are the best suited for modeling the reasoning schemes of legal practitioners, and 2) how to extract from natural language arguments the formal components used for reasoning (*e.g.* the weights, or the preference order over values). All these works assume that arguments and their relationships have already be obtained (from natural

language or from logical knowledge) and abstracted away. However, arguments strength has also received some attention in structured (logic-based) argumentation and natural language argumentation.

Within the dialectical dimension, Spaans [50] proposes a principle-based approach and concrete methods for computing the intrinsic strength of logic-based arguments from their internal structure. Macagno et al. [39] distinguish premise, conclusion, and inferential (undercutting) attacks, the latter being particularly relevant in legal reasoning.

Beirlaen et al. [10] formalize argument strength through three dimensions: *support* (*e.g.* premises and inference rules reliability), *dialectical* (interactions between competing arguments), and *evaluative* (cumulative support). Approaches for reasoning with abstract argumentation, mentioned previously, focus on the dialectical and evaluative aspects, but ignore the support dimension. For the support dimension, Lenz et al. [38] propose graph-based models labeling edges as support or attack relations.

From a procedural perspective, Gordon et al. [28] introduce the Carneades model, which evaluates arguments through proof standards and dynamically allocated burdens of proof. In this framework, the acceptability of a claim depends not only on its supporting and attacking structure but also on whether it satisfies the applicable proof standard under the current dialectical stage. By distinguishing ordinary premises, assumptions, and exceptions, and by allowing burdens of production and persuasion to shift between parties, Carneades captures an institutional dimension of argumentative strength particularly suited to legal contexts.

Finally, integrating structural and dynamic perspectives, Obermaier et al. [42] show that multiple weak arguments can paradoxically strengthen or weaken a stronger one, highlighting the non-linear dynamics of argumentative influence (in the same vein as the approach by [48]).

3 Methodology

This work proposes a three modules: argument extraction, NL-Logic translation, and strength evaluation. Figure 1 shows the complete pipeline.

3.1 Argument mining

Berghegger et al. [12] propose an LLM-based methodology composed of three pipelines: (1) extracting arguments, composed by the claim and premise(s), directly from the text, (2) extracting claims first and then the complete argument, and (3) extracting the claims, identifying the paragraphs related to those claims, and subsequently identifying the premises within those paragraphs. For evaluation they propose two approaches:

1. **Unstructured evaluation:** Arguments are compared as whole units using vector similarity in two directions (original-to-LLM and LLM-to-original). A match is defined when similarity exceeds a threshold (t = 0.75), and performance is measured using a matching ratio and mean similarity.

2. **Structured evaluation:** Arguments are decomposed into claims and premises. Similarity is computed separately for each component and combined using a weighted formula inspired by similarity measures in logical argumentation [6, 21] ($sim\_arg(A_1, A_2) = \alpha \times simP(P_1, P_2) + (1 - \alpha) \times simC(C_1, C_2)$), where a parameter α (set to 0.7) balances the relative importance of claim and premise similarity.

Their experiments use nine short texts and six long texts from the ECHR dataset tested in *meta-llama/Meta-Llama-3-8B-Instruct*[2] and *Equall/Saul-7B-Instruct-v1*[19]. They also compare their results with traditional ML approaches such as *ArgueMapper/ArgueBuf* [38].

Following this approach, we have run similar experiments on the whole ECHR corpus (Corpus details are in Section 2.1.). Initial results indicate that while LLMs can identify arguments, they often miss key elements, especially in claim detection, and tend to over-generate arguments. As a post-processing step, we introduce a mechanism to merge redundant arguments and refine their structure. To this end, we propose using several LLMs as annotators, and aggregate their results. The arguments extracted by Llama and Saul can be interpreted as alternative annotations of the same legal text. Our objective, in future experiments, is to establish an agreement mechanism between them in order to generate a consolidated final annotation.

Following the findings of Berghegger et al. [12], who report that Llama achieves higher mean similarity scores, we use Llama's arguments as the starting point for the agreement process. In Figure 2 we can observe the agreement process.

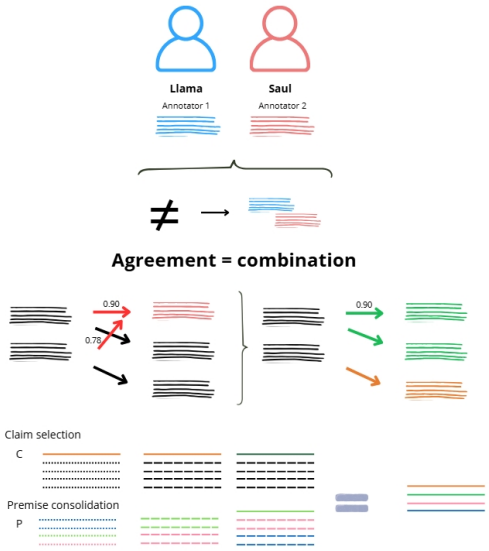


Figure 2: The automatic agreement process between two LLM annotators.

The agreement process consists of four stages:

1. **Matching.** We first matched arguments produced by Saul and Llama using a unstructured similarity threshold of 0.8[3]. This step follows the same procedure employed for matching ECHR arguments with LLM-generated outputs, computing cosine similarity between argument embeddings.
2. **Keep unmatched arguments.** The agreement process is asymmetric, prioritizing LLM1 to preserve its matching ratio and similarity. Its outputs are retained to maintain coverage, while agreement is applied only to matched arguments to reduce redundancy and improve structure.
3. **Make groups.** During matching, it is possible for one Llama argument to match multiple Saul arguments, and vice versa. This may artificially inflate the matching ratio due to highly similar or redundant arguments. To mitigate this effect when multiple matches occurred, we retained only the pair with the highest similarity score. After filtering, we created groups where one Llama argument could correspond to multiple Saul arguments.
4. **Argument combination.** For each group, we constructed a consolidated argument by first determining the claim and then merging the premises.
   (a) Claim selection: all claims within the group were collected and their frequencies computed. If frequencies were equal, the Llama claim was retained (based on prior evidence of higher similarity performance); otherwise, the most frequent claim was selected. Non-selected claims were moved to the premises set.
   (b) Premise consolidation: Similar premises were clustered using the Python string-matching method[4], and one representative premise per cluster was selected, the longest premise containing an article, or otherwise a random one. We intentionally preserve as much information as possible rather than applying aggressive filtering, as providing more context is preferable to omitting potentially relevant content, allowing annotators to determine what is relevant for argument construction.

To enhance the robustness of this module, we evaluated the agreement mechanism under different similarity thresholds. We also incorporated additional LLMs as annotators, such as *Qwen/Qwen2.5-7B-Instruct*, and introduced lexical features to strengthen the structured evaluation process.
Future work includes extending the annotation framework to newly created corpora across different legal domains. Two validation strategies are planned: (1) expert-based evaluation of the extracted arguments to assess their practical relevance, and (2) collaborative refinement aimed at building larger and more diverse annotated corpora.

[3] in 4 the results show that this threshold work better that the others
[4] `difflib.get_close_matches(..., cutoff=0.9)`

Argument	Translation	Attribution
P0 - The applicant considers...	P0 - ...	Applicant
P1 - The Government consider...	P1 - ...	Government
P2 - The Commission finds...	P2 - ...	Commission
P3 - He refers ...	P3 - ...	Applicant
C - There is no ...	C - ...	Undefined

Table 1: Scheme for the manual translation of the ECHR corpus to ASP-DDL.

3.2 Arguments to Logic

The translation methodology presented here is preliminary and part of ongoing work.
To automate the translation we began by refining the annotation guidelines. During this process, we observed that claims and premises frequently begin with attribution markers such as *The applicant considers that*, *The Government submits that*, or *The Commission finds that*. These recurrent formulations introduce additional complexity in the translation step, as they mix argumentative content with speaker attribution.
To address this issue, we separated attribution from the argumentative content by introducing a dedicated *Attribution* attribute and omitting these expressions from the logical translation. The *Attribution* field can take the following values: *Applicant*, *Government*, *Commission*, or *Undefined*.
We then proposed a structured annotation scheme to clarify both the attribution labeling and the subsequent translation process, as shown in Table 1.
For the translation phase, we adopted the ($\approx$ 30) predicates proposed by Governatori [32]. Although annotators were allowed to introduce additional predicates when necessary to capture specific semantic, the use of ASP-DDL predefined predicates was encouraged whenever possible to ensure consistency and formal alignment.
The manual translation process involved two human annotators. Translator 1 holds degrees in Computer Science and Law, while Translator 2 has a degree in Mathematics. This could be interesting for analysing differences between their background in their translations.
To construct the gold standard for each document in the corpus, the annotators conducted video meetings to compare their translations, preserve agreed-upon arguments, and resolve discrepancies in the remaining cases through discussion. An example of the argument translation is provided in Figure 4.
For the automated translation, we adopted a prompt-based strategy testing a zero-shot and few-shot approaches, with step-by-step task instructions to guide the model's reasoning process. Figure 3 shows the prompt template.
We evaluate the prompt by incorporating an explanation accompanied by a brief example of the predicates introduced in ASP to model DDL (<*Explanation of DDL predicates*>), as illustrated in the following example:

permissiveRule(r,x): a legal provision grants a

Template prompt for Argument → ASP-DDL

Your task is to translate the given argument into Defeasible Deontic Logic (DDL) in Answer Set Programming (ASP) syntax, following the exact specifications below.
1. Structure of input:
P0 - <premise 0>
...
PN - <premise N>
C - <claim>
Premises (P0...PN) are factual, legal, or argumentative statements providing support.
The claim (C) is the conclusion supported by the premises.

<Explanation of DDL predicates>
<Argument schemes in ASP-DDL>
<Example(s)>

Now translate the following argument: {}
Output:

Figure 3: Prompt template for translating NL arguments into ASP-DDL.

permission or exception.
Domestic law allows proceedings without an oral hearing in minor cases.
permissiveRule(domestic_law, non(oral_hearing)).

We also include argumentation schemes [29] (<*Argument schemes in ASP-DDL*>) such as *Position to Know* (A claim is presumptively accepted if it is asserted by a source who is in a position to know, unless the source is unreliable.), *Verbal Classification* (If something has a property that entails membership in a category, it can be classified under that category.), *Established Rule* (If the conditions of a valid rule are met and no exceptions or higher-priority rules override it, its conclusion follows.), and *Precedent Case* (If two cases are sufficiently similar and a proposition holds in one, it provides a reason to accept it in the other.). For instance, in the case of verbal classification:

Individual Premise. a has property f.
Classification Premise. For all x, if x has property f, then x can be classified as having property g.
Conclusion. a has property g.

---

constitutiveRule(r_vc, property(a,f)).
applicable(r_vc, property(a,g)) :- property(a,f), property(x,f), property(x,g).

Additionally, we provide examples of translated arguments (<*Example(s)*>). Experiments are conducted under three conditions: without examples, without explanations, and with both explanations and examples. Preliminary results indicate that including examples does not improve performance, LLMs tend to focus excessively on them.
We propose evaluating the translation from two complementary perspectives: first, the semantic similarity between the manual and automated translations, and second, the preservation of the logical structure. A more comprehensive investigation and improved metrics are needed to fully address this goal, but for the present study, we adopt BERTScore [54] to assess semantic similarity.

		Sim.	Gran.	Match.	Mean
Baselines [12]					
Llama		83.42	67.69	72.75	74.62
Saul		75.28	63.48	53.43	64.06
Agreement-based approach					
Llama	0.6	84.24	69.66	82.16	78.69
Llama	0.7	84.15	69.87	81.99	78.67
Llama	0.8	84.00	69.46	84.21	**79.23**
Approach [12] for texts: 33, 35, 38, 40, 41, 42					
Llama		84.29	61.21	56.09	67.20
Saul		82.28	70.43	53.59	68.77
Agreement	0.8	83.24	65.50	75.99	**74.91**

Table 2: Evaluation results for Similarity (Sim.), Granular Similarity (Gran.), Matching Ratio (Match.), and their average (Mean). Baselines from Berghegger et al. [12] (including the missing Saul long-text results) are compared with our agreement-based approach (thresholds 0.6–0.8) with the generalization of this method to six additional ECHR texts. Bold indicates the best Mean score, underlined values denote the best score per metric.

To evaluate logical fidelity, we use a modified version of *LogicSim* [26] adapted to the ASP-DDL syntax. This metric compares a translation x with its reference y across multiple logical dimensions:

$$LogicSim(x, y) = pd + ap + tp + ld + IoU$$

where pd, ap, tp, and ld denote differences in premises, predicates, and logical operators, and IoU measures predicate overlap.

3.3 Arguments strength

Since this research is still in its early stages, we propose a preliminary notion of argument strength based on consistency and absence of contradictions, which can be assessed within an ASP-DDL framework using tools such as Clingo [27]. However, this notion is limited, as argument strength also depends on rule prioritization and contextual adequacy. Therefore, additional legal knowledge (case facts, statutes, and precedents) is required, motivating the use of a RAG-like framework to incorporate external context.
Challenges mentioned in Section 2.3, regarding the methods for extracting relative arguments strength in different legal contexts, as well as the choice of a reasoning approach for evaluating the final strength of arguments, remain to be investigated.

4 Results and analysis

In this section, we present the results achieved in the first steps of this PhD research project, in particular concerning the first two modules.

4.1 Argument mining

The results of the experiments on the short and long texts that we mentioned in Section 3.1, including the results

for the long texts with Saul that were missing in [12], are presented in the first two rows of Table 2. Overall, LLama outperforms Saul across all evaluation metrics. This difference may be partially attributed to model capacity, as LLama (8B parameters) is larger than Saul (7B).

Introducing agreement-based filtering consistently improves performance across all metrics. In particular, all agreement thresholds (0.6, 0.7, and 0.8) yield higher scores than the work of Berghegger et al. [12] which shows an improvement in the methodology. The effect of the threshold varies depending on the metric: a threshold of 0.6 slightly improves overall similarity (column *Sim*), 0.7 yields the highest granular similarity (column *Gran*), and 0.8 achieves the highest matching ratio (column *Match*). When averaging across metrics, the 0.8 threshold obtains the best overall mean score (column *Mean*).

We also evaluated the agreement-based approach on six additional ECHR texts, all of which are relatively short (*i.e.*, containing fewer than ten arguments). These new experiments showing an increase in the matching ratio for these texts, while overall similarity decreases slightly. This suggests that agreement may favor the recovery arguments maintaining the similarity. Interestingly, for these shorter documents Saul achieves better results in granular similarity than LLama, suggesting that future work could explore the agreement mechanism using Saul as the base model.

Importantly, we prioritize improvements in matching ratio over similarity. While similarity measures capture general semantic closeness, the matching ratio more directly reflects the recovery of relevant argumentative components. Therefore, the threshold of 0.8 appears to provide the most desirable balance between structural precision and content coverage.

4.2 Arguments to Logic

For our preliminary experiments, we selected Documents 30 and 41 from the ECHR corpus, as both contain only five arguments. This controlled setting allowed us to compare annotation strategies and identify an appropriate methodology before scaling to the full corpus.

As shown in Figure 4, differences between annotators can be partly explained by their backgrounds: Translator 1 (T1: computer science and law) incorporates more implicit structure and domain knowledge, whereas Translator 2 (T2: mathematics) adopts a more minimal formalization.

As illustrated in Figure 4, the annotators produce different translations of the same argument, which can be partly explained by their respective backgrounds (Section 3.2). Translator 1 (T1), with training in law, makes implicit normative reasoning explicit. In this example, T1 interprets the text as expressing a general requirement of independence and impartiality, together with a special attention in contexts involving specific vulnerabilities such as PTSD. This leads to the introduction of two rules (a general one and a context-sensitive one) and their prioritization via a superiority relation, as well as the representation of lack of independence and impartiality.

> *The applicant submits that, inter alia, the above factors demonstrate a lack, or at least a perceived lack, of independence and impartiality particularly when, as in his case, an important policy issue in respect of Post Traumatic Stress Disorder (PTSD) arose for consideration.*

T1

```
prescriptiveRule(x, court-martial(independence)).
prescriptiveRule(x, court-martial(imparciality)).
prescriptiveRule(y, court-martial(reinforce_imparciality)).
prescriptiveRule(y, court-martial(reinforce_independence)).
superior(y, x) :- important_policy_issue(ptsd).
non(court-martial(independence)).
non(court-martial(imparciality)).
```

T2

```
defeasible(non(independence)).
defeasible(non(impartiality)).
```

Gold

```
prescriptiveRule(x, imparciality).
prescriptiveRule(x, independence).
prescriptiveRule(y, reinforce_imparciality).
prescriptiveRule(y, reinforce_independence).

fact(important_policy_issue(ptsd)).
superior(y, x) :- important_policy_issue(ptsd).
defeasible(non(independence)).
defeasible(non(impartiality)).
```

LLama

```
fact(applicant_submits_factors_demonstrate
     _lack_of_independence),
fact(applicant_submits_factors_demonstrate
     _lack_of_impartiality).
```

GPT

```
fact(important_policy_issue(post_traumatic_
     stress_disorder)).
defeasible(non(independence_impartiality(court_martial))).
```

Figure 4: Example of NL → ASP-DDL translations generated by T1 and T2, compared against the Gold standard, along with the automatic translations produced by GPT and Llama.

By contrast, Translator 2 (T2), with a mathematical background, focuses strictly on explicitly stated content. As the text does not directly express rules or prioritization, T2 encodes only the observable propositions, treating independence and impartiality as defeasible facts and omitting contextual elements such as PTSD, which are not directly operational in the argument structure. The Gold standard aims to balance these approaches by preserving as much relevant information as possible while avoiding unsupported assumptions. In particular, elements such as `court-martial` are excluded, as they cannot be inferred from the argument alone, whereas implicit reasoning structures are retained when sufficiently grounded in the text.

The second part of Figure 4 shows the translations produced by LLama and GPT. Both models struggle to infer implicit implications, tending instead to extract explicitly stated information, similarly to Translator 2. This behavior is more pronounced in LLama, which only

	Model	Document 30		Document 41	
		LogicSim	BERTScore	LogicSim	BERTScore
T1	GPT	0.682	<u>0.858</u>	0.747	<u>0.879</u>
	Llama	**0.828**	0.831	0.736	0.850
T2	GPT	0.591	0.825	<u>0.781</u>	0.866
	Llama	0.719	0.813	**0.791**	0.832
Gold	GPT	0.708	**0.865**	0.714	**0.881**
	Llama	<u>0.809</u>	0.833	0.721	0.843

Table 3: NL→ASP-DDL translation results for Documents 30 and 41, evaluated using LogicSim and BERTScore. *T1* and *T2* denote the two translators, and *Gold* the gold standard. Bold indicates the highest score per metric among T1, T2, and Gold (across GPT and Llama), while underlined values denote the second-best score.

captures the lack of independence and impartiality as factual statements. GPT, in contrast, recovers additional contextual information by referencing PTSD and encodes uncertainty through a defeasible statement. Nevertheless, neither model captures the underlying rule structure, as implications between factors are systematically collapsed into facts.

These observations highlight that the main discrepancies lie at the structural level, particularly in the failure to represent implications. This is further reflected in the evaluation results (Table 3): while BERTScore and LogicSim indicate that the models recover key semantic content, they do not adequately penalize structural or syntactic deviations in ASP-DDL. This claim is based on the structural characteristics of the outputs for Texts 30 and 41, which exhibit patterns similar to those in Figure 4.

This limitation points to two necessary improvements. First, LogicSim should be extended to explicitly account for predefined DDL predicates, rather than evaluating only predicate overlap and logical components. Second, evaluation should include reasoning evaluation, verifying whether the generated ASP-DDL programs produce derivations comparable to those of the Gold standard via syntactic tree construction and subsequent graph similarity analysis and comparing the answer sets adapting metrics like Jaccard. Such an approach would allow us to assess not only textual and structural similarity, but also functional equivalence at the reasoning level.

5 Conclusions

This work introduces a novel modular pipeline for argument analysis in legal texts, offering significant advancements over existing approaches in several key ways. First, it addresses the challenge of limited resources by providing a new method for (semi-)automatic annotation, paving the way to the creation of new, diverse annotated datasets for argument mining. It also provides new annotated data for NL-Logic translation in the legal domain. Furthermore this work explores the connection between LLM-based extraction and formal reasoning through structured prompts, alignment strategies, and argument schemes to test how LLMs can generate coherent and logical representations. This aspect of the study directly addresses the question of how far LLMs can be effectively be applied in law, showing that, even with limitations, they can still provide meaningful support to legal professionals in improving the clarity and transparency of legal arguments.

Finally, the datasets, guidelines, and prompts produced in this study lay the foundation for training future domain-specific models that will be better equipped to perform accurate, explainable, and interpretable legal reasoning.

At this stage, we have presented only the preliminary results of the complete pipeline. As future work, we plan to complete the experimental evaluation following the methodology of [12] and to extend the agreement analysis to additional large language models, such as Qwen (Qwen/Qwen2.5-7B-Instruct). We also intend to incorporate lexical and semantic features to further improve performance in granular similarity metric. Regarding the NL → ASP-DDL translation, we aim to complete the translation of the datasets and conduct a systematic analysis of the correspondence between the natural language texts and their formal semantic representations, in order to establish a gold standard. Based on these results, we will refine the prompting strategies for automatic translation. If performance gains remain limited, we plan to explore the integration of reinforcement learning techniques to further enhance the models' reasoning and translation capabilities.

In conclusion, this work not only offers valuable new resources but also contributes a comprehensive and extensible framework for computational legal analysis, advancing the development of more transparent, reliable, and semantically grounded AI systems for the legal domain.

6 Acknowledgements

This work was funded by the French National Research Agency (grant AIDAL, ANR-22-CPJ1-0061-01).

Experiments presented in this paper were carried out using the OCCIDATA platform that is administered by IRIT and supported by CNRS (https://occidata.irit.fr).

References

[1] Lasha Abzianidze. LangPro: Natural language theorem prover. In *Proc. of EMNLP*, pages 115–120, 2017.

[2] AI@Meta. Llama 3 model card, 2024.

[3] Abdullah Al Zubaer, Michael Granitzer, and Jelena Mitrović. Performance analysis of large language models in the domain of legal argument mining. *Frontiers in Artificial Intelligence*, Volume 6 - 2023, 2023.

[4] Leila Amgoud and Jonathan Ben-Naim. Ranking-based semantics for argumentation frameworks. In *Proc. of SUM*, pages 134–147, 2013.

[5] Leila Amgoud, Jonathan Ben-Naim, Dragan Doder, and Srdjan Vesic. Acceptability semantics for weighted argumentation frameworks. In *Proc. of IJCAI*, pages 56–62, 2017.

[6] Leila Amgoud and Victor David. Measuring similarity between logical arguments. In *Proc. of KR*, pages 98–107, 2018.

[7] Katie Atkinson and Trevor J. M. Bench-Capon. Argumentation schemes in AI and law. *Argument Comput.*, 12(3):417–434, 2021.

[8] Katie Atkinson and Trevor J. M. Bench-Capon. Value-based argumentation. *FLAP*, 8(6):1543–1588, 2021.

[9] Katie Atkinson and Trevor J. M. Bench-Capon. ANGELIC II: an improved methodology for representing legal domain knowledge. In *Proc. of ICAIL*, pages 12–21, 2023.

[10] Mathieu Beirlaen, Jesse Heyninck, Pere Pardo, and Christian Straßer. Argument strength in formal argumentation. *FLAP*, 5(3):629–676, 2018.

[11] Trevor J. M. Bench-Capon and Sanjay Modgil. Norms and value based reasoning: justifying compliance and violation. *Artif. Intell. Law*, 25(1):29–64, 2017.

[12] Christina Berghegger, César Philippe, Karla Salas-Jimenez, Jean-Guy Mailly, Leila Moudjari, and Laurent Perrussel. Discovering the Potential of LLMs in Annotating Legal Texts for Argument Mining. In *Proc. of Arg&App*, 2025.

[13] Elise Bonzon, Jérôme Delobelle, Sébastien Konieczny, and Nicolas Maudet. A parametrized ranking-based semantics compatible with persuasion principles. *Argument Comput.*, 12(1):49–85, 2021.

[14] Johan Bos and Katja Markert. Recognising textual entailment with logical inference. In *Proc. of HLT/EMNLP*, pages 628–635, 2005.

[15] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran. Abstract dialectical frameworks. an overview. *FLAP*, 4(8), 2017.

[16] Elena Cabrio and Serena Villata. Five years of argument mining: a data-driven analysis. In *Proc. of IJCAI*, pages 5427–5433, 2018.

[17] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Graduality in argumentation. *J. Artif. Intell. Res.*, 23:245–297, 2005.

[18] Joe Collenette, Katie Atkinson, and Trevor J. M. Bench-Capon. Explainable AI tools for legal reasoning about cases: A study on the european court of human rights. *Artif. Intell.*, 317:103861, 2023.

[19] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. Saullm-7b: A pioneering large language model for law, 2024.

[20] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001.

[21] Victor David, Jérôme Delobelle, and Jean-Guy Mailly. Similarity measures for first-order logical arguments. In *Proc. of NMR*, pages 46–59, 2025.

[22] Adrian de Wynter and Tangming Yuan. "I'd like to have an argument, please": Argumentative reasoning in large language models. In *Proc. of COMMA*, pages 73–84, 2024.

[23] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[24] Thomas Eiter, Wolfgang Faber, Michael Fink, and Stefan Woltran. Complexity results for answer set programming with bounded predicate arities and implications. *Ann. Math. Artif. Intell.*, 51(2-4):123–165, 2007.

[25] Elliot Faugier, Frédéric Armetta, Angela Bonifati, and Bruno Yun. Assisted debate builder with large language models. In *Proc. of ECAI*, pages 4447–4450, 2024.

[26] Francisco Fernando Lopez-Ponce and Gemma Bel-Enguix. Into the limits of logic: Alignment methods for formal logical reasoning. In *Proc. of MathNLP*, pages 112–123, 2025.

[27] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot ASP solving with clingo. *Theory Pract. Log. Program.*, 19(1):27–82, 2019.

[28] Thomas F. Gordon, Henry Prakken, and Douglas Walton. The carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10):875–896, 2007. Argumentation in Artificial Intelligence.

[29] Thomas F Gordon and Douglas Walton. Legal reasoning with argumentation schemes. In *Proc. of ICAIL*, pages 137–146, 2009.

[30] Deniz Gorur, Antonio Rago, and Francesca Toni. Can large language models perform relation-based argument mining? In *Proc. of COLING*, pages 8518–8534, 2025.

[31] Guido Governatori. Practical normative reasoning with defeasible deontic logic. In *RW Summer School*, pages 1–25. Springer, 2018.

[32] Guido Governatori. An asp implementation of defeasible deontic logic. *KI-Künstliche Intelligenz*, 38(1):79–88, 2024.

[33] Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Simone Scannapieco. Computing strong and weak permissions in defeasible logic. *J. Philos. Log.*, 42(6):799–829, 2013.

[34] Gopal Gupta, Sarat Varnasi, Kinjal Basu, Zhuo Chen, Elmer Salazar, Farhad Shakerin, Serdar Erbatur, Fang Li, Huaduo Wang, Joaquín Arias, et al. Formalizing informal logic and natural language deductivism. In *ICLP Workshops*, 2021.

[35] Jesse Heyninck, Kenneth Skiba, and Matthias Thimm. Preface for the special issue on argument strength. *Argument Comput.*, 14(3):245–246, 2023.

[36] Elias Horner, Cristinel Mateis, Guido Governatori, and Agata Ciabattoni. From legal texts to defeasible deontic logic via llms: A study in automated semantic analysis. In *Proc. of ASAIL@ICAIL*, pages 83–100.

[37] Yuntao Kong, Ye Xiong, Shuyuan Zheng, and Ken Satoh. Reinforcement learning with argument-structured reward for court decision abstractive summarization. In *Proc. of JURIX*, pages 312–317, 2025.

[38] Mirko Lenz and Ralph Bergmann. User-centric argument mining with arguemapper and arguebuf. In *Proc. of COMMA*, pages 367–368, 2022.

[39] Fabrizio Macagno, Douglas Walton, and Chris Reed. *Argumentation Schemes*. Cambridge University Press, 2018.

[40] Raquel Mochales and Marie-Francine Moens. Study on the structure of argumentation in case law. In *Proc. of JURIX*, pages 11–20, 2008.

[41] Jack Mumford, Katie Atkinson, and Trevor J. M. Bench-Capon. Annotated insights into legal reasoning: A dataset of article 6 ECHR cases. *Argument Comput.*, 15(2):113–119, 2024.

[42] Magdalena Obermaier and Thomas Koch. The paradox of argument strength: how weak arguments undermine the persuasive effects of strong arguments. *Scientific Reports*, 14(1):22244, 2024.

[43] Gabriella Pigozzi and Srdjan Vesic. Preface for the special issue on argument strength. *Argument Comput.*, 12(1):1–2, 2021.

[44] Prakash Poudyal, Jaromir Savelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. ECHR: Legal corpus for argument mining. In *Proc. of ArgMining*, pages 67–75, 2020.

[45] Amin Rabinia and Sepideh Ghanavati. The FOL-based legal-grl (FLG) framework: Towards an automated goal modeling approach for regulations. In *Proc. of MoDRE@RE*, pages 58–67, 2018.

[46] Chris Reed, Raquel Mochales Palau, Glenn Rowe, and Marie-Francine Moens. Language resources for studying argument. In *Proc. of LREC*, pages 2613–2618, 2008.

[47] Livio Robaldo, Sotiris Batsakis, Roberta Calegari, Francesco Calimeri, Megumi Fujita, Guido Governatori, Maria Concetta Morelli, Francesco Pacenza, Giuseppe Pisano, Ken Satoh, et al. Compliance checking on first-order knowledge with conflicting and compensatory norms: a comparison among currently available technologies. *Artificial Intelligence and Law*, 32(2):505–555, 2024.

[48] Julien Rossit, Jean-Guy Mailly, Yannis Dimopoulos, and Pavlos Moraitis. United we stand: Accruals in strength-based argumentation. *Argument Comput.*, 12(1):87–113, 2021.

[49] Ramon Ruiz-Dolz, Joaquín Taverner, John Lawrence, and Chris Reed. NLAS-multi: A multilingual corpus of automatically generated natural language argumentation schemes. *CoRR*, abs/2402.14458, 2024.

[50] Jeroen Paul Spaans. Intrinsic argument strength in structured argumentation: A principled approach. In *Proc. of CLAR*, pages 377–396, 2021.

[51] Guilherme Trajano, Débora C Engelmann, Rafel H Bordini, Stefan Sarkadi, Jack Mumford, and Alison R Panisson. Translating natural language arguments to computational arguments using LLMs. In *Proc. of COMMA*, pages 289–300, 2024.

[52] Elfia Bezou Vrakatseli, Oana Cocarascu, and Sanjay Modgil. EthiX: A dataset for argument scheme classification in ethical debates. In *Proc. of ECAI*, 2024.

[53] Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. Harnessing the power of large language models for natural language to first-order logic translation. In *Proc. of ACL*, pages 6942–6959, 2024.

[54] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *Proc. of ICLR*, 2020.

[55] Abdullah Al Zubaer, Michael Granitzer, and Jelena Mitrovic. Performance analysis of large language models in the domain of legal argument mining. *Frontiers Artif. Intell.*, 6, 2023.

Session 2 : Apprentissage par renforcement & Multi-agents

Vers un contrôle par apprentissage par renforcement de l'alimentation électrique dans un avion hybride

Aubin Delaveau[1,2], Olivier Buffet[2] Florent Teichteil-Königsbuch[1],
[1] Airbus Central Research & Technology, F-31000, Toulouse
[2] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy

prenom.nom@airbus.com, prenom.nom@loria.fr

Résumé

*Nous nous intéressons à la gestion et l'optimisation de la consommation électrique dans les avions hybrides. Face aux dynamiques non-linéaires coûteuses à simuler, les approches de contrôle par correcteur PID (proportionnel, intégral, dérivé) et MPC (*model predictive control*) ne sont pas adaptées aux systèmes exigeant une grande réactivité. Nous proposons ici une formulation du problème comme un processus de décision markovien (MDP), présentons des résultats expérimentaux préliminaires obtenus avec une approche myope, et discutons des pistes envisageables dans le cadre de l'apprentissage par renforcement.*

Mots-clés

Avion hybride, apprentissage par renforcement, stratégie myope.

1 Introduction

La gestion optimale de la consommation d'énergie est un objectif technologique majeur pour le développement des avions hybrides pour lesquels l'énergie électrique est produite à la fois par des batteries et les moteurs. Elle doit répondre à la variabilité des phases de vol (décollage, vol de croisière, atterrissage) et des états internes de l'avion.
Cependant, le contrôle du système électrique est régi par des équations différentielles ordinaires (EDO) non linéaires. Ces dynamiques rendent l'usage des méthodes de contrôle classiques, basées sur le calcul de gradients, particulièrement ardues, voire impraticables en raison de la nature "boîte noire" de la simulation et du bruit numérique inhérent à la résolution des EDO.

Travaux antérieurs Le contrôle de systèmes électriques repose classiquement sur des correcteurs PID (proportionnel, intégral, dérivé) ou des méthodes de contrôle prédictif (MPC). Les correcteurs PID souffrent d'un manque de méthode de paramétrage générique face aux fortes non-linéarités des systèmes hybrides d'énergie qui nous intéressent. Les approches MPC standard, elles, bien que performantes pour gérer des contraintes explicites, nécessitent généralement une linéarisation locale du modèle dynamique. Cette simplification entraîne souvent une perte de précision, tandis que la résolution directe du problème non linéaire est trop coûteuse pour du temps réel.

Face à ces limites, nous proposons une formalisation comme un processus de décision markovien (MDP), l'évolution de l'état du système se faisant à temps discret en simulant numériquement l'EDO sous-jacente. On va ainsi pouvoir considérer des approches de résolution "boîte noire" ne reposant pas sur des calculs de dérivés.

Plan La section suivante présente le formalisme MDP. Nous formalisons ensuite notre scénario comme un MDP et proposons une première stratégie de contrôle myope, c'est-à-dire optimale sur un pas de temps. De premiers résultats expérimentaux sont enfin présentés avant de discuter des limitations de notre approche et des pistes envisagées.

2 Processus de décision markoviens

Le cadre des processus de décision markoviens (MDP) fournit une formalisation mathématique standard pour les problèmes de contrôle séquentiel [3]. Un MDP, ici déterministe, est défini par un quadruplet $\langle S, A, T, R\rangle$ où S est l'espace des *états*; A est celui des *actions* possibles; $s' = T(s,a)$, la *fonction de transition*, indique l'état s' atteint quand l'action a est effectuée dans l'état s; et $R(s,a,s')$, la *fonction de récompense*, associe une récompense scalaire immédiate à chaque transition.
La stratégie de contrôle est dictée par une *politique* $\pi : S \to A$, qui associe à chaque état une action. L'objectif est de trouver une politique optimale π^*, c'est-à-dire maximisant en tout s la fonction de valeur $V_\pi(s)$, définie comme l'espérance de la somme cumulée des récompenses actualisées : $V_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s\right]$, où $\gamma \in [0,1[$, le facteur d'actualisation, permet de pondérer l'importance des récompenses futures. Ce formalisme permet d'aborder le problème de la commande optimale comme une recherche de politique π dans un espace continu, ici résolue par optimisation sans dérivée.

3 Approche proposée

Notre scénario Le système étudié est un réseau électrique d'avion hybride composé de deux générateurs associés aux moteurs (HP et LP), d'une batterie, et d'une charge utile (CPL).
La dynamique du système est gouvernée par un système d'équations différentielles ordinaires (EDO) non linéaires : $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, P_{CPL}(t))$, où $\mathbf{x}$ représente le vecteur des va-

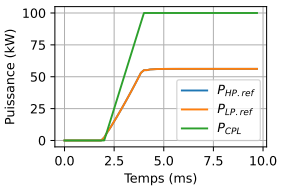
riables d'état internes (ex : tensions des bus, courant batterie) ; $\mathbf{u}$ est le vecteur des variables de contrôle (puissances fournies par les générateurs) ; et $P_{CPL}(t)$ est la puissance consommée par la charge, entrée exogène connue a priori.

Formalisation MDP Le système est modélisé comme un MDP comme suit : on discrétise le temps, en supposant ici un pas $\delta t = 1$ pour pouvoir noter t le temps dans le modèle physique comme dans le modèle MDP ; l'**état** est défini par le vecteur $s_t = [\mathbf{x}_t, t]^T$, où $\mathbf{x}_t$ représente les variables internes de l'avion et t l'instant courant ; l'**action** $a_t \in A$ correspond aux puissances de commande $\mathbf{u}_t$ appliquées aux générateurs pendant δt, en imposant un certain ratio entre ces puissances ; la **transition** $s_{t+1} = T(s_t, a_t)$ s'obtient en résolvant une équation aux dérivées ordinaires (EDO) sur un pas de temps ; pour ici minimiser l'emploi de la batterie (donc le courant la traversant), la **récompense** instantanée correspond à $-\|i_{bat}(s_{t+1})\|^2$. On notera que la dynamique de P_{CPL} est une entrée exogène connue (déterministe), indépendante de l'action a_t, et intégrée dans l'EDO.

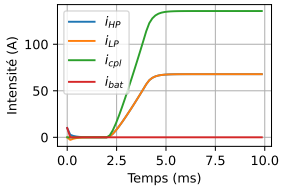
Algorithme de contrôle Les espaces d'états et d'actions étant continus, un tel MDP serait typiquement abordé par un algorithme d'apprentissage par renforcement profond tel que *proximal policy optimization* [6]. Dans un premier temps, afin d'obtenir rapidement une première solution pragmatique à notre problème, nous nous limitons à mettre en œuvre une stratégie myope. À chaque pas de temps, étant donné s_t, on cherche l'action a_t maximisant la récompense immédiate. On doit donc optimiser une fonction $J(a) \stackrel{\text{def}}{=} -\|I_{bat}(T(s_t, a))\|^2$ non dérivable, mais simulable par l'EDO Φ, dans un espace continu. Nous employons à cette fin l'algorithme COBYQA (*Constrained Optimization BY Quadratic Approximation*) [5].

4 Résultats expérimentaux

L'algorithme de contrôle proposé a été évalué sur un premier circuit type, permettant de satisfaire la demande de la charge tout en assurant la stabilité du système. La figure 1a illustre la répartition des puissances entre les générateurs (courbes $P_{HP.ref}$ et $P_{LP.ref}$ superposées), démontrant la capacité du solveur à gérer la dynamique non linéaire. L'intensité batterie i_{bat} est maintenue proche de zéro (figure 1b, i_{HP} et i_{HP} superposées), confirmant que les générateurs couvrent la demande de la charge.
Réduire le pas de temps conduit toutefois à de fortes oscillations de i_{bat}, illustrant les limites d'un contrôle myope dans certaines situations.



(a) Puissances de commande



(b) Intensités sources et charge

FIGURE 1 – Évolution de diverses grandeurs. $P_{HP.ref}$ et $P_{LP.ref}$ sont confondues. De même pour i_{HP} et i_{LP}.

5 Discussion et Perspectives

Discussion La stratégie proposée permet d'atteindre des trajectoires de contrôle stables avec une intensité batterie maintenue proche de zéro dans des situations simples. Toutefois, cette approche comporte des verrous techniques : 1. une vision "myope" qui ne garantit pas une optimalité globale sur l'horizon temporel, en particulier pour des dynamiques rapides de $P_{CPL}(t)$, 2. une latence computationnelle élevée (30–60 s) incompatible avec le temps réel, et 3. une dépendance à une connaissance immédiate des états.

Perspectives Pour lever ces verrous, nous envisageons plusieurs axes de recherche dans le contexte de l'apprentissage par renforcement : se tourner vers le *Deep RL* en espaces d'états et d'actions continus (par ex. avec PPO [6]), et éventuellement guider un *apprentissage par imitation* avec notre stratégie myope [4], [7] ; exploiter les *Physics-Informed Neural Networks* [2] pour intégrer la connaissance du modèle physique dans l'apprentissage ; intégrer la présence de *délais d'observation* dans la boucle de contrôle [1].
Cette approche hybride, combinant la robustesse de l'optimisation sans dérivée et la capacité de généralisation des modèles neuronaux, constitue une voie prometteuse pour le contrôle embarqué haute performance des avions hybrides.

Références

[1] M. AGARWAL et V. AGGARWAL, "Blind decision making : Reinforcement learning with delayed observations," *Pattern Recognition Letters*, 2021.

[2] C. BANERJEE, K. NGUYEN, C. FOOKES et M. RAISSI, "A survey on physics informed reinforcement learning : Review and open problems," *Expert Systems with Applications*, 2025.

[3] R. BELLMAN, "A Markovian Decision Process," *Journal of Mathematics and Mechanics*, 1957.

[4] G. LIBARDI, G. D. FABRITIIS et S. DITTERT, "Guided Exploration with Proximal Policy Optimization using a Single Demonstration," in *Proc. of ICML*, 2021.

[5] T. M. RAGONNEAU, "Model-based derivative-free optimization methods and software," thèse de doct., Hong Kong Polytechnic University, 2023.

[6] J. SCHULMAN, F. WOLSKI, P. DHARIWAL, A. RADFORD et O. KLIMOV, "Proximal policy optimization algorithms," *arXiv :1707.06347*, 2017.

[7] M. ZARE, P. M. KEBRIA, A. KHOSRAVI et S. NAHAVANDI, "A survey of imitation learning : Algorithms, recent developments, and challenges," *IEEE Transactions on Cybernetics*, 2024.

Modèles pour la planification multi-agents de tâches complexes

Jules Dubanet[1], Josselin Guéneron[1]

[1] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

prenom.nom@unicaen.fr

Résumé

Nous proposons dans cet article deux nouveaux modèles pour la planification multi-agents de tâches simples et complexes. Le premier modèle s'appuie sur une exploration gloutonne de l'espace d'états, tandis que le deuxième modèle intègre un processus de négociation. Nous avons ensuite comparé nos deux modèles à un processus décisionnel markovien décentralisé partiellement observable résolu à l'aide de l'algorithme Multi-Agent Value Iteration. Nous avons principalement montré que nos modèles atteignent une optimalité comparable avec une résolution plus rapide.

Mots-clés

Systèmes multi-agents, Négociation, Allocation de tâches, Planification

Abstract

In this article, we propose two new models for multi-agent planning of simple and complex tasks. The first model is based on greedy exploration of the state space, while the second model incorporates a negotiation process. We then compared our two models to a decentralized partially observable Markov decision process solved using the Multi-Agent Value Iteration algorithm. We mainly showed that our models achieve comparable optimality with faster resolution.

Keywords

Multi-agent systems, negociation, task allocation, planning

1 Introduction

Les agents autonomes, à l'instar des drones sont utilisés de manière croissante dans de nombreux domaines comme le militaire, la logistique, les réseaux électriques intelligents ou encore la protection civile [3]. La gestion des interactions entre ceux-ci présente un véritable défi que l'étude des systèmes multi-agents tente de relever, en particulier lorsque les agents doivent se coordonner et coopérer [8]. À titre d'exemple, une des applications critiques des systèmes multi-agents coopératifs est le *disaster response* (gestion de catastrophe). Dans un problème de *disaster response*, un ensemble d'agents autonomes, parfois hétérogènes dans leurs compétences et dans leur nature (UAV, UAG, parfois humains), doit coopérer et se coordonner afin de répondre à une catastrophe, telle que des inondations ou un séisme. Ce problème se déroule donc dans un environnement dangereux et incertain en raison du caractère imprévisible des catastrophes et de leurs conséquences. Par conséquent, les tâches à réaliser nécessitent souvent plusieurs compétences, les rendant complexes.

Plusieurs approches existantes permettent de formaliser ces systèmes multi-agents. Parmi elles nous retrouvons les processus de décision markoviens (MDP) et leurs extensions multi-agents [1]. Ces processus présentent de nombreux avantages, ils sont facilement modélisables et gèrent très bien la stochasticité de l'environnement. Cependant, ils sont limités dans la représentation de tâches complexes nécessitant la coopération de plusieurs agents.

Nous pouvons également évoquer le formalisme des formations de coalitions qui propose de regrouper les agents en sous-groupes nommés "coalitions", leur permettant ainsi de mettre en commun leurs compétences afin de coopérer pour effectuer des tâches complexes [5]. Ce formalisme est donc tout à fait adapté aux environnements complexes. Il existe déjà des méthodes de planification basées sur la formation de coalitions, mais ces travaux ne prennent en compte aucune stochasticité, les tâches étant supposées parfaitement réalisables et les gains obtenus par les agents certains. Or, si l'on s'intéresse au contexte de la *disaster response*, l'environnement est rempli d'incertitudes et il est donc nécessaire de pouvoir proposer des modèles capables de traiter des tâches complexes dont la réalisation est stochastique, et ce, de manière répétée. De plus, la formation de coalitions est souvent coûteuse à résoudre en termes de temps et mémoire.

En prenant en compte ces restrictions, le domaine de l'*allocation de tâches multi-agents* (MRTA, *multi-robot task allocation*) semble prometteur. En effet, la gestion de catastrophes implique de devoir assigner dynamiquement des tâches critiques à des agents aux compétences variées, en tenant compte à la fois des incertitudes liées à l'environnement et des contraintes de coopération, ce qui correspond très bien à ce qui est réalisable grâce à la MRTA. Les approches existantes présentent donc des limites : les modèles de MDP ne prennent pas en compte des tâches complexes, tandis que les modèles de formation de coalitions peinent à intégrer de l'incertitude [6] et sont coûteux. L'enjeu est donc de développer un modèle où les agents peuvent planifier la réalisation de tâches complexes, en possible coopération, dans un contexte distribué ou décentralisé. Dans cet article, nous nous intéressons à cette notamment à dernière piste en nous basant sur la MRTA, et nous présenterons des

pistes d'ouvertures vers l'intégration du dynamisme et de la stochasticité dans nos modèles.

Cet article est structuré comme suit. Dans la section 2, nous présenterons les différents formalismes de la littérature sur lesquels nous nous appuyerons, avant de présenter nos contributions en section 3. Des expérimentations seront présentées en section 4 avant de conclure et donner des perspectives en section 5.

2 État de l'art

Dans cette section nous allons présenter les éléments de littérature utiles à la compréhension des modèles développés, à savoir l'allocation de tâches multi-robots, les processus de décision markovien, et leur extension décentralisée et partiellement observable qui nous serviront d'élément de comparaison.

2.1 Allocation de tâches

Dans le cadre des systèmes multi-agents, l' *allocation de tâches* est une catégorie de problèmes dans lesquels des agents doivent réaliser des tâches au sein d'un environnement. Le but de ces problèmes est de faire émerger entre les agents de la coordination voire de la coopération. Cette coordination peut être implicite (aussi appelée émergente) quand elle résulte d'une somme d'interactions locales entre les agents et l'environnement, mais elle peut aussi être explicite (ou intentionnelle) quand les agents sont explicitement assignés à certaines tâches. Ces problèmes peuvent être catégorisés selon 3 axes [4] : le type de tâche, le type d'agent, le type d'allocation. Pour chacun de ces axes, il y a 2 choix possibles. Respectivement il y a :

— **Single-robot Task (ST)** ou **Multi-robot Task (MT)**, Single-robot Task signifie que les tâches n'ont besoin que d'un seul agent pour être complétées alors que les Multi-robot Task nécessitent plusieurs agents ;
— **Single-task Robot (SR)** ou **Multi-task Robot (MR)**, Single-task Robot signifie que les agents ne peuvent exécuter qu'une seule tâche à la fois tandis que Multi-task Robot signifie que certains agents peuvent faire plusieurs tâches à la fois ;
— **Instantaneous assignement (IA)** ou **Time-extended assignement (TA)**, Instantaneous assignement signifie que le modèle ne permet que l'allocation instantanée des agents sans prévisions sur de futures allocations, tandis que dans le cas Time-extended allocation, de la prévision est possible, par exemple en fournissant l'ordre d'arrivée de nouvelles tâches.

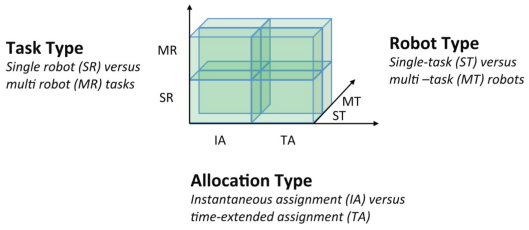


FIGURE 1 – Taxonomie de l'allocation de tâches [7]

La figure 1 résume graphiquement ces axes. Grâce à cette taxonomie, nous pouvons catégoriser des classes de problèmes, par exemple si nous qualifions un problème de ST-MR-IA, cela signifie que les tâches ne nécessitent qu'un agent, que les agents peuvent faire plusieurs tâches à la fois et que l'allocation se fait selon les informations dont nous disposons sur l'instant.
Un problème d'allocation de tâches multi-robots (MRTA) est donc un problème de coordination où un groupe d'agents doivent efficacement s'assigner à des tâches afin de les exécuter.

Définition 1 (Problème de MRTA) *Formellement, un problème de MRTA est défini par un tuple* $\langle R, T, u \rangle$ *tel que :*

— $R = r_1, r_2, ..., r_n$*, un ensemble de* n *robots,*
— $T = t_1, t_2, ..., t_m$*, un ensemble de* m *tâches,*
— $u : R \times T \to \mathbb{R}$*, une fonction d'utilité qui associe à chaque couple (robot, tâche) une valeur correspondant au gain du robot si celui-ci effectue la tâche.*

Nous notons donc u_{ij} *l'utilité gagnée (ou le coût infligé dans le cas d'une utilité négative) par le robot* r_i *réalisant la tâche* t_j*.*

Parmi les très nombreuses manières de résoudre les problèmes de *task allocation*, les algorithmes basés sur le principe d'enchères (*Auction based algorithms*) sont particulièrement intéressants [2]. Le principe général est que les agents peuvent miser sur les tâches à accomplir et s'ils remportent la mise, ils sont alloués à la tâche. Ces mises peuvent dépendre de plusieurs paramètres, un des plus courant étant la capacité de l'agent à effectuer la tâche, et plus il est compétent pour une tâche, plus il pourra enchérir dessus. Un autre paramètre commun est la distance à la tâche (pouvant être associée à un coût de déplacement). L'algorithme CBAA (*Consensus-Based Auction Algorithm*) se base sur ce principe. Il se déroule en 2 phases. Premièrement, les agents communiquent une enchère pour la tâche qui est la plus avantageuse de leur point de vue. Les agents mettent ensuite à jour une liste associant les tâches avec l'agent qui a formulé la meilleure enchère. Cette première phase est la phase d'enchères, la deuxième phase qui la suit est la phase de consensus. Les agents partagent leur liste avec leurs voisins et pour chaque tâche, ils gardent l'offre la plus compétitive. Ce processus est répété itérativement et converge vers un état où tous les agents ont la même liste.

2.2 Processus de Décision Markovien

Un processus de décision markovien (MDP) est un modèle stochastique de la théorie de la décision permettant de modéliser la prise de décision d'un agent dans un environnement incertain [9].

Définition 2 (Processus de Décision Markovien)
Un processus de décision markovien est un tuple $\mathcal{M} = \langle S, A, T, R \rangle$ *avec :*

- S *un ensemble d'états,*
- A *un ensemble d'actions réalisables par l'agent,*
- $T(s'|s, a)$ *une fonction de transition qui renvoie un état en fonction de l'état courant et de l'action effectuée par l'agent,*
- R *une fonction de récompense qui prend en entrée l'état courant et une action réalisée par l'agent* $R(s, a)$.

La résolution d'un MDP consiste à déterminer une *politique* optimale (*i.e.* maximisant l'espérance de gains).

Définition 3 (Politique) *Une* politique *décrit l'action à effectuer par l'agent dans chaque état possible du MDP. Mathématiquement, il s'agit d'une fonction* $\pi : S \rightarrow A$ *qui à chaque état associe une action. Une politique peut être déterministe* ($\pi(s) = a$) *ou stochastique* ($\pi(a|s) = P(a|s)$). *La politique optimale* $\pi*$ *est calculée à partir de l'équation de Bellman :*

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} [R(s, a) + \gamma V^*(s')] T(s, a, s')$$

avec $\gamma \in [0, 1]$ *un facteur d'atténuation.*

Les DEC-POMDP (*Decentralized Partially Observable MDP*) sont une extension des MDP permettant de considérer plusieurs agents évoluant dans un même environnement partiellement observable [1]. Les agents possèdent des croyances individuelles sur l'environnement, qu'ils affinent à partir de nouvelles observations, et calculent une politique individuelle à partir de leurs croyances.

Définition 4 (DEC-POMDP) *Un DEC-POMDP est un tuple*
$\langle S, \{A_1, .., An\}, T, R, \{\Omega_1, .., \Omega_n\}, \{O_1, ..O_n\} \rangle$ *avec :*

- S *un ensemble d'états,*
- $\{A_1, .., A_n\}$ *l'ensemble des actions de chaque agent,*
- $T(s'|s, a)$ *une fonction de transition,*
- R *une fonction de récompense,*
- $\{\Omega_1, .., \Omega_n\}$ *un espace d'observation pour chaque agent,*
- $\{O_1, .., O_n\}$ *une fonction d'observation propre à chaque agent.*

Ces agents agissent de manière décentralisée afin de calculer une politique jointe, c'est-à-dire une politique qui prend en compte les actions de l'ensemble des agents.

3 Contributions

Dans cet article, nous proposons deux modèles. Ces deux modèles partagent certaines caractéristiques : les agents agissent de manière coopérative, l'exécution des actions des agents se fait de manière cyclique, et les agents ont le moins d'informations possible. En effet dans le contexte du *disaster response*, les communications sont parfois limitées (les infrastructures de communication peuvent être endommagées, ou les services saturés). La coopération est essentielle car, d'une part tous nos agents poursuivent le même but, à savoir la complétion des tâches, et d'autre part, la plupart des situations réelles peuvent nécessiter diverses compétences qu'un seul agent ne peut pas toujours posséder. Le premier modèle prend en considération des tâches dites simples (c'est-à-dire qui ne nécessitent qu'une seule compétence pour être réalisées), et utilise une méthode d'exploration gloutonne. Pour ces raisons, nous appellerons ce premier modèle le modèle simple ou à exploration. Le deuxième modèle quant à lui prend en compte des tâches plus complexes (c'est-à-dire qui nécessitent une combinaison de compétences pour être réalisées) et utilise un processus de négociation. Nous l'appellerons donc le modèle complexe ou le modèle à négociation. En utilisant la taxonomie des modèles de *task allocation*, le modèle simple peut se noter ST-SR-IA et le modèle complexe MT-SR-IA. Bien que les contraintes de précédence entre tâches, le dynamisme et la stochasticité sont de fort intérêt dans le contexte du *disaster response*, les intégrer directement aux modèles aurait été complexe, c'est pourquoi nous nous sommes concentrés dans cet article sur des versions statiques, déterministes et sans contraintes de précédence.

3.1 Éléments communs aux deux modèles

Les deux modèles présentés dans cette section utilisent des mécanismes de décision différents mais sont fondés sur des caractéristiques et concepts de modélisation similaires. Dans les deux modèles, nous avons un ensemble d'agents qui ont pour objectif de réaliser des tâches afin de gagner des récompenses. Pour ce faire, ils peuvent s'associer aux tâches : l'association de tous les agents forme un état.

Définition 5 (Environnement des modèles)
L'environnement de nos modèles est défini par un tuple $\langle N, J, C, r \rangle$ *tel que :*

- $N = \{n_1, n_2, ..., n_l\}$ *: l'ensemble des agents,*
- $J = \{j_1, j_2, ..., j_m\}$ *: l'ensemble des tâches,*
- $C = \{c_1, c_2, ..., c_k\}$ *: l'ensemble des compétences, avec* $\forall i \in [1, k],\ c_i \in \mathbb{R}$
- $u : J \rightarrow \mathbb{R}$ *: la fonction d'utilité associant à chaque tâche une utilité réelle.*

Les agents du modèles peuvent s'associer à une tâche, nous pouvons donc définir :

- j_n *: la tâche associée à un agent* n,
- $N_j = \{n \mid n \in N\ t.q. j_n = j\}$ *: l'ensemble des agents associés à la tâche* j.

Définition 6 (Tâches) *Les tâches représentées dans les modèles doivent être réalisées par les agents. Elles pos-*

sèdent des besoins spécifiques :

$$B_j = \{c_1^j, c_2^j, ..., c_k^j\}$$

À leur réalisation, les tâches fournissent donc une utilité aux agents responsables de leur réalisation, cette utilité est notée u_j.

La fonction de récompense (différente de la fonction d'utilité) est spécifique à chaque modèle et sera donc présentée dans les parties dédiées aux modèles.

Définition 7 (Agent) *Chaque agent possède un certain nombre de compétences, notées :*

$$C_n = \{c_1^n, c_2^n, ..., c_k^n\}$$

Avec $C_n \subseteq C$. Ces compétences permettent de réaliser les tâches en fonction de leurs besoins.

Définition 8 (Action d'un agent) *L'action d'un agent n est définie comme le fait de se positionner sur une tâche. Les actions sont définies de la même manière dans les deux modèles. L'ensemble des tâches réalisables par l'agent a se définit comme :*

$$A_n = \{j \mid j \in J \; \exists c_i \in C, \; t.q. \; B_j(c_i) \leqslant C_n(c_i)\}$$

S'il existe au moins un besoin d'une compétence dans les besoins de j où l'agent n est suffisamment compétent, il pourra être alloué à j.

Un agent possède donc autant d'actions possibles que de tâches réalisables en adéquation avec son vecteur de compétences. Pour chaque agent, la taille de l'espace d'action maximal est $|J|$.

Définition 9 (État des modèles) *Dans les modèles, un état est défini par les ensembles d'agents associés à chaque tâche. Nous avons donc :*

$$s = \{j_1 : \{n_1, n_2...\}, .., j_m : \{n_{l-1}, n_l\}\}$$

L'espace d'états est constitué de l'ensemble des allocations entre tâches et agents. Nous pouvons également considérer un état comme étant une action jointe de tous les agents, pour cette raison l'espace d'états est le produit de l'espace d'actions de tous les agents.

Exemple 1

- *Soit $N = \{n_1\}$ un ensemble d'agents de taille 1,*
- *Soit $A_{n_1} = \{j_1, j_2, ..., j_m\}$ l'ensemble des tâches réalisables par n_1,*

Ici, il n'y a qu'un seul agent, l'espace d'états se résume donc à l'espace des tâches auxquelles il peut s'associer. Nous avons donc $|S| = |A_n| = m$.

Nous pouvons en déduire que pour un système avec l agents dont les actions sont bornées par J, l'espace d'états est $|J|^l$.

Définition 10 (Social Welfare) *Le social welfare, ou bien-être social, est une mesure globale de la satisfaction des agents, c'est une fonction qui prend en entrée les récompenses individuelles des agents et retourne un indicateur de la qualité du système. Ici, la fonction d'agrégation utilisée est la somme. Nous avons donc :*

$$SW(s) = \sum_{n \in N} R(s, n)$$

Cette fonction est commune aux deux modèles où seule la définition de $R(s, n)$ change. C'est également cette fonction qui est utilisée dans la formalisation sous forme de DEC-POMDP pour calculer la récompense de chaque état.

3.2 Modèle à exploration

Le premier modèle présenté est un modèle de *task allocation* de type ST-SR-TA, c'est-à-dire que les agents ne peuvent s'associer qu'à une seule tâche et que les tâches ne nécessitent qu'un agent pour être réalisées. Les agents utilisent une exploration et une évaluation gloutonne des états, par conséquent, ils vont explorer les états qu'ils sont en mesure d'atteindre selon leurs connaissances de l'environnement. Ils connaissent l'état actuel, c'est-à-dire l'allocation tâches-agents courante, ainsi que leurs propres compétences desquelles ils peuvent déduire les tâches sur lesquelles ils peuvent se positionner. L'exécution du modèle s'articule ainsi :

1. Les agents calculent une politique pour chaque état qu'ils considèrent atteignable et pour l'état actuel.
2. Les agents appliquent leur politique puis en calcule une nouvelle pour les nouveaux états atteignables.

Les états que les agents prennent en compte sont uniquement ceux où les autres agents ne bougent pas, puisque les agents ne considèrent que leurs propres compétences. De ce fait, les agents se retrouvent dans des états qu'ils n'avaient pas anticipés, étoffant ainsi leurs connaissances des états possibles. Le calcul d'une politique se fait en regardant tous les mouvements possibles à partir d'un état (c'est-à-dire envisager de se positionner sur toutes les tâches réalisables) et de garder l'action qui mène à l'état maximisant la récompense.

3.2.1 Fonction de récompense

Les récompenses sont calculées grâce à une fonction et permettent d'influencer le comportement des agents. Elle est définie comme suit :

$$R(s, n) = \frac{u_j c_j^n}{N_j} \prod_{j' \in A_n} [N_{j'} + \epsilon]$$

Avec :

- s un état,
- n un agent,
- j la tâche à laquelle l'agent n est associé,
- u_j l'utilité associée à la tâche j,
- c_j^n la compétence de n pour le besoin de j,
- N_j le nombre d'agents associés à j (ce nombre vaut toujours au moins 1 car quand un agent calcule la

récompense d'un état, il se projette dans celui-ci et calcule la récompense liée à l'association avec j, donc si personne n'était déjà associé à j, N_j vaudra 1),
— $\epsilon \in]0, 1[$ un terme évitant la présence de 0 dans le produit.

Le produit des $N_{j'} + \epsilon$ sert à favoriser les cas de répartition égale entre les tâches et à pénaliser les cas où une ou plusieurs tâches seraient laissées sans agents associés. Puisque ϵ est plus petit que 1, si un $N_{j'}$ est à 0, le produit final sera impacté et nullifié. La valeur maximale de ce produit est atteinte quand la répartition entre les tâches est parfaitement égale. Nous allons démontrer cette affirmation ci-dessous.

Démonstration 1 *Prenons un exemple : soient deux tâches, et $2x$ agents que nous répartissons sur les deux tâches. Nous souhaitons démontrer qu'en cas de répartition égale des agents, le produit du nombre d'agents associés à chaque tâche est supérieur au produit du nombre d'agents associés à chaque tâche en cas de répartition inégale. Nous avons donc deux situations :*

1. *Répartition inégale : respectivement $x - y$ et $x + y$ pour la première et deuxième tâche.*
2. *Répartition égale : x pour chaque tâche.*

Nous pouvons donc formuler une inéquation dont nous devons vérifier la cohérence, avec à gauche le terme représentant une répartition inégale, et à droite, une répartition égale :

$$(x - y)(x + y) \leq x \times x$$

Nous avons bien la somme des termes de chaque côté de l'équation qui est égale à $2x$ (notre nombre d'agents). Tentons de développer l'inéquation. Selon les identités remarquables, nous avons l'égalité suivante.

$$(a + b)(a - b) = a^2 - b^2$$

Nous pouvons donc développer le terme de gauche de notre inéquation.

$$x^2 - y^2 \leq x^2$$

L'inéquation est donc correcte. Elle est également généralisable à davantage de termes de façon similaire. Nous pouvons donc affirmer que lorsque le nombre de termes est égal, ainsi que leur somme, le produit des termes lorsque la répartition est égale est supérieur à celui obtenu avec une répartition inégale.

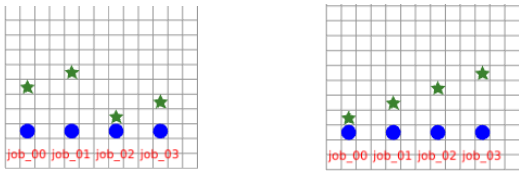
Ce produit se fait entre toutes les tâches visibles par a (noté A_a). Cette restriction permet de diminuer les calculs et ne change rien quant aux résultats.

Exemple 2 *Par exemple, prenons :*
— *n_1 un agent,*
— *j_1, j_2 et j_3 des tâches,*
— *$A_n = \{j_1, j_2\}$.*

L'agent fera en sorte que la répartition entre j_1 et j_2 soit égale, mais l'état de j_3 n'influencera pas sa décision. Si nous prenons en compte toutes les tâches, l'état de j_3 changerait la valeur de la récompense, mais pas la dynamique entre les états (à savoir est-ce qu'un état a une récompense plus élevée qu'un autre).

Avec cette fonction de récompense, les agents vont chercher à maximiser leur récompense individuelle, mais elle est maximale dans un état où les agents sont bien répartis, les agents vont donc prioriser ces états au lieu de tous se positionner sur la tâche ayant la récompense la plus élevée.

La récompense dépend en partie de l'expertise de l'agent pour la tâche, ainsi plus un agent est compétent pour une tâche, plus il va préférer celle-ci, comme il peut-être observé sur la figure 2.



(a) Compétences homogènes (b) Expertise des agents

FIGURE 2 – Exemple de répartition après exécution du modèle simple avec 4 agents et 4 tâches. Dans ces images, les agents (étoiles vertes) évoluent dans une grille et se placent au dessus des tâches (disques bleus) avec lesquelles ils s'associent. Leur positionnement en vertical est égal à leur indice + 1.

La figure 2a (gauche) présente une répartition des agents où leurs compétences sont homogènes et la figure 2b (droite) où ils sont experts pour la compétence qui a le même nom ou numéro qu'eux. Les agents sont placés de manière à ce que leur coordonnée (en ordonnée) est égale à l'ordonnée de la tâche additionnée à leur numéro (+1 car la numérotation commence à partir de 0, cela empêche donc les chevauchements). Nous constatons donc que les agents se positionnent sur les tâches dont ils sont experts. Il est important de noter que dans le cas où les compétences sont homogènes, toute configuration où il n'y a qu'un seul agent par tâche est optimale.

3.3 Modèle à négociation

Ce deuxième modèle permet de pallier une des limites du premier modèle : la non prise en compte de tâches plus complexes. Ces tâches sont dites complexes, car leurs besoins sont multiples et peuvent donc nécessiter une certaine diversité de compétences pour être complétées.

Définition 11 (Tâche complexe) *Une tâche est complexe lorsque son vecteur de besoin est de taille supérieur à 1 :*

$$|B_j| > 1$$

Une collaboration des agents est donc nécessaire dans ce modèle. L'approche gloutonne utilisée dans le premier modèle n'est pas adaptée à de la collaboration, car les agents doivent se mettre d'accord sur la tâche à réaliser là où avec l'approche gloutonne les agents se déplacent et agissent de manière égoïste.

C'est pourquoi nous utiliserons un processus de négociation entre les agents. Cette négociation se fait grâce à un négociateur, par exemple un agent tiré aléatoirement. Le seul prérequis du négociateur est d'être en communication avec l'ensemble des autres agents, à la fois en réception et en emission. Cette communication peut-être directe ou indirecte. Le but de la négociation est de trouver un état qui satisfait tous les agents, par conséquent les agents doivent pouvoir exprimer des préférences sur les états. Cette méthode de négociation n'engendre pas de conflit car l'état à atteindre est calculé par le négociateur qui cherche à atteindre un consensus, mais en cas d'égalité, la préférence du négociateur prévaudra sur les autres.

Définition 12 (Vecteur de préférence) *Pour représenter les préférences sur les états des agents, les agents possèdent un vecteur de préférence défini tel que :*

$$V_n = \{s \in S \mid s \succeq s' ssi\ R(s,n) \geq R(s',n)\}$$

Ce vecteur de préférence est limité en taille par un paramètre d, lors de la construction de ce vecteur chaque agent parcourt tous les états possibles et les ajoute au vecteur si c'est possible (c'est-à-dire si le vecteur est de taille inférieure à d ou si le dernier état du vecteur à une récompense inférieure ou égale).

3.3.1 Fonction de récompense

La fonction de récompense utilisée pour construire ce vecteur est différente de celle utilisée dans le premier modèle, mais s'appuie sur la même base. Elle est définie comme suit :

$$R(s,n) = \sum_{i=1}^{k} \left(\frac{u_j c_i^n}{N_j} \left(\frac{|b_{sat}^j|}{|B_j|} \right)^f \frac{1}{\sum_{q \in j} |C_n \cap C_q| \epsilon} \right)$$

Cette fonction est découpée en 3 grands termes :

- $\frac{u_j c_i^n}{N_j}$: très similaire à celui dans la fonction du premier modèle, où la seule différence est que c_j^n qui était la compétence requise unique de la tâche devient c_i^n
- $\left(\frac{|b_{sat}^j|}{|B_j|}\right)^f$: il s'agit du nombre de besoins satisfaits par les agents alloués à j $|b_{sat}^j|$ divisé par l'ensemble des besoins $|B_j|$. Ce terme rend donc les états où les tâches sont satisfaites plus attirants pour les agents.
- $\frac{1}{\sum_{l \in j} |C_n \cap C_q| \epsilon}$: ce terme est l'inverse de la somme de la taille des intersections des compétences entre l'agent n et les agents q qui sont également associés à la tâche j. Ce troisième terme évite la redondance des tâches au sein des agents présents sur j.

La grande différence avec la fonction de récompense du modèle simple est que le comportement des agents que l'on veut inciter *via* la fonction n'est pas le même. Dans le premier modèle pour les tâches simples, si un agent est associé à une tâche, c'est que celle-ci va pouvoir être réalisée, donc pour maximiser l'efficacité des agents, il faut les inciter à se repartir sur l'ensemble des tâches. Or ici, la présence d'un agent sur une tâche ne garantit pas sa faisabilité ; si nous voulons maximiser le nombre de tâches faites, il faut récompenser les agents quand l'ensemble des besoins de la tâche sont satisfaits. C'est exactement ce que fait le deuxième terme de la fonction $\left(\frac{|b_{sat}^j|}{|B_j|}\right)^f$. Le deuxième terme est modulé par f car dans certains cas, les deux premiers termes se compensent.

Exemple 3 *Par exemple, si un agent est seul et satisfait la moitié des besoins de j, nous avons :*

- $N_j = 1$
- $\frac{|b_{sat}^j|}{|B_j|} = 0.5$,

mais si un deuxième agent vient compléter les besoins, nous aurons

- $N_j = 2$
- $\frac{|b_{sat}^j|}{|B_j|} = 1$

Or, avec u_j fixe, nous avons :

$$\frac{u_j}{1} \times \frac{1}{2} = \frac{u_j}{2} \times \frac{2}{2}$$

Donc les récompenses des deux cas sont équivalentes alors que le deuxième cas est préférable car j est pleinement satisfaite.

Afin de pouvoir réaliser une tâche, les agents doivent être complémentaires dans leurs compétences et nous voulons donc encourager cette complémentarité et cela est fait grâce au troisième terme : $\frac{1}{\sum_{l \in j} |C_n \cap C_q| \epsilon}$. Le ϵ évite d'avoir une division par 0 lorsqu'un agent est seul ou parfaitement complémentaire. Grâce à ce terme, un comportement intéressant émerge : imaginons qu'il y ait 2 agents qui partagent une part de leurs compétences mais qu'ils n'ont pas, même a 2, les compétences requises pour satisfaire pleinement une tâche, alors ils vont se mettre sur 2 tâches différentes, préférant faire chacun une part d'une tâche plutôt que de se partager une part plus grande mais incomplète d'une seule tâche.

Tous les termes de cette fonction sont encapsulés dans une somme qui concerne les compétences de l'agent n, c'est-à-dire que la récompense est la somme pour chaque besoin de j qu'il est en mesure de satisfaire.

Comme le modèle simple, cette fonction de récompense dépend en partie des compétences de l'agent, et il préférera des tâches qui correspondent plus à son vecteur de compétences. Cependant, l'effet produit par l'introduction d'agents experts dans certaines compétences est moins impactant sur le résultat qu'avec le modèle simple, notamment à cause du fait que certaines tâches peuvent partager une compétence requise et que les tâches ne se résument pas à une seule compétence. Dans le cadre des tâches simples, quand un agent est expert pour une compétence, il sera expert pour toutes les tâches nécessitant cette compétence, les mettant donc largement en priorité face aux autres, alors que dans le cadre des tâches complexes pour qu'un agent soit expert sur une tâche il faut qu'il soit expert sur l'ensemble des compétences qui constituent ses besoins.

3.3.2 Processus de négociation

Le processus de négociation se déroule de la manière suivante : dans un premier temps, un négociateur initial est choisi aléatoirement par les agents. Ce négociateur initial est le point de départ du cycle de négociation (dans l'ordre des indices des agents par exemple), c'est-à-dire qu'à chaque nouvel appel du processus de négociation, c'est l'agent suivant dans le cycle qui sera le nouveau négociateur. Pour une itération de négociation, le négociateur prend son propre vecteur de préférence comme référence, et pour chaque état dans ce vecteur de référence, il calcule l'indice carré moyen de cet état parmi les vecteurs de préférence de chaque agent. Si l'état n'apparaît pas dans un vecteur de préférence, l'indice considéré sera $d + 1$. Une fois que tous les états du vecteur de référence ont été parcourus, l'état dont l'indice carré moyen est le plus petit est désigné comme l'état négocié qui sera transmis à tous les agents. Chaque agent s'associera à la tâche en adéquation avec cet état. En cas d'égalité entre plusieurs états, c'est l'état préféré du négociateur parmi ceux-ci qui sera choisi. L'algorithme 1 donne sa version en pseudo-code, pour un appel avec le négociateur courant.

Algorithme 1 : Algorithme de Négociation

Données :
Ensemble des agents N, ensemble des états S, vecteurs de préférences V_n pour chaque agent $n \in N$.
Cycle de négociation $\mathcal{C} = \{0, \dots, |N| - 1\}$, indice du négociateur courant l

Début
- $n \leftarrow \mathcal{C}_l$
- $V_{ref} \leftarrow V_n$
- **pour** *chaque état* $s \in V_{ref}$ **faire**
  - $sqr\_sum(s) \leftarrow 0$
  - **pour** *chaque agent* $n \in \mathcal{N}$ **faire**
    - **si** $s \in V_n$ **alors**
      - $i \leftarrow$ indice de s dans V_n
    - **sinon**
      - $i \leftarrow d + 1$
    - $sqr\_sum(s) \leftarrow sqr\_sum(s) + i^2$
  - $MSI(s) \leftarrow \frac{1}{|N|} sqr\_sum(s)$
- $s^* \leftarrow \arg\min_{s \in V_{ref}} MSI(s)$
- $l \leftarrow (l + 1) \bmod |N|$

Transmettre s^* à tous les agents $n \in N$
Chaque agent n s'associe à la tâche allouée par s^*

L'algorithme de négociation a une complexité en $O(Nd^2)$. Le paramètre d est donc très important, car il contrôle la difficulté computationnelle, mais également le degré d'optimalité de l'algorithme. En effet, si d est trop petit, il y a moins de chance que le négociateur trouve un état qui fasse consensus et il prendra un état du vecteur de référence. À l'inverse, si d est grand, la négociation sera longue. Il faut donc trouver un équilibre entre optimalité et rapidité, nous suivons donc l'intuition telle que d doit être strictement supérieur au nombre d'états optimaux du système, mais des travaux sont encore nécessaires pour mieux comprendre la gestion et l'impact de ce paramètre.

4 Expérimentations

Nous avons précédemment présenté nos modèles, leurs spécificités et leur fonctionnement. Nous allons maintenant nous intéresser à leurs performances. Afin d'évaluer les performances de nos modèles, nous allons les comparer aux DEC-POMDP résolus par l'algorithme *Multi-Agent Value Iteration* (MA-VI), une des approches à l'état de l'art en planification, assurant des résultats optimaux. La politique jointe calculée par MA-VI est ensuite utilisée à la place des états alloués/négociés de nos modèles.

4.1 Résultats

Dans nos expérimentations, les seuls paramètres variants sont le nombres d'agents et le nombre de tâches.
Pour le modèle utilisant la négociation, le paramètre d a été fixé à 10 suite à des tests empiriques évaluant son influence, cette valeur ressortant comme étant pertinente. Les valeurs présentées dans le tableau 1 représentent le temps d'exécution moyen (sur 5 exécutions), en millisecondes, pour la complétion de toutes les tâches, pour chacun de nos modèles ainsi que MA-VI. La première colonne indique la configuration utilisée respectivement en termes d'agents et tâches. L'abréviation DNF signifie *did not finish* et indique que l'exécution n'a pas abouti avant un timeout préconfiguré d'une heure.

Config.	Simple	Comp.	MA-VI
2-2	53.5	110.7	138.9
2-4	80.7	234.6	735.2
4-4	81.5	3902.6	68708.1
6-6	136.2	DNF	DNF

TABLE 1 – Temps de calculs moyens

Nous constatons que le modèle simple, c'est-à-dire celui qui utilise une exploration gloutonne des états, est beaucoup plus rapide que le modèle complexe et également plus rapide que MA-VI. De plus, le modèle simple est capable de gérer un nombre de tâches et d'agents plus important, en témoigne le test sur la configuration 6 agents et 6 tâches où seul le modèle simple est parvenu à un résultat avant la fin du timeout. La rapidité du modèle simple peut s'expliquer facilement car à chaque cycle, chaque agent explore au plus $|J|$ états, donc la complexité ne dépend pas du nombre d'agents mais du nombre de tâches. La scalabilité de ce modèle sur un système distribué est donc très grande (si le système n'est pas distribué, la complexité dépend quand même du nombre d'agents). Le modèle complexe, qui utilise la négociation, est quant à lui plus lent que le modèle simple (mais plus rapide que MA-VI) et ne parvient pas à résoudre des problèmes où le nombre d'états est très grand. En effet, le modèle complexe nécessite un calcul préalable de tous les états, par conséquent, les calculs des vecteurs de

préférences se complexifient avec le nombre d'états. MA-VI, qui est plus lent que le modèle complexe, doit calculer une politique jointe sur l'ensemble des états et des agents, ce qui est extrêmement coûteux en termes de calculs, d'où sa lenteur.
Dans un second temps nous allons comparer l'optimalité des trois approches en utilisant 2 métriques : le social welfare et le nombre d'étapes pour réaliser toutes les tâches.

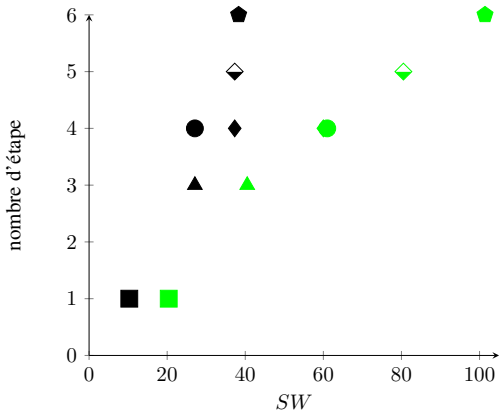


FIGURE 3 – Comparaison entre le modèle simple et MA-VI pour le social welfare et le nombre d'étapes avant complétion

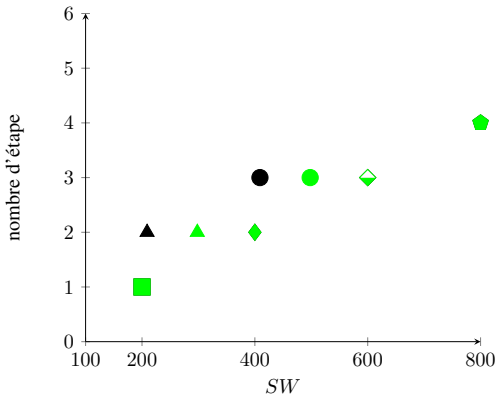


FIGURE 4 – Comparaison entre le modèle complexe et MA-VI pour le social welfare et le nombre d'étapes avant complétion

La figure 3 présente les résultats d'exprérimentations comparant le modèle simple avec MA-VI et la figure 4 présente les résultats d'exprérimentations comparant le modèle complexe avec MA-VI. Les 2 modèles ont été lancés sur des configurations contenant 2 agents et un nombre de tâches variant entre 2 (carré), 3 (triangle), 4 (losange), 5 (cercle), 6 (losange à moitié plein) et 8 (pentagone). Les résultats des modèles sont en noir tandis que ceux de MA-VI sont en vert. Ces résultats présentent le total accumulé de récompense (Social Welfare) en abscisse et le nombre d'étapes pour satisfaire toutes les tâches en ordonnée. Pour le modèle simple, nous constatons que MA-VI est plus optimal au sens du Social Welfare mais est équivalent pour le nombre d'étapes. De plus, ce gain en optimalité s'aggrandit avec le nombre de tâches disponibles. Pour le modèle complexe, les performances en termes de Social Welfare entre le modèle et MA-VI sont très similaires et seules les configurations avec un nombre impair de tâches (3 et 5) présentent une différence. Cela peut s'expliquer par le fait que lorsqu'il ne reste qu'une seule tâche, MA-VI va préferer mettre un agent seul sur la tâche. À l'inverse, les agents vont aller à 2 sur la tâche avec le modèle à négociation, ce qui se traduit par un Social Welfare inférieur.

En résumé, MA-VI dépasse le modèle simple en termes de Social Welfare, mais pour un temps de calcul beaucoup plus grand. Quant au modèle complexe, il est très similaire à MA-VI en termes de Social Welfare, toutefois, le modèle complexe est plus rapide, et la différence de rapidité s'accentue lorsque l'espace d'état grandit. Concernant le nombre d'étapes pour réaliser l'ensemble des tâches, tous les modèles sont équivalents.

5 Conclusion

Nous avons proposé dans cet article deux modèles pour la planification multi-agents de tâches. Ces modèles, fondés sur l'allocation de tâches, s'inscrivent dans un contexte où les agents ont des tâches simples ou complexes à accomplir au sein d'un environnement. Le premier modèle utilise une exploration gloutonne des états. Ce modèle est prometteur, toutefois, dû à son caractère glouton, nous supposons qu'il n'existe pas de garantie théorique de trouver un état optimal, la recherche pouvant s'arrêter sur un optimum local. Expérimentalement, l'optimalité en termes de nombre d'étapes a été atteinte à chaque fois, cependant, en termes de Social Welfare, ce modèle est légèrement en dessous. Il offre cependant une grande scalabilité ainsi qu'une grande rapidité d'exécution. Ce modèle est dédié à la résolution de problèmes ne contenant que des tâches simples, cela limite grandement les possibilités de modélisation, les tâches nécessitant l'intervention de plusieurs agents ne sont pas modélisables.

Le second modèle est quant à lui pensé pour les tâches complexes, nécessitant l'intervention de plusieurs agents. Il utilise un mécanisme de négociation et permet ainsi la coopération entre les agents. Le mécanisme de négociation de ce modèle nécessite de connaître l'ensemble des états possibles, et calculer l'ensemble des états est très vite coûteux, en témoigne les temps de calculs qui sont plus importants que ceux du modèle simple, mais reste toutefois plus rapide que MA-VI, pour une optimalité en termes de nombre d'étapes et de Social Welfare similaire. Une limite de ce modèle est liée au paramètre d, qui contrôle la taille des vecteurs de préférences des agents, et qui agit donc sur la complexité du modèle. Plus spécifiquement, la complexité de l'algorithme de négociation dépend de d, et donc si ce dernier doit être grand pour garantir l'optimalité, l'algorithme de négociation sera lent.

Les pistes de recherches futures concernent principalement les extensions à des contextes dynamiques et stochastiques,

qui nécessiteront d'adapter les modèles à l'apparition et disparition des tâches, ainsi qu'à l'estimation des récompenses désormais stochastiques. Cela nécessitera en particulier d'adapter le processus de négociation afin de mettre à jour les vecteurs de préférence, mais également de s'appuyer sur des modèles d'apprentissage (en particulier par renforcement) afin de palier aux aspects stochastiques. L'integration de tâches temporellement contraintes (durée, expiration) est également une piste de réflexion afin de se rapprocher d'une situation applicative réelle. De plus, il serait intéressant de repenser les modèles afin que le mécanisme de décision ne s'appuie pas uniquement sur la fonction de récompense.

Références

[1] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4) :819–840, 2002.

[2] Han-Lim Choi, Luc Brunet, and Jonathan P. How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4) :912–926, 2009.

[3] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. Multi-agent systems : A survey. *IEEE Access*, 6 :28573–28593, 2018.

[4] Brian P. Gerkey and Maja J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9) :939–954, 2004.

[5] Josselin Guéneron and Grégory Bonnet. De la diversité des jeux de coalitions à utilité transférable. In *29es Journées Francophones sur les Systèmes Multi-Agents*, Bordeaux, France, June 2021.

[6] Josselin Guéneron and Grégory Bonnet. Un concept de solutions avec un biais d'exploration pour les jeux de coalitions stochastiques répétés. In *31es Journées Francophones sur les Systèmes Multi-Agents*, Strasbourg, France, July 2023.

[7] Ayorkor G. Korsah, Anthony Stentz, and M Bernardine Dias. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32 :12, 2013.

[8] Carla Mouradian, Jagruti Sahoo, Roch H. Glitho, Monique J. Morrow, and Paul A. Polakos. A coalition formation algorithm for multi-robot task allocation in large-scale natural disasters. In *13th international wireless communications and mobile computing conference*, pages 1909–1914. IEEE, 2017.

[9] Laurent Péret and Frédérick Garcia. On-line search for solving markov decision processes via heuristic sampling. *Learning*, 16 :2, 2004.

Trustworthy and Efficient Deep Reinforcement Learning-Driven Physical-Layer Security for 6G Networks

Alex Pierron[1], Joaquin Garcia-Alfaro[1], Jose Rubio-Hernan[1], Michel Barbeau[2], Luca De Cicco[3]

[1] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France
[2] Carleton University, Ottawa, Canada
[3] Politecnico di Bari, Bari, Italy

alex.pierron@telecom-sudparis.eu

Résumé

Cet article court présente nos recherches sur le contrôle par Deep Reinforcement Learning (DRL) de surfaces intelligentes reconfigurables (RIS) pour la sécurité physique des futurs réseaux 6G. Nous mentionnons d'abord notre résultat sur l'intégration de l'équité dans un contrôleur DRL-RIS multi-utilisateur, puis situons nos travaux en cours sur les backdoors en DRL à partir d'une littérature fondatrice encore peu appliquée aux systèmes physiques. Nous soulignons enfin l'intérêt de notre environnement de simulation CTRL_RIS comme ressource open source reproductible pour relier résultats actuels et pistes futures sur l'équité, la sécurité et la robustesse des contrôleurs DRL-RIS pour les réseaux 6G.

Mots-clés

Intelligence Artificielle Physique, Apprentissage par Renforcement, 6G, Cybersécurité, Sécurité de la Couche Physique, Surfaces Intelligentes Reconfigurables

Abstract

This short paper presents our research on Deep Reinforcement Learning (DRL) control of reconfigurable intelligent surfaces (RIS) for the physical security of future 6G networks. We first mention our results on integrating fairness into a multi-user DRL-RIS controller, then situate our ongoing work on DRL backdoors based on foundational literature that has yet to be widely applied to physical systems. Finally, we highlight the relevance of our CTRL_RIS simulation environment as a reproducible open-source resource for linking current results and future avenues of research on fairness, security, and robustness in DRL-RIS controllers for 6G networks.

Keywords

Physical AI, Reinforcement Learning, 6G, Cybersecurity, Physical-Layer Security, Reconfigurable Intelligent Surfaces

1 Introduction

6G systems are expected to increasingly rely on Reconfigurable Intelligent Surfaces (RIS), in passive, active, or hybrid forms, to shape propagation conditions in difficult environments [1, 2]. Yet RIS remain embedded, hardware-constrained devices: large arrays of low-cost elements must be controlled in real time with limited on-board resources, and the joint optimization of phase shifts, beamforming, and physical-layer security quickly becomes difficult for classical iterative methods in dense settings [3, 4, 5]. This motivates a Physical AI approach where a policy trained by Deep Reinforcement Learning (DRL) [6, 7] directly controls the RIS, adapts to radio and security conditions, and makes fast decisions under physically grounded channel and control constraints, while also opening a new attack surface [8, 4, 9, 10].

Our recent work addressed one side of this problem, namely fairness in DRL-driven RIS-assisted physical-layer security [11]. We showed that maximizing aggregate performance is not sufficient in multi-user settings, where unfair allocations may deprive weaker users from service. Using the Jain Fairness Index [12] and the broader notions of the *price of fairness* developed in [13], we showed a misalignment between the classical Sum Secrecy Rate (SSR) used generally in multi-user telecommunications systems [14] and the behaviour obtained once the neural network is trained. This contribution provides a first step toward trustworthy DRL-driven RIS and a concrete baseline for future robustness studies. The second block concerns backdoors

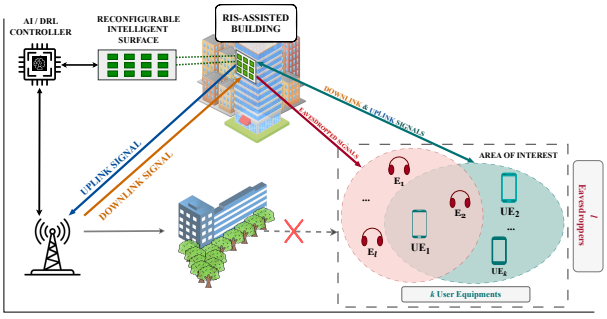


Figure 1: General DRL-driven RIS-assisted secure communication setup for 6G networks.

in DRL. While backdoor attacks are now well established

in supervised learning [15, 16], their adaptation to reinforcement learning is more recent and especially relevant for controllers deployed in long-lived cyber-physical systems. In the RIS setting, poisoning can affect training observations, rewards or environmental dynamics. The attack surface is also shaped by hardware realities such as imperfect channel estimation, finite-resolution phase control, reconfiguration latency, and RF impairments [14, 9]. This short paper positions our research along three complementary axes: fairness-aware RIS control, a concise taxonomy of DRL backdoors, and an open-source experimental basis through the *CTRL_RIS* DRL environment.[1] We provide a reproducible Python environment for secure RIS beamforming, configurable multi-user scenarios with eavesdroppers, and DRL baselines for fairness and robustness experiments.

2 Fairness vs. Backdoor-Aware RIS

Figure 1 summarizes the secure wireless environment considered in our prior fairness study [11], with a RIS model kept consistent with physically grounded formulations [14]. The main result of that paper is simple but important regarding alignment: fairness is not only a side metric but a structural requirement for reward engineering for DRL-driven RIS systems.If a reward only encourages global performance, the agent tends to privilege the user with the most favourable channels; fairness-aware shaping is needed to maintain acceptable service across users while keeping secrecy objectives meaningful [9, 11].

This observation naturally connects to trustworthiness. A poisoned DRL policy does not need to produce a spectacular failure to be harmful. An attacker may instead cause a subtle but targeted degradation of service, for instance by selectively hurting disadvantaged users, biasing resource allocation, or weakening the secrecy-fairness trade-off. In a RIS controller, such effects are further filtered by hardware limits: coarse phase updates, stale channel estimates, or impaired RF chains can hide small policy deviations while still moving the system toward an unfavourable operating point. Fairness therefore becomes both a design objective and a possible indicator of malicious behaviour.

3 Representative DRL Backdoors

Recent DRL backdoor literature assumes two main dimensions [17, 18, 19]: the attack loop and the adversarial access. *Inner-loop* attacks poison the training stream step by step, typically through state and reward manipulations; *outer-loop* attacks operate at the trajectory or episode level and can exploit richer information. A second dimension concerns the attacker interface: poisoning may alter the observed state, the reward signal, or part of the environment itself. Table 1 condenses three selected representative techniques from the current DRL backdoor literature [17, 18, 19]. TroJDRL [17] first showed that targeted malicious behaviour can be implanted through poisoned training interactions. BadRL [18] reduces the amount of poisoning required, making the attack more stealthy. SleeperNets [19] broadens the threat model through trajectory-level poisoning. However, this literature still mostly relies on benchmark-style RL environments rather than on Physical AI systems with sensing, actuation, channel, and hardware constraints. For RIS controllers, this distinction matters: a backdoor must remain effective despite quantized phase shifts, estimation noise, delayed reconfiguration, and constrained embedded control, while these same limitations can amplify the impact of small poisoned inputs once the policy is deployed [14, 9]. The literature thus provides a strong vocabulary, but not a complete answer for dynamic physical systems. The RIS case gives a concrete physical interpretation. A recent study on *pilot backdoor attacks* against DRL-empowered RIS control shows how an adversary-controlled IRS can contaminate channel state information and implant malicious behaviour in a radio controller [10]. This result is particularly relevant for our agenda because it bridges the DRL-backdoor literature and programmable wireless environments without requiring unrealistic assumptions about the attack outcome. In practice, the most plausible triggers are those aligned with the hardware and signal chain itself, such as biased pilots, corrupted Channel State Information (CSI), or malicious phase updates, rather than arbitrary perturbations. In such settings, compromised behaviour may affect secrecy, fairness, or both, which reinforces the need for evaluation protocols that go beyond average utility and remain compatible with physically grounded RIS models [14]. It is our primary goal to find effective detection and mitigation techniques to prevent problematic behaviours caused by these potential backdoors in the DRL controller.

Table 1: Three representative DRL backdoor techniques.

Method	Loop	Access	Effect
TroJDRL [17]	Inner	State/reward	Targeted policy shift
BadRL [18]	Inner	Sparse poisoning	Stealthier attack
SleeperNets [19]	Outer	Trajectories	Universal behavior

4 Conclusion

Our research addresses the challenge of building trustworthy Physical AI for future 6G networks. For DRL-driven RIS control, fairness, security, and robustness must be considered jointly. First contribution has established that fairness-aware reward design is necessary for secure multiuser RIS control [11]. We now extend this perspective to adversarial training and backdoor-aware threat modeling for DRL, using our open-source *CTRL_RIS* [2] DRL environment to evaluate poisoned training, triggered behaviour and robustness under physically grounded constraints.

Acknowledgments

The work was funded by the French National Research Agency under the France 2030 ANR program “PEPR Networks of the Future” (NF-HiSec ANR-22-PEFT-0009).

[1] https://github.com/alex-pierron/CTRL_RIS

[2] https://github.com/alex-pierron/CTRL_RIS

References

[1] Y. Liu, X. Liu, X. Mu, T. Hou, J. Xu, M. Di Renzo, and N. Al-Dhahir, "Reconfigurable Intelligent Surfaces: Principles and Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1546–1577, 2021.

[2] ETSI ISG RIS, "Reconfigurable Intelligent Surfaces (RIS); Use Cases, Deployment Scenarios and Requirements," Tech. Rep. V1.2.1, European Telecommunications Standards Institute (ETSI), February 2025.

[3] M. Di Renzo, A. Zappone, M. Debbah, M.-S. Alouini, C. Yuen, J. De Rosny, and S. Tretyakov, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2450–2525, 2020.

[4] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser miso systems exploiting deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1839–1850, 2020.

[5] Y. Feng, Q. Hu, K. Qu, W. Yang, Y. Zheng, and K. Chen, "Reconfigurable intelligent surfaces: Design, implementation, and practical demonstration," *Electromagnetic Science*, vol. 1, no. 2, pp. 1–21, 2023.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. Adaptive computation and machine learning, Cambridge, Mass: MIT Press, 1998.

[7] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5064–5078, 2022.

[8] H. Yang, Z. Xiong, J. Zhao, D. Niyato, L. Xiao, and Q. Wu, "Deep reinforcement learning-based intelligent reflecting surface for secure wireless communications," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 375–388, 2020.

[9] Z. Peng, Z. Zhang, L. Kong, C. Pan, L. Li, and J. Wang, "Deep reinforcement learning for RIS-aided multiuser full-duplex secure communications with hardware impairments," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21121–21135, 2022.

[10] Y. Huang, H.-M. Wang, Z. Wang, and W. Liu, "Pilot backdoor attack against deep reinforcement learning empowered intelligent reflection surface for smart radio," *IEEE Transactions on Wireless Communications*, 2025.

[11] A. Pierron, M. Barbeau, L. De Cicco, J. Rubio-Hernan, and J. Garcia-Alfaro, "A fairness-aware strategy for b5g physical-layer security leveraging reconfigurable intelligent surfaces," in *Foundation and Practice of Security 2025*, Springer, 2025.

[12] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, *et al.*, "A quantitative measure of fairness and discrimination," tech. rep., Digital Equipment Corporation, 1984.

[13] D. Bertsimas, V. F. Farias, and N. Trichakis, "The price of fairness," *Operations Research*, vol. 59, no. 6, pp. 1380–1393, 2011.

[14] E. Björnson, Ö. T. Demir, *et al.*, *Introduction to multiple antenna communications and reconfigurable surfaces*. Now Publishers, Inc., 2024.

[15] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 1, pp. 5–22, 2022.

[16] S. Zhang, Y. Pan, Q. Liu, Z. Yan, K.-K. R. Choo, and G. Wang, "Backdoor attacks and defenses targeting multi-domain ai models: A comprehensive review," *ACM Computing Surveys*, vol. 57, no. 4, pp. 1–35, 2024.

[17] P. Kiourti, K. Wardega, S. Jha, and W. Li, "TrojDRL: Evaluation of Backdoor Attacks on Deep Reinforcement Learning," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, July 2020. ISSN: 0738-100X.

[18] J. Cui, Y. Han, Y. Ma, J. Jiao, and J. Zhang, "BadRL: Sparse Targeted Backdoor Attack against Reinforcement Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 11687–11694, Mar. 2024.

[19] E. Rathbun, C. Amato, and A. Oprea, "SleeperNets: Universal Backdoor Poisoning Attacks Against Reinforcement Learning Agents," *Advances in Neural Information Processing Systems*, vol. 37, pp. 111994–112024, Dec. 2024.

A Survey of Multi-Agent Deep Reinforcement Learning with Graph Neural Network-Based Communication*

Valentin Cuzin-Rambaud[1], Laetitia Matignon[1], Maxime Morge[1]

[1]Université Lyon 1, INSA Lyon, CNRS, LIRIS, UMR 5205, Lyon, France

{valentin.cuzin-rambaud, laetitia.matignon, maxime.morge}@univ-lyon1.fr

Résumé

En apprentissage par renforcement multi-agents (MARL), l'intégration de mécanismes de communication permet aux agents d'apprendre à coordonner leurs actions et à converger vers leurs objectifs en partageant des informations. Sur la base d'un graphe d'interaction, une sous-classe de méthodes utilise des réseaux de neurones de graphes (GNN) pour apprendre à communiquer, ce qui permet aux agents d'améliorer leur représentation interne enrichie par les informations échangées. Avec l'essor récent des travaux dans ce domaine, nous constatons un manque de visibilité dans la distinction et la classification des approches MARL avec communication basée sur les GNNs. Ainsi, cet article passe en revue les travaux récents dans ce domaine. Nous proposons ici un processus généralisé de communication basé sur les GNNs dans le but de rendre plus évidents et accessibles les concepts sous-jacents à ces approches.

Mots-clés

Apprentissage par renforcement multi-agents, Communication, Réseau de neurones de graphes.

Abstract

In multi-agent reinforcement learning (MARL), the integration of a communication mechanism, allowing agents to better learn to coordinate their actions and converge on their objectives by sharing information. Based on an interaction graph, a subclass of methods employs graph neural networks (GNNs) to learn the communication, enabling agents to improve their internal representations by enriching them with information exchanged. With growing research, we note a lack of explicit structure and framework to distinguish and classify MARL approaches with communication based on GNNs. Thus, this paper surveys recent works in this field. We propose a generalized GNN-based communication process with the goal of making the underlying concepts behind the methods more obvious and accessible.

Keywords

Multi-Agent Reinforcement Learning, Communication, Graph Neural Networks.

*We gratefully acknowledge Université Lyon 1 and the AAP AEC for their support of this research

1 Introduction

Multi-Agent Systems (MAS) address a wide range of applications in robotics, video games, and cybersecurity. The main difficulty in designing MAS lies in developing the internal mechanisms that govern agents' behaviors and interactions. Reinforcement Learning (RL) provides a framework through which an agent can learn to behave effectively through experience. In the Multi-Agent Reinforcement Learning (MARL) setting, agents learn simultaneously while attempting to coordinate with one another by executing local policies. However, each agent typically operates under partial observability of the global state of the system and must therefore act based on incomplete information. Moreover, because all agents update their policies concurrently, the environment becomes non-stationary from the point of view of each agent: the transition dynamics depend on the evolving behaviors of the other agents. This non-stationarity significantly complicates learning in MAS and may prevent convergence to optimal policies.

To tackle partial observability and non-stationarity in MARL, we can consider communication between agents. Indeed, agents can communicate some information, e.g. their local observation or an internal representation of their mental state, to obtain a broader view of the environment and make a well-informed decision. Yet communication raises additional challenges. It requires specifying how messages are created, when and with whom communication should occur, how incoming messages are interpreted and aggregated, and how communication is integrated into the learning and execution phases.

Various approaches have been proposed in the literature to integrate communication into MARL and address its challenges. A particularly active research direction in recent years has focused on the use of Graph Neural Networks (GNNs) [20] for communication in MARL. GNNs have emerged as a popular solution to learn and extract knowledge from a graph. When data contains relationships between entities, modeling a graph permits extracting useful information. In many cases, graphs are used to represent data such as infrastructure, biological, social, and collaboration networks. In MARL with a communication context, we can thus represent the state of an environment as a graph where agents are positioned as nodes, and communication

between them is represented by edges. Zhu et al. cover MARL with communication [32] more broadly than our work, which focuses on a deeper analysis of twelve major recent GNN-based communication methods, reflecting the growing interest in GNNs for communication.
We note a lack of explicit structure and framework to distinguish and classify MARL approaches with communication based on GNNs, as there are many methods, with great diversity. Hence, this survey analyzes different methods of communication in MARL based on GNNs. We motivate the interest in using GNNs, and dive into state-of-the-art methods, by comparing them, extracting tendencies, and pointing out limitations of such approaches. Our study leads us to derive a generic algorithm which generalizes GNN-based communication process in MARL.
We first take a step back in the background Section 2 to present GNNs, RL, MARL, and MARL with communication. Section 3 presents the generic GNN-based communication process we propose and a survey of GNN-based communication MARL methods. This is followed by a specific focus on how realistic communication constraints are taken into account in state-of-the-art approaches. Finally, we conclude with future research directions in Section 4.

2 Background

In this section, we first outline key aspects of GNNs, as they are widely used for communication. Secondly, we briefly establish the foundation of RL, then explore MARL, focusing progressively on communication in MARL.

2.1 Graph Neural Networks

Several tasks on graphs, such as node classification, link prediction, and graph classification leverage machine learning methods [31]. A major approach for solving these tasks is the use of deep learning methods, in particular Graph Neural Networks (GNNs) [20].

Definition 2.1. A **graph** at time t, denoted $G^t(V, E)$, is defined by a set of n nodes V, and a set of edges E, where an edge represents the influence of the source over the target, possibly reciprocal. $\mathcal{N}(i)$ denotes the set of neighbors of node i, as $\forall j \in \mathcal{N}(i), \exists (j,i) \in E$ and $N_i = deg(i) = |\mathcal{N}(i)|$. The graph can potentially admit self-loops such as $\forall i \in V, (i,i) \in E$ and thus i is included in its neighborhood $i \in \mathcal{N}(i)$. We denote $A \in \mathbb{R}^{n \times n}$ the adjacency matrix, with $A_{ji} = 1 \iff (j,i) \in E$, and $X \in \mathbb{R}^{n \times d}$ the node feature matrix, with $X[i] \in \mathbb{R}^d$ being the feature vector of the node i and d the feature dimension. If edges also have features, we denote E_{attr} the edge feature matrix, with $E_{attr}[(j,i)]$ abbreviating $e_{j,i}$ referring to the feature vector of the edge (j,i). The graph is directed and dynamic as it evolves in a finite number of steps $t \in [0, T]$, and can be potentially weighted, with each edge having $ew_{j,i}$ the weight of the edge $(j,i) \in E$.

A GNN is defined as a parameterized function $f(X, A, E_{attr}; \theta)$. In a layered GNN with L layers, learnable parameters θ are a set of weight matrices $\{W^{(l)}\}_{l=0}^{L-1}$ shared among all nodes of the graph. The GNN learns to compute node representations (or embeddings), noted $h_i^{(l)}$ for the representation of the node i at layer l. It does so by applying iteratively (L times) the layer process. One layer process consists of two main steps for each node: 1) the **aggregation** step to gather representations from its neighbors, defined by the adjacency matrix; 2) the **transformation** step to transform the aggregated information using a weight matrix to obtain updated node representations. Thus, the edges of the graph determine which neighboring representations are aggregated, while the weight matrix controls how this information is transformed.
Inspired by the Convolutional Neural Network (CNN), one of the first GNN architectures is the Graph Convolutional Networks (GCN) [11], which is applied on non-euclidean data. Applying a convolution on an image uses a kernel with a fixed size, which is impossible in a graph, as the number of neighbors N_i of each node i varies. Thus, the aggregation step of the GCN layer process consists of pooling neighbors' features, and the transformation step consists of computing the weighted sum of neighbors' values, with learnable weights. The updated representation of a node i at layer l is:

$$h_i^{(l)} = \sigma\Big(\sum_{j \in \mathcal{N}(i)} \frac{ew_{j,i}}{\sqrt{deg(i)deg(j)}} h_j^{(l-1)} W^{(l-1)} \Big) \quad (1)$$

The average is weighted by the degree of nodes preventing high-degree nodes from dominating the aggregation and keeping the feature magnitudes stable.
More recently, Graph Attention Networks (GAT) [26] proposes a new model based on attention heads:

$$h_i^{(l)} = \sum_{j \in \mathcal{N}(i)} \alpha_{ji} h_j^{(l-1)} W^{(l-1)} \quad (2)$$

with α_{ji} the attention weights from j to i. The use of a learnable attention vector allows choosing how much attention to give to each neighbor.
GCN and GAT are two popular architectures among the many variants of GNNs. Most of them can be instantiated to a single common framework: Message Passing Neural Networks (MPNNs) [7]. The layer process with MPNN is:

$$h_i^{(l)} = \psi^{(l)}(h_i^{(l-1)}, \bigoplus_{j \in \mathcal{N}(i)} \phi^{(l)}(h_i^{(l-1)}, h_j^{(l-1)}, e_{j,i})) \quad (3)$$

with ψ and ϕ differentiable functions, e.g. Multi Layer Perceptrons (MLP), and $\bigoplus$ the aggregator function, (e.g. mean, add, max). $\phi^{(l)}$ represents the message construction, and for GCN and GAT cases, it would contain weights $W^{(l)}$.
The main advantage of using MPNN is transmitting information in the graphs to l-hops by stacking layers, with nodes communicating only with immediate neighbors. In Figure 1, the green node aggregates information from its neighbors (nodes 2,3 and 4) in 1-hop. Repeating the process permits accessing information from 2-hop nodes (node 5) as their features were previously aggregated by 1-hop

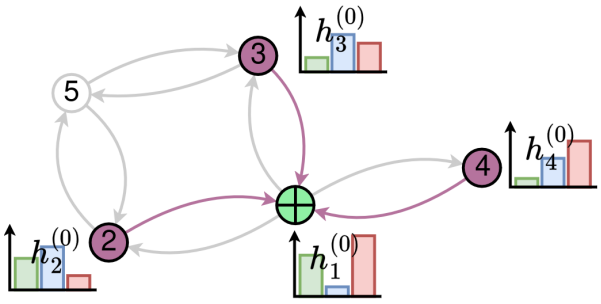


Figure 1: An example of MPNN: the green node, aggregates features from its 1-hop neighbors $\in \mathcal{N}(1)$

nodes. Other advantages of GNNs include generalizability to unseen nodes (inductive learning), invariance of permuting node order during aggregation, handling of non-Euclidean structure, and capturing local and global information to learn complex patterns. All these advantages have led communication methods to integrate GNNs in their process. In particular, in distributed communication using GNNs with $l > 1$, agents (considered as nodes) indirectly aggregate information about unreachable agents, effectively extending information propagation despite limited communication range.

2.2 Reinforcement Learning

In a sequential decision-making setting, reinforcement learning (RL) algorithms enable an agent to learn solutions through repeated experiments with an environment.

The standard model to define the sequential decision process is the Markov Decision Process (MDP) defined with S the set of environment states and μ its initial state distribution such that $s_0 \sim \mu$, A the set of actions for the agent, $\mathcal{R}$ the reward function and $\mathcal{T}$ the state transition probability function. In a more realistic case, the agent only partially observes the state of the environment. Formally, we define the Partially Observable Markov Decision Process (POMDP) as an MDP extended with O, the set of observations. $\mathcal{O}$ defines a probability distribution over possible observations. The agent interacts with the environment during a set of episodes (i.e. sequence of states and actions until a terminal state).

The discounted return from time step t is defined as the sum of discounted rewards over time, i.e. $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ where the discount factor $\gamma \in [0, 1[$ ensures finite return in non-terminating MDPs, and r_t is the reward received at time step t. A policy $\pi(a|s)$ defines a mapping from states to a probability distribution over actions. The goal of an agent is to find an optimal policy π^* which maximizes the *expected discounted return* from every state $s \in S$, i.e. $\pi^* = \arg\max_\pi \mathbb{E}_\pi[R_t|s_t = s]$. We can define two useful functions: the value function $V^\pi(s) = \mathbb{E}_\pi[R_t|s_t = s]$, and the action-value function $Q^\pi(s, a) = \mathbb{E}_\pi[R_t|s_t = s, a_t = a]$ which enable to evaluate the expected return based on a state s or a couple (s, a). The use of V or Q is essential, as an action can lead to a high immediate reward, but in the long term, a poor return.

In RL, a common assumption is that the agent has no a priori knowledge about the transition and reward functions, leading to the need to collect experiences to learn a policy π [27].

The use of deep neural networks to parameterize and approximate value functions and policies over large state and action spaces has become standard practice, giving rise to Deep Reinforcement Learning (DRL). We can classify DRL algorithms into two categories: value-based algorithms, which optimize a value function like DQN [17]; and policy-based algorithms, which optimize the policy, like REINFORCE [28]. A particular subclass of policy-based algorithm is the actor-critic algorithm, which combines the use of a value function as a critic to guide policy learning such as A2C [16] or PPO [21].

2.3 Multi-Agent Reinforcement Learning

In a multi-agent setting, a POMDP extends to a Partially Observable Stochastic Game (POSG).

Definition 2.2. A **POSG** [1] is defined by a tuple $\langle I, S, \mu, \{O_i\}, \{A_i\}, \{\mathcal{R}_i\}, \mathcal{T}, \{\mathcal{O}_i\}\rangle$, with I the set of agents indexed as $\{1, \ldots, n\}$[1], S the set of states and μ the initial state distribution such as $s^0 \sim \mu$, O_i the set of observations for agent i, A_i the set of actions. We denote the joint action space $\mathbf{A} = \times_{i \in I} A_i$. Thus, $\mathcal{R}_i : S \times \mathbf{A} \times S \to \mathbb{R}$ is the reward function for agent i, $\mathcal{T} : S \times \mathbf{A} \times S \to [0, 1]$ is the state transition probability function, and $\mathcal{O}_i : S \times \mathbf{A} \times O_i \to [0, 1]$ is the observation function of agent i.

At each time step t, each agent i receives a local observation of the current state $o_i^t \subset s^t$ given by its observation function $\mathcal{O}_i(o_i^t|s^t, a^{t-1})$. All the individual observations give the joint observation $o^t = \langle o_1^t, \ldots, o_n^t\rangle$, which approximates the current state $o_i^t \subset o^t \subseteq s^t$. In addition, each agent can memorize its observation-action history $\tau_i^t = [(o_i^0, a_i^0), \ldots, (o_i^t, a_i^t)]$ often encoded by a Recurrent Neural Network (RNN). Thus, by following its policy $\pi_i(a_i^t|\tau_i^t)$, each agent i chooses action a_i^t, forming the joint action $a^t = \langle a_1^t, \ldots, a_n^t\rangle$. The game transitions to the next state $s^{t+1} \in S$ with $\mathcal{T}(s^{t+1}|s^t, a^t)$ probability, and each agent i receives its reward $r_i^t = \mathcal{R}_i(s^t, a^t, s^{t+1})$. We note $R_i = \sum_{t=0}^{\infty} \gamma^t r_i^t$ the discounted return for the agent i.

Depending on the reward structure, POSGs can be classified as competitive, cooperative, or mixed games. In competitive games, agents are opponents (e.g., $\sum_{i \in I} \mathcal{R}_i(a) = 0, \forall a \in \mathbf{A}$). In cooperative games, agents aim to maximize the global expected return of all agents $\sum_{i \in I} \mathbb{E}_{\pi_i}[R_i]$ (e.g., common-reward: $\mathcal{R}_1 = \mathcal{R}_2 = \cdots = \mathcal{R}_n$). In mixed games, agents may share partially aligned interests while still pursuing individual objectives.

Example: Predator-Prey. The Predator-Prey environment is well-known and massively used to evaluate MARL algorithms. Let n agents (predators) evolve in a square grid by

[1] In the multi-agent setting, time steps are denoted using superscripts rather than subscripts to avoid confusion with agent indices.

choosing a movement action {up, down, left, right, stay}. Their objective is to catch a stationary prey while constrained by a limited vision range. Once a predator reaches the prey, it stays there and always gets a positive reward until the end of the episode (reaching the time step limit or all predators have reached the prey). The environment is cooperative: agents obtain a better reward when more agents reach the prey.
Multi-Agent DRL (MADRL) algorithms fall into two paradigms.

Decentralized Training and Execution (DTDE). DTDE addresses the problem independently for each agent. Each agent uses its own local information to choose its own behavior. This learning framework permits both training and execution in a physically distributed setting, such as robotic scenarios. All RL algorithms can be adapted to DTDE, e.g. IDQN [25] and IPPO [3]. However, the main problem resides in the non-stationarity of the environment, as many agents learn simultaneously. From the point of view of one agent, all other agents are part of the environment, so the environment always changes even without its action.

Centralized Training and Decentralized Execution (CTDE). In the CTDE setting, additional global information can be centralized and exploited during training to stabilize and improve learning. However, at execution time, each agent follows a decentralized policy using only its own local observations, ensuring fully decentralized execution. The use of centralized information during training helps to reduce the non-stationarity environment problem. In most environments, centralized information is accessible only during training, as the training process is typically simulated. This is the reason why many popular algorithms follow the CTDE paradigm. Some use value decomposition to centrally combine individual utility functions during training, like VDN [24] or QMIX [19], while others use a centralized critic and decentralized policies like MADDPG [15] and MAPPO [30].
Whatever the paradigm is, agents can either share the same set of network parameters or not. If agents share their parameters, only one set of parameters is learned and then duplicated into each agent, for the decentralized execution. In common-reward cooperative games, sharing weights of the model during learning permits converging faster, but at the cost of having less chance to develop diversity in behaviors between policies [1, 2].

2.4 MARL with Communication

Classical CTDE settings assume no communication between agents during execution. However, the centralization of information during training can be seen as an implicit communication process: all agents send their local observation to a centralized node that aggregates information to form the joint observation. Indeed, the messages exchanged during the training phase are not formally specified. In contrast, the explicit communication process that we refer here is learnable, where information is explicitly communicated between agents. This grant a broader representation of the global state for each agent, while remaining in a decentralized execution setting.
Most methods use communication during execution to balance the partial observability while enabling better coordination strategies. The communication process can also be used only during training, for instance to learn a more effective information combination than predefined CTDE schemes. The positive impact of communication on coordination explains why cooperative scenarios are the most relevant setting for its use.
Incorporating explicit communication in MARL requires extending the joint action space to include communication actions. Zhu et al. extend POSG with $\mathcal{M}$, the shared message space, leading to POSG-Comm [32].
Learning to communicate brings some **new challenges**:

1. Generating the content of the message to be sent.
2. Choosing when and with whom to communicate.
3. Interpreting/combining received messages.
4. Leveraging the acquired knowledge.

Many methods have been proposed to solve these new challenges [32]. Each of these issues can be resolved either through learning or by relying on a fixed, manually designed solution. Most state-of-the-art approaches incorporate learning to handle at least one of these challenges. The communication is often learned in an end-to-end fashion (with backpropagation flow for all differentiable functions), so agents communicate directly to improve the global return.
A key distinction between MARL methods with communication is whether the communication requires a central control node or not. This means that some methods need a centralized data structure to control the communication policy. This is endorsed by the proxy, an agent that cannot directly interact with the environment but is responsible for the communication policy, e.g. learning to aggregate observations for a centralized critic. The use of a proxy during the training phase is a soft constraint, as it is often implemented within a simulator and centralized. Yet during execution it is a harder constraint, as it requires perfect communication between all the agents and the proxy. While this constraint may be realistic in some settings, such as warehouse environments, it restricts the applicability to other contexts, e.g. fleet of drones. In methods without a proxy, each agent communicates in a distributed manner, within a communication-range, which makes the approach more flexible and applicable to a wider set of scenarios.
Algorithm 1 summarizes the process for the training and the execution phases. Parametric functions are any common learnable function $f(\cdot;\theta)$ with associated weight θ. For example, it can be used to compute action probabilities for the policy or to serve a critic in actor-critic methods. The communication process permits obtaining relevant information by exchanging messages between agents, which influences the policy of agents. Agents are either fully connected, or constrained by the communication range. Various mechanisms using GNNs to address communication challenges will be explored in the following section.

Algorithm 1: General Pipeline for MARL with communication

Input: The environment env

Initialize all parametric functions $f_i(\cdot;\theta_i)$ with their associated weights θ_i ;
for *every episode* **do**
  Observe at $o_1^0, \ldots, o_n^0$ from env at $t = 0$;
  for *time step* $t = 1, 2, \ldots, T-1$ **do**
    ▷ Create representations with Algo. 2
    $h_1^t, \ldots, h_n^t = Communicate(o_1^t, \ldots, o_n^t)$;
    Sample actions $a_1^t, \ldots, a_n^t \sim \pi_1^t(\cdot|h_1^t;\theta_1), \ldots, \pi_n^t(\cdot|h_n^t;\theta_n)$;
    Perform actions in env ;
    Collect rewards $r_1^t, \ldots, r_n^t$ from env ;
    Observe $o_1^{t+1}, \ldots, o_n^{t+1}$ from env ;
    if *in Training phase* **then**
      Compute losses for parametric functions, based on agents' interactions with env ;
      Update θ_i by backpropagating gradients ;

3 MADRL with GNN-Based Communication

GNNs excel at propagating information between nodes and at modeling structured and dynamic interactions among agents [10, 12, 13]. In this section, we first abstract and generalize the communication process used in existing state-of-the-art approaches to design a generic algorithm for communication in MADRL. Secondly, we survey GNN-based communication methods, their advances, and limitations, including the widespread neglect of communication constraints.

3.1 Communication Model with GNNs

Introducing GNNs in the communication process can help to solve challenges. Creating messages through GNNs permits communication in multiple rounds. Building a graph at each timestep and designing an MPNN determines when[2] and with whom to communicate, and how to create messages and interpret received messages. The last aggregation and transformation made by the MPNN gives a final representation (i.e., communication-aware representation) that is integrated in the learning process, and optionally in the execution.

To aggregate many GNN-based methods within a high-level framework, we propose a generic communication process using GNNs in MADRL, presented in Algorithm 2. We distinguish between two cases: the use of a proxy or distributed communication without a proxy. The associated proxy part is detailed in Algorithm 3.

The algorithm can be unfolded into four main parts:

[2] At each timestep, the agent decides whom to communicate with, so agents implicitly learn when to communicate.

Algorithm 2: Generic communication process in MADRL with GNNs at time t for agent i

Input: o_i^t: the local observation

▷ Encode the message to send
Encode current representation: $x_i^t \leftarrow E(o_i^t)$;
▷ Decide with whom to communicate
if *exists a proxy* P **then**
  Send x_i^t to P ;
  ▷ Execute Algo. 3
  Receive $h_{P,i}^{t,L}$ from P;
if *exists a distributed comm* **then**
  Send x_i^t to $\{j \in \mathcal{N}_r^t(i)\}$;
  $X_i^t \leftarrow \{x_j^t | i \in \mathcal{N}_r^t(j), \forall j \in I\}$;
  ▷ Combine received msg
  $G_i^t(V_i^t, E_i^t) \leftarrow G_{build}(X_i^t)$: the local graph ;
  $H_i^{t,0} = X_i^t$;
  for $l = 1 \ldots L$ **do**
    ▷ Aggregate msg
    $h_i^{t,l} \leftarrow GNN_i^l(H_i^{t,l-1}, E_i^t)[i]$;
    if *multi-round comm. and* $l > 1$ **then**
      ▷ Exchange updated representation
      Send $h_i^{t,l}$ to $\{j \in \mathcal{N}_r^t(i)\}$;
      $H_i^{t,l} \leftarrow \{h_j^{t,l} | i \in \mathcal{N}_r^t(j), \forall j \in I\}$;
    if *evolution of relation* **then**
      $G_i^t(V_i^t, E_i^t) \leftarrow G_{build}^l(H_i^{t,l})$;

Output: The final representation $h_i^{t,L}$, Possibly final representation from the proxy $h_{P,i}^{t,L}$

▷ Inner integration to policy/value-level

1. Encode the message to send. To generate the first message to send at time t, we use $E(\cdot)$, the message encoder: it maps local observation of the agent o_i^t to encoded message x_i^t (line 1). It can be implemented as any parametric function (e.g., MLP, RNN, CNN), or as a predefined mapping, such as the identity function, which preserves the raw input. Using an RNN introduces a hidden state to capture history.

2. Decide with whom to communicate. In proxy communication settings, all agents send their local representations to the proxy (line 3). In distributed settings, each agent i sends its representation to agents $j \in \mathcal{N}_r^t(i)$ with $\mathcal{N}_r^t(i)$ containing the set of reachable neighbors from i, at time t (line 6). Each agent i, received messages from every agent $j \in I$, that considers $i \in \mathcal{N}_r^t(j)$ (line 7). $\mathcal{N}_r^t(k)$ is determined by potential communication range or learnable choices to decide with whom it is possible or beneficial to communicate.

3a. Combine received messages with distributed communication. Following the previous message-sending phase at time t, each agent i receives a set of messages X_i^t

from the others (line 7). A local graph for agent i at time t, noted G_i^t, is constructed by $G_{build}(\cdot)$ function, with all previous received messages X_i^t (line 8). This local graph G_i^t represents all reachable agents from i (all $j \in \mathcal{N}_r^t(i)$) as nodes, their communication as edges, and message contents as node feature $X_i^t[j] : \forall j \in \mathcal{N}_r^t(i)$ (see Def. 2.1). G_i^t is then used to aggregate messages with GNNs. Combining messages until the last layer L of GNNs leads the agent to obtain a final representation for its corresponding node in the graph, $h_i^{t,L}$, which is intended to be a broader representation of the current global state of the world. If $l = 1$, information is aggregated only with the 1-hop neighbor, but if $l > 1$, two possible cases occur. In the *multi-round communication* case, at each layer of GNNs, each agent communicates with its neighbors, and its intern representation $h_i^{t,l}$ is updated at layer l (line 13). The *multi-round communication* leverages the diffusion/propagation mechanism of MPNN (see Fig. 1). Otherwise, each agent i computes representations l times, for all reachable nodes, which implies that the i's representation of agent j may differ from j's self-representation (i.e., they do not have the same local graph). The multi-round communication ensures construction of more globally consistent representations, but demands more messages. Whereas without multi-round communication, agents communicate only once, but representations have limited global consistency and may be biased. Furthermore, with the *evolution of relation* enabled, G_i^t can be updated at each layer via $G_{build}^l(\cdot)$ to learn a different communication structure within the same timestep (line 16). The final representation $h_i^{t,L}$ is integrated in the MADRL pipeline in the next phase. The final representation replaces the usage of raw observation o_i^t.

Algorithm 3: Proxy' communication process at time t

1 Receive $X_P^t \leftarrow \{x_i^t | \forall i \in I\}$;
2 $G_P^t(V_P^t, E_P^t) \leftarrow G_{build}(X_P^t)$: the graph of Proxy ;
3 $H_P^{t,0} = X_P^t$;
4 **for** $l = 1 \dots L$ **do**
5 $\quad H_P^{t,l} \leftarrow GNN_P^l(H_P^{t,l-1}, E_P^t)$;
6 $\quad$ **if** *evolution of relation* **then**
7 $\quad\quad G_P^t(V_P^t, E_P^t) \leftarrow {G_{build}}^l(H_P^{t,l})$;
8 $\forall i \in I$, send $H_P^{t,L}[i]$;

3b. Combine received messages with the proxy (Algo. 3). In proxy communication, the construction of the graph is centralized. The proxy receives messages from all agents X_P^t (line 1). This global view allows to build a more globally consistent graph than the distributed communication, where all nodes are represented, and edges are not restricted by a communication range (line 2). Communication is established once, with the strong requirement that all agents are connected to the proxy. Information of nodes are aggregated through successive layers via the GNN. The final joint representation of all nodes $H_P^{t,L}$ can be used directly as centralized information (replacing the joint observation), and the proxy sends back to each agent its self-representation $h_{P,i}^{t,L} = H_P^{t,L}[i]$.

4. Inner integration to policy/value-level. The obtained final representation $h_i^{t,L}$ is leveraged in training, execution, or both, depending on the MADRL method (value-based, value-decomposition, or actor-critic). If a proxy exists, each agent retrieves its final representation made by the proxy $h_{P,i}^{t,L}$ and can potentially use its. We note that the full representation $H^{t,L}$ can serve for centralized training. The final representation can be combined with other knowledge of the agent before any usage, e.g. the concatenation with raw observation $h_i^{t,L} \leftarrow [o_i^t \oplus h_i^{t,L}]$. The communication is often learned in an end-to-end manner, but in specific methods, an additional objective function can update communication-dependent weights during supervised, self-supervised, or reinforcement learning.

Example: Predator-Prey with distributed communication. We assume that three agents are in range at time t. If agents communicate their local observations, they can cover a broad search space. By sharing information about previously explored areas where the prey was not observed, agents can take more informed actions. If agent i finds the prey, its message will inform its two neighbors. With *multi-round communication*, the information can propagate beyond the communication range limit of the agent, similarly to Figure 1. Any agent connected to the real communication graph, and at $L - hop$ from agent i, can obtain the prey's position.

3.2 GNN for communication

Building upon the generic algorithm introduced earlier, we provide in this section a survey of GNN-based communication MADRL methods by instantiating each component of our generic algorithm (cf. Table 1). The proposed generic algorithm facilitates comparative analysis, reveals common tendencies and structural patterns, and enables the classification of existing methods.

Proxy-Communication Methods. Many methods are based on a proxy to handle all communications. One of the first methods to use a proxy is Deep Implicit Coordination Graphs (DICG) [12], which extends MAPPO [30] and uses communication for the centralized critic only. This permits fully decentralized execution since the critic is used only during training. DICG uses the proxy to compute a complete weighted graph called Implicit Coordination Graph. Each edge obtains a weight computed by a self-attention mechanism. The proxy then applies a GCN (Eq. 1) on the entire graph to obtain the final joint representation matrix, which feeds the centralized critic. Basically, DICG aggregates observations instead of using a concatenation of local observations to create the joint observation.

Unlike DICG, Game Abstraction Communication (GA-Comm) [13] uses a proxy for both training and execution. The graph is built in two phases: hard attention (dropping edge) and soft attention (weighting all remain edges via self-attention). This process is called the game abstraction of agents' interactions. The GNN used is a

Table 1: Taxonomy of GNN-based communication methods, organized in two categories: proxy-based (upper part) and distributed-based (lower part) according to the generic Algorithm 2. $E(\cdot)$ encodes the local representation, $\mathcal{N}_r^t(\cdot)$ defines reachable agents, $G_{build}(\cdot)$ builds a graph, GNN is the architecture used, and Inner integration shows how to leverage the final representation.

Methods	$E(\cdot)$	$\mathcal{N}_r^t(\cdot)$	$G_{build}(\cdot)$	GNN	Inner integration
DICG [12]	$\{LSTM, MLP\}(o_i^t)$	All agents	G_P^t : complete, weighted	$L=2, GCN$	$V(H_P^{t,L})$
GA-Comm [13]	$LSTM(MLP(o_i^t))$	All agents	G_P^t : sparse, weighted	$L=1, MPNN$ with attention	$\pi_i(\cdot\|h_{P,i}^{t,L})$
GAAC [13]	$LSTM(MLP(o_i^t))$	All agents	G_P^t : sparse, weighted	$L=1, MPNN$ with attention	$Q_i(h_{P,i}^{t,L})$
MAGIC [18]	$MLP(LSTM(MLP(o_i^t)))$	All agents	$G_P^{t,l}$: sparse, dynamic	$L=2/3, GAT$	$\pi_i(\cdot\|h_{P,i}^{t,L}), V_i(h_{P,i}^{t,L})$
GACG [4]	$MLP(o_i^t)$	All agents	G_P^t : sparse, weighted, group	$L=2, GCN$	$Q_i(h_{P,i}^{t,L})$
LTS-CG [5]	$MLP(o_i^t)$	All agents	G_P^t : sparse, weighted, temporal learned	$L=2, GCN$	$Q_i(h_{P,i}^{t,L})$
DGN [10]	$MLP(o_i^t)$	near agents	G_i^t : sparse	$L=2, MPNN$ with attention	$Q_i(h_i^{t,L})$
LSC [14]	$MLP(o_i^t)$	near hierarchic agents	G_i^t : sparse, hierarchic, heterogeneous	$L=3, MPNN$	$Q_i(h_i^{t,L})$
MAGE-X [29]	$MLP(o_i^t)$	near agents	G_i^t : sparse	$L=2, GCN$	$\pi_i(\cdot\|h_i^{t,L})$
MAGEC [8]	$Identity(o_i^t)$	near agents	G_i^t : sparse, heterogeneous	$L=10, GraphSAGE$	$\pi_i(\cdot\|h_i^{t,L}), V_i(h_i^{t,L})$
(Het)GPPO [2]	$Identity(o_i^t)$	near agents	G_i^t : sparse, undirected	$L=1, MPNN$	$\pi_i(\cdot\|h_i^{t,L}), V_i(h_i^{t,L})$
HetNet [22]	$MLP_i(LSTM(MLP(o_i^t)))$	near agents	G_i^t : sparse, undirected	$L=3, HetGAT$	$\pi_i(\cdot\|h_i^{t,L}), V(H_{P,i}^{t,L})$

custom one (Eq. 3), resulting from aggregation weighted by both hard and soft attention computed weights: $h_i = \sum_{j \neq i} W_h^{i,j} W_s^{i,j} h_j$. The final representation is then used for the policy trained with an independent multi-agent version of the REINFORCE algorithm [28]. A second implementation called Game Abstraction Actor-Critic (GAAC) is used to compute the local Q_i critic, thus, execution remains totally decentralized, without communication [13].

Multi-Agent Graph-attention Communication (MAGIC) [18] can be seen as an upgrade of GA-Comm, as it keeps the principle of hard and soft attention. The hard attention is handled here by a "sub-scheduler" process, while soft attention is directly learn through GAT (Eq. 2). The first sub-scheduler uses a GAT in the complete graph to compute the first node feature matrix X_P^t, then it processes through a sample of effective edges (probabilities to binary). Any additional sub-scheduler reuses X_P^t and samples a new adjacency matrix, updating the graph. The key contribution is the possibility of stacking several sub-schedulers followed by GAT aggregation. Thus, the graph is rebuilt between each aggregation of messages, (cf. line 6 of Algo. 3). Several layers of GNNs with evolving structure permit learning different kinds of relationships at each timestep. All created representations of an agent are concatenated and used for critic during training, and for policy during both training and execution.

Group Aware Coordination Graph (GACG) [4] builds a graph by enforcing the group structure to help agents cohesion. First, groups are formed with a classifier looking at a temporal window in the joint-observation history: if two agents have similar representations, they are in the same group. Then, the proxy computes the agent-pair matrix, which encode weight of relation between all agents, and besides it computes the edge-group matrix, which encodes if edges belong or not to the same group. A weighted adjacency matrix is then sample from a Gaussian distribution based on the agent-pair matrix as mean and the edge-group matrix as covariance. The method, which is built on top of QMIX [19], uses a GCN to aggregate messages, and the final representation is integrated into individual Q_i values.

Latent Temporal Sparse Coordination Graph (LTS-CG) [5] is another proxy method, focusing on building graphs by leveraging temporal information. First, the method creates an agent-pair matrix as GACG, from correlations between previous trajectories of agents. Then, a graph is sampled on a Bernoulli distribution with probabilities from the agent-group matrix. The key contribution comes from two pretext tasks used during the learning: predict-future observation and infer-present states. These tasks guide graph construction toward a latent temporal sparse graph that integrates temporal information in the structure of the graph. Furthermore, the graph is weighted and uses GCN (Eq. 1) to aggregate messages, integrated in the individual Q_i value for the QMIX framework [19].

Using a proxy during the execution requires all agents to have perfect communication with the proxy. Even under decentralized execution, the proxy centralizes the communication policy. So the proxy is a very strong constraint, and proxy-methods can not adapt to fully decentralized execution. It is worth noting that among proxy-based methods, only DICG and GAAC assume a fully decentralized execution, as they use the proxy only during training and do not communicate during execution.

Distributed Communication Methods. As discussed previously, using a proxy is a strong constraint on the environment, and establishing communication in a distributed manner among agents enables a wider range of applications. Moreover, communications between nearby agents are often more relevant than those between distant agents, because distant agents' observations are often not useful to others and can introduce irrelevant context. For this purpose, Graph Convolutional Reinforcement Learning (DGN) extends IDQN with GNNs for communication [10]. Each agent exchange its encoded message to reachable neighbors (cf. lines 6-7 of Algo. 2). A custom MPNN (Eq. 3) aggregates messages using self-attention weights during averaging. DGN enable *multi-round communication* (cf. lines 12-14 of Algo. 2), so each agent re-sends its updated representation $h_i^{t,1}$ to its neighbors. As explained in Example 2.3, this leverages the MPNN propagation mechanism (as

in Fig. 1). The inner integration plays a role in training and execution as Q_i is conditioned on the concatenation of $h_i^{t,1}$ and $h_i^{t,2}$ for helping the model to understand different kinds of relationships.

Scaling to larger number of agents is challenging in MARL. Using a hierarchical structure leverage more sparse communication topology. Learning Structured Communication (LSC) exploit a two-level hierarchical structure, with two agent types: low and high [23]. All low agents can communicate only with high-level neighbors. High agents are elected depending on a computed weight attribute based on local observations. If in the agent i range there is no high agent, and the i's weight is bigger than neighbors' weight, i becomes the new high agent. Agents do not aggregate information in the same ways, and thus, the behavior is not the same for all agents. Integrating this approach into our generic algorithm implies that high-level agent receive information, build a local graph G_i, and aggregate information at $l = 1$ with a GNN. Then, they exchange with high-level agents only, updating topology of G_i (cf. lines 15-16 of Algo. 2), and perform a second aggregation at $l = 2$. Finally, they send back final representations to each reachable low-level agent. A low-level agent, just sends information to high-level agents and receives an answer from them, which will be aggregated only once with $L_{low} = 1$. Their custom GNN sums received features and updates via a MLP (like a GCN without degree-normalization). The inner integration is on computing local Q_i values, during both training and execution, as LSC extends IDQN [25].

Although Multi-Agent Graph-Enhanced Commander-Executor (MAGE-X) [29] is presented as a distributed communication method, it still requires centralization at execution time. A supervising agent first assigns a goal to each agent in navigation tasks. Then, each agent constructs a local complete subgraph that includes all reachable neighbors. A first GCN is applied to this subgraph to compute node attributes, after which an adjacency sampling process is performed to obtain the graph G_t^i. A second GCN is then applied to G_t^i to produce the final representation. This representation is combined with the encoded goal assignment and integrated into both the policy and value functions within an IPPO-based framework [3].

Multi-Agent Graph Embedding-based Coordination (MAGEC) [8] addresses the patrolling problem, where agents navigate a graph to minimize the node idleness. This method builds a heterogeneous graph that includes both game's nodes and neighboring agents. An agent builds first a sub-graph with its limited observation: each node contains the type of node (agent/game node), the idleness time, and the degree of the node. Edges include a relative-position attribute. Agents communicate their position, goal-reached notifications, and attribution notifications (i.e. if the agent dies) to others. All received communication extends the graph with new agent node. The GNN used is GraphSAGE [9], with ten layers and a single-round communication, so many aggregations happen to enrich the final representation. GraphSAGE was designed for inductive learning and generalizes well to unseen nodes. The final representation is passed to the policy, which chooses the next node to visit. The main advantage is that the method shows resilience against noisy communication and agent attributions, compared to other patrolling algorithms.

Graph Proximal Policy Optimization (GPPO) and Heterogeneous GPPO (HetGPPO) [2] are two models of communication extending IPPO [3] with GNNs. GPPO uses parameter sharing while HetGPPO uses independent parameters. In detail, the local sub-graph is built in a predefined manner: bidirectional edges appear if nodes are in range of communication. Then, any MPNN (Eq. 3) process only once to aggregate received messages. Finally, $[o_i^t \oplus h_i^{t,1}]$ are fed into an MLP, and the output serves both policy and critic. The authors argue that HetGPPO leads to better heterogeneous behaviors, while being more resilient in noisy environments.

Heterogeneous Policy Networks (HetNet) studies communication between physically heterogeneous agents [22]. The method posits that agents needs to learn different type of communication, depending on the class of agents. The graph is built in a predefined manner, like GPPO. Then, a custom MPNN called HetGAT learns different sets of weights for each class received messages. Concretely, if agent i receives a message x_j^t from an agent of a different class, it uses a dedicated attention weight α^{j2i}. If it receives a message x_k^t from an agent of the same class, it instead uses the attention weight α^{i2i}. Moreover, HetGAT accounts for bandwidth by learning to sample fixed bit size messages. HetNet stacks several HetGAT layers, with *multi-round communication* at each layer (cf. lines 12-14 of Algo. 2). Extending MAPPO, the final representation serves the policy during training and execution. For the centralized critic, the method uses a proxy as DICG to learn with a HetGAT on a complete graph. This method uses both a proxy and distributed communication (cf. lines 2 and 5 of Algo. 2), but the proxy is used only during the centralized training.

3.3 Communication constraints

To deploy MADRL with communication in real-world applications, several other constraints come into play, in particular concerning the communication. Indeed, in realistic scenarios, communication is subject to constraints and associated costs. Table 2 provides a summary of the communication constraints evaluated during execution and/or accounted for in state-of-the-art methods. The first constraint relates to connectivity, characterized by a limited communication range (CR) among agents. While this constraint is not compatible with proxy-based architectures, it is naturally incorporated in distributed communication methods, especially, multi-round communication provides a solution to overcome this limitation. Another constraint is the scalability of the method in terms of number of agents. Scaling to larger number of agents is easier in distributed communication, while a proxy communication suffers from greater complexity due to the size of the joint observation space. Another important aspect of realistic communica-

tion is the limited bandwidth (LB), which requires to optimally encode information when communication capacity is limited. Few works begin to focus on this, as does HetNet [22] or Bandwidth-constrained Variational Message Encoding (BVME) [6] very recently. Moreover, noisy messages (NM) and communication loss (CL) are yet to be fully taken into account. Few methods are resilient to noisy messages [2, 8].

Table 2: Tested communication constraints during execution

Methods	CR	$\max(n)$	LB	NM	CL
GA-Comm [13]		20			
MAGIC [18]		20			
GACG [4]		10			
LTS-CG [5]		27			
DGN [10]	×	60			
LSC [14]	×	60			
MAGE-X [29]	×	50			
MAGEC [8]	×	6		×	×
GPPO [2]	×	2			
HetGPPO [2]	×	2		×	
HetNet [22]	×	10	×		

To deploy communicating learning agents in real-world environment, integrating directly these constraints into the learning process of communication will enable the handling of more complex and realistic scenarios.

4 Conclusion

Communication has emerged as a major solution to overcome partial observability and non-stationarity problems of Multi-Agent Deep Reinforcement Learning (MADRL) algorithms. Nevertheless, communication methods must address several key design challenges to be effective: determining the content of transmitted messages, selecting appropriate recipients, interpreting and combining received information, and leveraging communication-aware representations for decision-making. Our survey highlights the use of Graph Neural Networks (GNNs) in the Multi-Agent Deep Reinforcement Learning (MADRL) literature as a principled approach to partially address these four challenges.

We propose a generic algorithm that generalizes the GNN-based communication processes, which presents how GNNs handle communication challenges. Our algorithm captures the diversity in methods using GNNs and permits a comparative study of existing approaches, emphasizing common and different structural elements and limitations. We have classified existing methods from two complementary perspectives: proxy communication and distributed communication. The proxy assumption enables the construction of a global graph, facilitating stronger coordination among agents. However, it relies on the assumption of perfect communication. In contrast, distributed communication supports local coordination within communication range, reflecting common practical constraints. In addition, we show the limitations of GNN-based communication methods in communication-constrained environments. Lastly, we observe that taking into account the temporal information and the structural information is a key point for better communication between agents. Encoding messages often uses an RNN to aggregate observations from the past, and its help to converge in a partially observable environment. However, the correlation of temporal and structural information, as LTS-CG leverages [5], can be a potential research direction.

The validation of the proposed generalization is intended to rely on the design of a unified framework enabling systematic comparison, reproducibility and empirical verification across existing and future methods.

Finally, future research should more explicitly address realistic constraints, as summarized in Table 2, to bridge the gap between theoretical communication models and practical applications of multi-agent systems.

References

[1] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *"Multi-Agent Reinforcement Learning: Foundations and Modern Approaches"*. MIT Press, 2024.

[2] Matteo Bettini, Ajay Shankar, and Amanda Prorok. "Heterogeneous Multi-Robot Reinforcement Learning". In *AAMAS*, pages 1485–1494, 2023.

[3] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. "Is Independent Learning All You Need in the Starcraft Multi-Agent Challenge?". *arXiv preprint arXiv:2011.09533*, 2020.

[4] Wei Duan, Jie Lu, and Junyu Xuan. "Group-Aware Coordination Graph for Multi-Agent Reinforcement Learning". In *IJCAI*, 2024.

[5] Wei Duan, Jie Lu, and Junyu Xuan. "Inferring Latent Temporal Sparse Coordination Graph for Multiagent Reinforcement Learning". *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):14358–14370, 2025-08.

[6] Wei Duan, Jie Lu, En Yu, and Junyu Xuan. "Bandwidth-Constrained Variational Message Encoding for Cooperative Multi-Agent Reinforcement Learning". In *AAMAS*, page 13, 2026.

[7] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. "Neural Message Passing for Quantum Chemistry". In *ICML*, pages 1263–1272, 2017.

[8] Anthony Goeckner, Yueyuan Sui, Nicolas Martinet, Xinliang Li, and Qi Zhu. "Graph Neural Network-Based Multi-Agent Reinforcement Learning for Resilient Distributed Coordination of Multi-Robot Systems". In *IROS*, pages 5732–5739, 2024.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs". In *NeurIPS*, volume 30, pages 1025–1035, 2017.

[10] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. "Graph Convolutional Reinforcement Learning". In *ICLR*, 2020.

[11] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In *ICLR*, 2017.

[12] Sheng Li, Jayesh K. Gupta, Peter Morales, Ross Allen, and Mykel J. Kochenderfer. "Deep Implicit Coordination Graphs for Multi-Agent Reinforcement Learning". In *AAMAS*, pages 764–772, 2021.

[13] Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. "Multi-Agent Game Abstraction via Graph Attention Neural Network". In *AAAI*, volume 34, pages 7211–7218, 2020.

[14] Zeyang Liu, Lipeng Wan, Xue Sui, Zhuoran Chen, Kewu Sun, and Xuguang Lan. "Deep Hierarchical Communication Graph in Multi-Agent Reinforcement Learning.". In *IJCAI*, pages 208–216, 2023.

[15] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In *NeurIPS*, volume 30, pages 6382–6393, 2017.

[16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning". In *ICML*, pages 1928–1937, 2016.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing Atari With Deep Reinforcement Learning". *arXiv preprint arXiv:1312.5602*, 2013.

[18] Yaru Niu, Rohan Paleja, and Matthew Gombolay. "Multi-Agent Graph-Attention Communication and Teaming.". In *AAMAS*, pages 964–973, 2021.

[19] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. "Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning". *JMLR*, 21(178):1–51, 2020.

[20] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. "The Graph Neural Network Model". *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[21] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal Policy Optimization Algorithms". *arXiv preprint arXiv:1707.06347*, 2017.

[22] Esmaeil Seraj, Zheyuan Wang, Rohan Paleja, Daniel Martin, Matthew Sklar, Anirudh Patel, and Matthew Gombolay. "Learning Efficient Diverse Communication for Cooperative Heterogeneous Teaming". In *AAMAS*, pages 1173–1182, 2022.

[23] Junjie Sheng, Xiangfeng Wang, Bo Jin, Junchi Yan, Wenhao Li, Tsung-Hui Chang, Jun Wang, and Hongyuan Zha. "Learning Structured Communication for Multi-Agent Reinforcement Learning". *JAAMAS*, 36(2):50, 2022.

[24] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. "Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward". In *AAMAS*, pages 2085–2087, 2018.

[25] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. "Multiagent Cooperation and Competition With Deep Reinforcement Learning". *PloS one*, 12(4):e0172395, 2017.

[26] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In *ICLR*, 2018.

[27] Christopher JCH Watkins and Peter Dayan. "Q-learning". *Machine learning*, 8(3):279–292, 1992.

[28] Ronald J Williams. "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". *Machine learning*, 8(3):229–256, 1992.

[29] Xinyi Yang, Shiyu Huang, Yiwen Sun, Yuxiang Yang, Chao Yu, Wei-Wei Tu, Huazhong Yang, and Yu Wang. "Learning Graph-Enhanced Commander-Executor for Multi-Agent Navigation". In *AAMAS*, pages 1652–1660, 2023.

[30] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games". *NeurIPS*, 35:24611–24624, 2022.

[31] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. "Graph Neural Networks: A Review of Methods and Applications". *AI Open*, 1:57–81, 2020.

[32] Changxi Zhu, Mehdi Dastani, and Shihan Wang. "A Survey of Multi-Agent Deep Reinforcement Learning With Communication". *JAAMAS*, 38(1), 2024.

Session 3 : Santé & Environnement

A Knowledge Graph and Graph Neural Network Framework for Air Quality-Health Relationships

Elisa Drouot[1,2], Thierno Diallo[1], Gayo Diallo[2]

[1] Centre Scientifique et Technique du Bâtiment, Grenoble, France
[2] BPH Inserm 1219 Research Center, Univ. Bordeaux, F-33000, Bordeaux, France

elisa.drouot@cstb.fr

Résumé

L'air que nous respirons provient majoritairement de l'environement intérieur (estimé à 90% de notre temps). Or la qualité de l'air étant un enjeu de santé publique, de nombreuses études ont mis en évidence des liens significatifs entre exposition à différents polluants et santé humaine. Dans ce contexte, nos travaux visent à intégrer des données hétérogènes environnementales temporelles, issues des bâtiments, de la toxicogénomique et de données agrégées de santé (prévalence de pathologies, causes de décès) avec une approche à base de Graphes de Connaissances et de Graph Neural Networks (GNN) afin de caractériser et prédire les associations possibles entre polluants et pathologies. Les résultats préliminaires ont montré qu'il était possible d'obtenir une perte d'entraînement (train loss) de 0.0942 et une AUC (Area Under the Curve) sur les données de validation de 0.8473.

Mots-clés

Qualité de l'Air Intérieur, Santé environnementale, Prédiction de liens, Graphes de Connaissances, GNN

Abstract

The air we breathe comes predominantly from indoor environments, where we spend an estimated 90% of our time. As indoor air quality represents a major public health concern, numerous studies have established significant links between exposure to various pollutants and human health. In this context, our research aims to integrate heterogeneous temporal environmental, from buildings, toxicogenomics, and aggregated health records (such as disease prevalence and causes of death). By employing an approach based on Knowledge Graphs and Graph Neural Networks (GNN), we seek to characterize and predict potential associations between pollutants and pathologies. Preliminary results demonstrate the ability to achieve a training loss of 0.0942 and a validation AUC (Area Under the Curve) of 0.8473.

Keywords

Indoor Air quality, Environmental Health, Link Prediction, Knowledge Graph, GNN

1 Introduction

According to the State of Global Air (SoGA) report published by the Health Effects Institute (HEI), air pollution is the second leading cause of death worldwide, responsible for 8.1 million deaths globally in 2021. More precisely, 38% of these deaths are attributed to household air pollution [6]. Indoor pollutant concentrations depend heavily on occupant behavior and building characteristics, such as construction materials and ventilation systems [9]. A major challenge in linking indoor air pollution to health outcomes lies in the vast diversity of both pollutants and their associated diseases. Consequently, most studies tend to focus either on a single disease, such as acute lower respiratory infections [11], or on a single pollutant [3].

As a result, comprehensively linking building parameters and indoor pollutant concentrations to specific pathologies remains a complex endeavor. The availability of large, heterogeneous, and evolving datasets makes computational approaches highly relevant for addressing this gap. For instance, machine learning models have been employed to predict associations between respiratory diseases, air pollution, and climatic factors [8]. In the domain of health-related link prediction, Knowledge Graphs [5] have emerged as a commonly used and powerful tool ; for example, they have been successfully utilized to map connections between pesticides and diseases [16] or outdoor pollutants and diseases [7]. Applying Graph Neural Networks (GNNs) to these knowledge structures represents a highly promising avenue for discovering complex associations. Although GNN-based methods have already been introduced in the field of air pollution, their applications have primarily been restricted to air quality forecasting rather than health outcome prediction [14].

2 Methodology

2.1 Selecting and Cross-linking Relevant Data Sources

2.1.1 Data Sources Used

To effectively capture and model the complex relationships between pollutant exposure and health outcomes, three distinct databases were integrated into our study :

— **Clinical Data from the French National Health Data System (SNDS) :** Sourced from the open-access portal, this aggregated dataset contains patient counts, reference populations, and prevalence rates for 79 distinct pathological categories. The data is rigorously stratified by biological sex, five-year age cohorts, and administrative departments.
— **The Comparative Toxicogenomics Database (CTD) :** A robust resource linking chemical exposures to biological outcomes [2]. It provides both direct and inferred chemical-disease associations, with the latter being deduced from complex interactions involving genes, phenotypes, and biological pathways.
— **Residential Exposure and Building Characteristics :** Detailed building-specific data and simulated indoor residential exposures were provided by the French Scientific and Technical Centre for Building (CSTB).

2.1.2 Mapping

A primary technical challenge involved harmonizing the health nomenclature systems between the CTD and the SNDS datasets through an integration process [12]. The CTD database relies on the Medical Subject Headings (MeSH) thesaurus—a hierarchically organized controlled vocabulary maintained by the U.S. National Library of Medicine—and occasionally employs the Online Mendelian Inheritance in Man (OMIM) coding system. In contrast, the SNDS classifies health outcomes into 79 distinct categories defined by ICD-10 (International Classification of Diseases, version 10), CCAM (Common Classification of Medical Acts), and GHM (Homogeneous Patient Groups) codes. To perform the cross-dataset mapping, the Unified Medical Language System (UMLS) Metathesaurus browser [1]facilitated the alignment of ICD-10 codes with their corresponding MeSH terms. To preserve the clinical semantic context, the hierarchical tree structure of MeSH was maintained, ensuring that each disease node remained accurately linked to its parent concept.

2.2 Knowledge Graph Construction

A Knowledge Graph (KG) [5] was designed and developed to integrate the three aforementioned datasets. Within this architecture, nodes are assigned explicit semantic classes—such as Pollutant, Gene, Disease, and Housing Unit—thereby preserving the inherent heterogeneity of the underlying data. The edges represent directed, typed relationships that encode specific biological or environmental mechanisms, rather than mere co-occurrences. A defining characteristic of this model is that relationships between identical node types can vary significantly ; for instance, a pollutant may exhibit an up-regulation (increased expression) relationship with a specific gene, while also potentially exhibiting a down-regulation (decreased expression) association under different conditions.

A central challenge in this framework is establishing a biologically and environmentally plausible link between simulated indoor pollutant concentrations and observed hospitalization trends. To bridge this gap, a mediator node designated as "Exposure Effect" was introduced. This node functions as a logical trigger : an association is only activated when the simulated residential pollutant concentration exceeds the critical threshold reported in the corresponding toxicological literature.

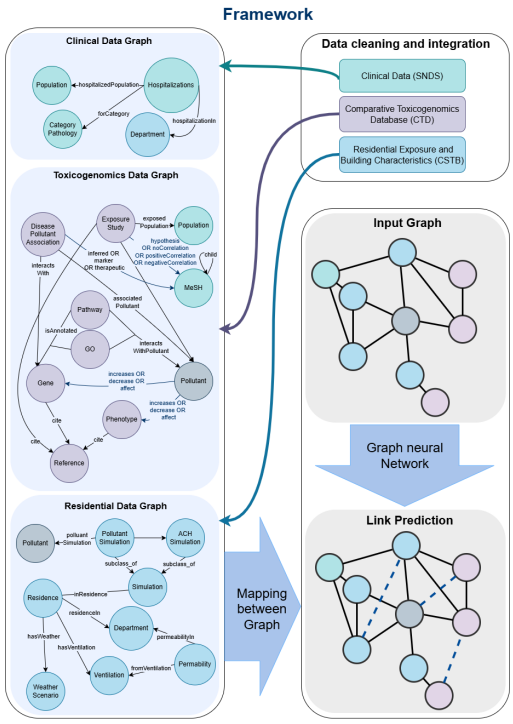


FIGURE 1 – Framework for link prediction integrating different data sources

An additional complexity involves the temporal constraints governing pollutant exposure and subsequent hospital admission. To resolve this limitation, a directed relationship was established between the Exposure Effect node and Hospitalization events, strictly conditioned on the exposure chronologically preceding the hospital admission. To better understand temporal constraints, one perspective is to use a temporal graph neural network [10].

2.3 Learning Over the Knowledge Graph

Given that the developed knowledge graph contains multiple node and edge types encoding rich semantic information, it is formally modeled as a heterogeneous graph. It is then required to translate this graph into a set of dense vectors that could then be used for the learning process. Therefore, node embeddings are first initialized as 64-dimensional vectors, without incorporating external node features. A single layer of a Heterogeneous Graph Neural Network (HeteroGNN) is then applied to learn node representations. This layer is implemented via PyTorch Geometric (PyG), an advanced graph learning library built upon the PyTorch framework. The network utilizes relation-specific message passing facilitated by a Hetero-

Conv layer. This architecture allows each distinct relation type within the heterogeneous graph to be processed independently, subsequently aggregating the extracted features to generate enriched node representations [13]. The convolutional operations are grounded in the GraphSAGE framework, which iteratively updates node embeddings by aggregating features from local neighborhoods, thereby effectively leveraging the structural topology of the graph for robust representation learning [4]. Negative sampling [15] is performed uniformly at random, which is suboptimal for Knowledge Graphs. This opens avenues for improvement, such as hard negative sampling techniques.For evaluation, edge scores are computed via a dot product between source and destination node embeddings, and optimized using binary cross-entropy with logits against the true labels.

3 Preliminary Results

The constructed knowledge graph comprises 3,462,139 edge instances and 536,924 nodes.
The GNN model was trained for 200 epochs, focusing exclusively on a specific edge type : the *associatedPollutant* relationship, which connects *DiseasePollutantAssociation* nodes to *Pollutant* nodes. This relationship was the only one using during training due to computational resources and time constraints.
The edges are split into training, validation, and test sets with proportions of 80%, 10%, and 10%, respectively. The model converged to a low training loss of 0.0857, indicating an effective extraction and learning of structural patterns within the training data. The validation loss stabilized at 0.5364, showing that there is gap between training and validation performance that may indicate a lack of ability to generalize. Furthermore, the model achieved a validation AUC (Area Under the Receiver Operating Characteristic Curve) of 0.8473 . This metric demonstrates a strong discriminative capability in binary edge classification, effectively distinguishing between positive and negative edges (i.e., the presence or absence of a valid link within the knowledge graph).

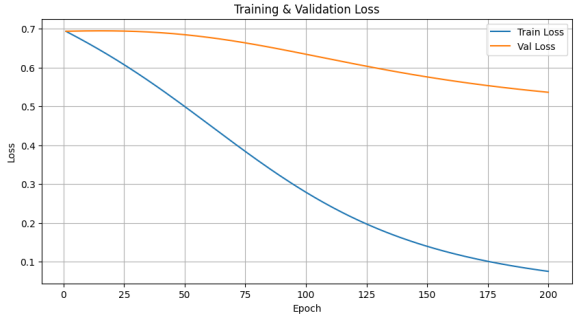


FIGURE 2 – Training and validation loss performance of the graph neural network model across epochs

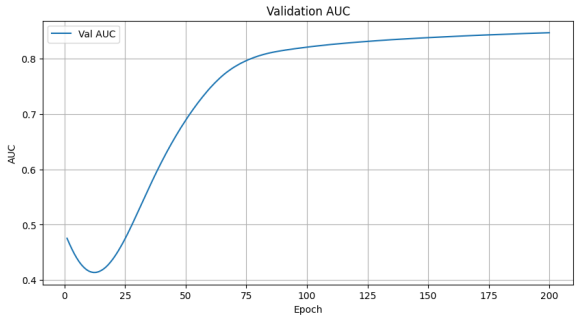


FIGURE 3 – Validation Area Under the Curve across epochs

4 Conclusion and Future Directions

Through the integration of heterogeneous data from environmental, toxicogenic, and clinical sources into a unified Knowledge Graph, the current work enables a comprehensive exploration of the relationships between these interlinked domains. The data used present certain limitations, as they do not provide a complete representation. For instance, the building data only include residential buildings. However, the proposed method, based on a GNN approach, enables effective link prediction within the knowledge graph.

Preliminary results demonstrate an ability to identify existing links, as shown by a validation AUC of 0.8473. However, the comparison between the training loss (0.0857) and validation loss (0.5364) suggests that while the GNN successfully learns structural information, its capacity for generalization remains limited. To improve the generalization capability of the GNN model, several strategies can be employed, including dropout techniques such as DropEdge, which randomly removes edges instead of nodes, as well as regularization methods such as weight decay.

Future work will focus on improving both the embeddings and the GNN architecture. Additionally, the environmental dataset will be expanded to include thermal comfort parameters. Another avenue of interest is the creation of a baseline, by comparing GNN to other models such as regression models, other Knowledge Graph embedding (i.e., TransE, RotatE), or other Neural Networks models (like R-CGN and Edge Transformers).

Références

[1] Olivier Bodenreider. The unified medical language system (umls) : integrating biomedical terminology. *Nucleic Acids Res.*, 32(Database-Issue) :267–270, 2004.

[2] Allan Peter Davis, Caroline G. Murphy, Cynthia A. Saraceni-Richards, Michael C. Rosenstein, Thomas C. Wiegers, and Carolyn J. Mattingly. Comparative toxicogenomics database : a knowledgebase and discovery tool for chemical-gene-disease networks. *Nucleic Acids Research*, 37(Database issue) :D786–D792, Jan 2009.

[3] A. Garcia, E. Santa-Helena, A. De Falco, J. de Paula Ribeiro, A. Gioda, and C. R. Gioda. Toxicological effects of fine particulate matter (pm2.5) : Health risks and associated systemic injuries—systematic review. *Water, air, and soil Pollution*, 234(6), May 2023.

[4] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc.

[5] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4), July 2021.

[6] Health Effects Institute. State of global air 2024, 2024.

[7] Nareesa Karmali, Abdougafarou Mamam, and Gayo Diallo. Outdoor air quality andnbsp ;health impact : Thenbsp ;panorama knowledge graph based approach. In *Computational Collective Intelligence : 17th International Conference, ICCCI 2025, Ho Chi Minh City, Vietnam, November 12–15, 2025, Proceedings, Part II*, page 32–46, Berlin, Heidelberg, 2025. Springer-Verlag.

[8] Y. Ku, S. B. Kwon, J.-H. Yoon, S.-K. Mun, and M. Chang. Machine learning models for predicting the occurrence of respiratory diseases using climatic and air-pollution factors. *Clinical and Experimental Otorhinolaryngology*, 15(2) :168–176, May 2022.

[9] P. Kumar, A.B. Singh, T. Arora, S. Singh, and R. Singh. Critical review on emerging health effects associated with the indoor air quality and its sustainable management. *Science of The Total Environment*, 872 :162163, May 2023.

[10] Antonio Longa, Veronica Lachi, Gabriele Santin, Monica Bianchini, Bruno Lepri, Pietro Lio, Franco Scarselli, and Andrea Passerini. Graph neural networks for temporal graphs : State of the art, open challenges, and opportunities, 2023.

[11] H. Nair, D. J. Nokes, B. D. Gessner, M. Dherani, S. A. Madhi, R. J. Singleton, K. L. O'Brien, A. Roca, P. F. Wright, N. Bruce, A. Chandran, E. Theodoratou, A. Sutanto, E. R. Sedyaningsih, M. Ngama, P. K. Munywoki, C. Kartasasmita, E. A. Simões, I. Rudan, M. W. Weber, and H. Campbell. Global burden of acute lower respiratory infections due to respiratory syncytial virus in young children : a systematic review and meta-analysis. *Lancet*, 375(9725) :1545–1555, May 2010.

[12] Inès Osman, Sadok Ben Yahia, and Gayo Diallo. Ontology integration : Approaches and challenging issues. *Information Fusion*, 71 :38–63, 2021.

[13] R. Ragesh, S. Sellamanickam, A. Iyer, R. Bairi, and V. Lingam. Hetegcn : Heterogeneous graph convolutional networks for text classification. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 860–868, New York, NY, USA, 2021. Association for Computing Machinery.

[14] J. Xu, S. Wang, N. Ying, X. Xiao, J. Zhang, J. Zhiling, Y. Cheng, and G. Zhang. Dynamic graph neural network with adaptive edge attributes for air quality prediction : A case study in china. *Heliyon*, 9 :e17746, 07 2023.

[15] Zhen Yang, Ming Ding, Chang Zhou, Hongxia Yang, Jingren Zhou, and Jie Tang. Understanding negative sampling in graph representation learning, 2020.

[16] D. Zhang, X. Wu, P. Chen, Q. Wang, Y. Li, C. Zhai, and G. Hao. Knowledge-driven pesticide repurposing via link prediction with pesticide graph embedding, 01 2025.

Détection de la désactivation des LFP dans le système neuromusculaire de macaques lors d'une tâche "Reach and Grasp" par l'apprentissage machine. 2026

Hedi Zeghidi[1], Florian Chambellant[1], Ian Moreau-Debord[2], Eleonore Serrano[2], Stephan Quessy[2], Numa Dancause[2],Elizabeth Thomas[1]

[1]Université Bourgogne Europe, INSERM CAPS UMR 1093, 21000, Dijon, France
[2] Université de Montréal, Département de neurosciences, Faculté de médecine, Montréal (Québec)

28 janvier 2026

Résumé

Les lésions cérébrales, comme les AVC, perturbent l'activité neuromusculaire et entraînent des déficits moteurs majeurs. Nous évaluons plusieurs méthodes d'apprentissage automatique appliquées sur les potentiels de champ locaux (LFP) chez le macaque aux stades très précoces d'une inactivation cérébrale focale et réversible. L'apprentissage automatique permet de distinguer les essais avec et sans inactivation à partir des amplitudes spectrales et des décalages spectraux. Ces résultats, obtenus alors que l'animal pouvait encore saisir des objets, suggèrent une détection très précoce de l'inactivation neuronale grâce aux enregistrements extracellulaires.

Mots-clés

Intelligence artificielle ; tâche d'atteindre et de saisir ; potentiel de champ local multielectrode ; système neuromusculaire ; singe macaque

Abstract

Brain injuries such as stroke disrupt neuromuscular activity and cause significant motor impairments. In this study, we evaluate several machine learning methods using local field potentials (LFPs) in macaques at very early stages after focal, reversible brain inactivation. Machine learning was used to distinguish trials with and without inactivation using power spectral density amplitudes and spectral shifts. Because predictions were made shortly after inactivation—while the monkeys could still perform reach-and-grasp tasks—these results suggest the potential for very early detection of neuronal inactivation from extracellular field recordings.

Keywords

Artificial Intelligence ; Reach-and-Grasp Task ; Multielectrode Local Field Potential ; Neuromuscular System ; Macaque Monkey

1 Introduction

Les lésions cérébrales, telles que les accidents vasculaires cérébraux (AVC) ou les infarctus cérébraux, représentent aujourd'hui l'une des principales causes de mortalité dans le monde. Selon l'Organisation mondiale de la santé (OMS), en 2021, les lésions cérébrales figuraient parmi les principales causes de décès et d'invalidité à l'échelle mondiale, avec 11,9 millions de nouveaux cas d'AVC. Ces affections entraînent des altérations des capacités cognitives, telles que la prise de décision, la planification ou l'organisation, ainsi que des déficits moteurs, avec une démarche ralentie et des troubles de l'équilibre.

Sur le plan moteur, les muscles post-lésionnels présentent une capacité réduite à se contracter de manière isolée, ce qui favorise l'apparition de synergies anormales entre fléchisseurs et extenseurs [11, 2]. Au niveau de la main, les prises puissantes, nécessitant peu d'individualisation des doigts, sont privilégiées [7, 13], tandis que les mouvements nécessitant une coordination interarticulaire fine sont souvent altérés [1, 6].

Dans des études précédemment réalisées, comme celle de [4], l'analyse des variations des corrélations des activités neuronales des LFP montrait une concentration de l'activité dans la bande de fréquence Delta. Ces études ont également mis en évidence des changements dans les corrélations lors des différentes phases du mouvement. De plus, ces variations étaient également visibles lorsque l'on identifiait les groupes de neurones à l'aide d'un hierarchical clustering, permettant d'observer l'évolution de l'activité neuronale au fil des différentes étapes du mouvement. Cette étude a appliqué donc des méthodes d'intelligence artificielle à des cerveaux de singes sains pour regarder l'évolution ; une question demeure toutefois quant à la possibilité d'étendre ces approches à d'autres méthodes plus avancées comme l'apprentissage profond à des cerveaux de singes présentant des déficits.

Cette étude vise donc à évaluer des méthodes récentes d'apprentissage automatique pour détecter les altérations de l'activité neuronale consécutives à une lésion cérébrale. La détection repose sur la classification de l'activité neuronale enregistrée lors d'une tâche de Reach-and-Grasp chez le singe, comparée à celle mesurée 30 minutes après l'injection d'un agent pharmacologique simulant une lésion cérébrale. Différents modèles, allant des SVM à l'apprentissage profond, ont été entraînés sur plusieurs phases de la tâche et selon diverses méthodes de traitement, montrant des performances satisfaisantes selon plusieurs métriques.

2 Setup

2.1 Modèle expérimental

Les données ont été enregistrées chez deux macaques rhésus femelles (Macaca mulatta), le singe M (5,5 kg) et le singe S (5,7 kg), selon des procédures chirurgicales et comportementales déjà décrites [8]. Des réseaux multielectrodes ont été implantés dans les régions prémotrices ventrale (PMv) et dorsale (PMd) des deux hémisphères, ainsi que dans le cortex moteur primaire (M1) de l'hémisphère droit. Les signaux LFP ont été échantillonnés à 2035 Hz. Les LFP ont été extraits par filtrage entre 0 et 500 Hz, tandis que les potentiels d'action ont été filtrés entre 100 et 5000 Hz.

2.2 Tâche expérimentale

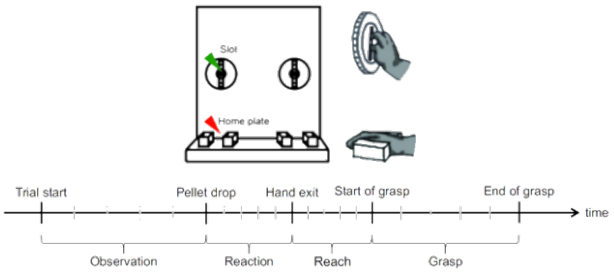


FIGURE 1 – Configuration expérimentale (image adaptée de [4]) illustrant les différentes étapes de l'essai : commencement de l'essai (Trial Start), dépôt de la boulette (Pellet Drop), sortie de la main (Hand Exit), début de la saisie (Start of Grasp) et fin de la saisie (End of Grasp).

Les singes étaient assis sur une chaise expérimentale adaptée aux primates, placée face à un distributeur de granulés alimentaires. Selon le bloc expérimental, une ouverture gauche ou droite était ouverte afin que le singe récupère une boulette de nourriture tombant dans un puits situé derrière une fente. La fente pourra s'orienter horizontalement ou verticalement, cette orientation variant selon les blocs, imposant une supination ou une pronation de la main à 90° (voir la Figure 1 avec la fente verticale).
Un essai commençait lorsque l'animal posait sa main sur la plaque d'accueil, située à 15 cm sous la fente à granulés. Après un intervalle aléatoire de 800 ms à 2 s, une boulette de nourriture (Pellet Drop) était déposée automatiquement dans le puits. Le bruit associé à cette déposition constituait le signal de départ, après lequel l'animal disposait de 2 s pour retirer sa main de la plaque d'accueil (Hand Exit) et prendre la boulette. Ce positionnement de la main au départ et à la sortie était détecté par un premier capteur laser infrarouge.
Le moment où l'animal introduisait sa main dans la fente pour saisir la boulette (Start of Grasp) et celui où il retirait sa main de la fente (End of Grasp) étaient détectés par un deuxième capteur laser infrarouge. Le singe portait la boulette à sa bouche et replaçait sa main sur la plaque d'accueil, avec un intervalle de trois secondes entre chaque essai. Les animaux répétaient la tâche 25 fois pour chaque main et chaque orientation de la fente (blocs randomisés).
À l'issue de cette phase, un agent pharmacologique, le muscimol, simulant une lésion cérébrale, était injecté dans le cortex moteur primaire (M1) de l'hémisphère gauche. Après l'injection, le singe réalisait à nouveau 100 essais selon le même protocole qu'avant l'injection. Chaque singe a répété ce protocole à plusieurs reprises. Après le prétraitement, nous avons obtenu environ 1 200 essais répartis entre les deux classes.

2.3 Prétraitement

Les données ont été traitées à l'aide de scripts Matlab ([10]) et de FieldTrip ([9]). Les essais incorrects ont été exclus lorsque le temps de réaction était inférieur à 200 ms, la durée de l'atteinte supérieure à 350 ms ou la saisie inférieure à 200 ms. Les signaux LFP ont ensuite été filtrés par un filtre Butterworth passe-bas (200 Hz, ordre 6), puis le bruit à 60 Hz a été supprimé par un filtre coupe-bande (59–61 Hz). Les artefacts, avec des rafales irrégulières et des oscillations anormales de basse fréquence, qui ont été retirés à l'aide d'une analyse en composantes indépendantes (ICA) [3, 12].

3 Methodology

3.1 Modèles

Decision Tree . Un « Decision Tree » est un modèle d'apprentissage automatique supervisé, utilisé pour la classification ou la régression. Il prédit les résultats en divisant les données de manière récursive selon les valeurs de leurs variables, créant ainsi une structure en arbre de règles de décision : chaque nœud interne correspond à une condition sur une caractéristique, chaque branche représente l'issue de cette condition, et chaque feuille fournit la prédiction finale.

SVM . Une Machine à Vecteurs de Support (SVM) est un algorithme d'apprentissage supervisé utilisé pour la classification. Elle fonctionne en trouvant un hyperplan qui sépare les classes de données et en choisissant cet hyperplan pour maximiser la marge, c'est-à-dire la distance entre l'hyperplan et les points les plus proches de chaque classe, appelés vecteurs de support.

Random Forest . Random Forest est un algorithme d'apprentissage automatique supervisé qui construit une collection d'arbres de décision à l'aide de données d'entraînement rééchantillonnées de manière aléatoire et de caractéristiques sélectionnées de manière aléatoire, puis combine leurs prédictions individuelles pour produire un résultat final.

Modèle KNN . L'algorithme K-Nearest Neighbors (KNN) est une méthode de classification supervisée basée sur l'idée que les points de données similaires ont tendance à appartenir à la même classe (« qui se ressemble s'assemble »). Il prédit la classe de nouvelles données en examinant les classes des K points de données les plus proches dans l'espace des caractéristiques et en attribuant la classe la plus courante parmi celles-ci.

Modèle CNN . Un réseau neuronal convolutif (CNN) est un type de réseau neuronal à propagation directe conçu pour apprendre et extraire automatiquement des caractéristiques hiérarchiques à partir de données, généralement des images, en appliquant des filtres convolutifs (ou noyaux) qui glissent sur l'entrée pour capturer des motifs tels que les contours, les textures et les formes, les filtres appris étant optimisés pendant l'entraînement afin d'améliorer les performances spécifiques à la tâche.

Vision Transformer . *Vision Transformer* (ViT) sont un type d'architecture de réseau neuronal pour la vision par ordinateur qui adapte le modèle des transformers, initialement développé pour le traitement du langage naturel, aux données d'images. Plutôt que de traiter l'image dans son ensemble, les ViT la divisent en patchs de taille fixe, transforment chaque patch en vecteur (embedding) et traitent ensuite la séquence de ces vecteurs à l'aide de mécanismes d'auto-attention. L'auto-attention permet au modèle de pondérer l'importance relative de chaque patch par rapport aux autres, en capturant les relations et dépendances à longue portée entre différentes parties de l'image. Cette approche permet au modèle d'apprendre des représentations riches et globales, utiles pour des tâches telles que la classification d'images, la détection d'objets ou la segmentation.

3.2 Metrics

Pour évaluer la capacité des modèles à détecter la désactivation, nous avons utilisé plusieurs indicateurs de performance, notamment la précision, le recall, le F1-score et l'accuracy. Ces indicateurs ont été choisis en raison du léger déséquilibre entre les deux classes (55% pour la classe non désactivée et 45% pour la classe désactivée) et afin d'obtenir une vision globale des performances. Ils sont dérivés de la matrice de confusion, qui résume les résultats d'une tâche de classification binaire comportant deux classes : positive et négative.

Précision . La précision quantifie la fiabilité des prédictions d'une classe donnée, en mesurant la proportion de vrais positifs parmi tous les positifs prédits. Elle est définie comme suit :

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall . Recall quantifie la capacité d'un modèle à identifier toutes les instances d'une classe donnée, en mesurant la proportion d'instances positives réelles qui ont été correctement prédites. Il peut être exprimé comme suit :

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-score . Le F1-score représente la moyenne de la précision et du rappel, fournissant ainsi une mesure unique qui équilibre les deux. Il est calculé comme suit :

$$\text{F1-score} = \frac{2 * TP}{2 * TP + FP + FN}$$

Accuracy .Accuracy mesure l'exactitude des prédictions du modèle en calculant la proportion de résultats corrects parmi le nombre total de résultats.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

3.3 Configuration d'entraînement

Dans l'ensemble de nos modèles classiques, les données ont été réparties en 80% pour l'entraînement et 20% pour le test. Les procédures d'entraînement variaient selon les modèles. Pour les modèles KNN, SVM et Random Forest, l'entraînement a été réalisé avec une validation croisée et une optimisation des hyperparamètres, en choisissant le F1-score comme critère pour déterminer les meilleurs paramètres.

Pour la première approche de classification, nous avons entraîné des modèles d'apprentissage profond, l'ensemble d'entraînement a été subdivisé en 60% pour l'entraînement, 20% pour la validation et 20% pour le test. Cette répartition offre un équilibre classique entre capacité d'apprentissage et robustesse de l'évaluation. Le modèle a été sauvegardé durant l'entraînement chaque fois que la validation loss diminuait. Le taux d'apprentissage initial était fixé à 1e-3. Dans l'ensemble des différentes périodes étudiées, nous avons observé une convergence de la loss d'entraînement et de validation après environ une quinzaine d'epochs.
Pour interpréter ces prédictions, nous avons utilisé la méthode Grad-CAM (Gradient-weighted Class Activation Mapping), une technique d'interprétabilité adaptée aux réseaux de neurones convolutionnels, qui permet de visualiser les régions de l'image ayant le plus d'influence sur la prédiction d'une classe donnée. Grad-CAM calcule le gradient de la sortie associée à la classe d'intérêt par rapport aux cartes d'activation d'une couche convolutionnelle, puis utilise ces gradients pour pondérer et combiner les cartes d'activation, produisant ainsi une carte de chaleur mettant en évidence les zones déterminantes. Cette approche rend les décisions des CNN plus transparentes, permet de vérifier que le modèle se concentre sur des caractéristiques pertinentes, de détecter d'éventuels biais ou erreurs, et de communiquer les résultats de manière visuelle et intuitive, ce qui est essentiel dans un contexte scientifique ou biomédical.

Pour la seconde approche de classification, basée sur les spectrogrammes, nous avons calculé la densité spectrale de puissance (PSD) et extrait le classement des amplitudes maximales pour les cinq bandes de fréquences (Delta, Theta, Alpha, Beta et Gamma). Ce classement a ensuite servi à entraîner un Decision Tree et une Random Forest afin d'évaluer si la désactivation neuronale entraînait des modifications dans la hiérarchie des bandes de fréquences.

4 Résultats

Pour détecter la désactivation neuronale, nous avons testé deux approches basées sur les spectrogrammes : l'utilisation directe des spectrogrammes et l'exploitation du classement des amplitudes maximales.

4.1 Détection avec les spectrogrammes normalisés des LFP

Afin d'identifier la désactivation neuronale, les signaux LFP ont été transformés en spectrogrammes couvrant une bande de fréquences de 0,5 à 100,5 Hz, avec un pas de 2 Hz (2,5 ; 4,5, etc.), puis normalisés par rapport à la période d'observation. Les données ont été sélectionnées sur l'ensemble des fréquences et sur des fenêtres temporelles de 500 ms, à différents moments décrits plus loin dans l'article. Les signaux ont ensuite été moyennés par zone d'enregistrement (lPMv, rPMv, lPMd, rPMd, M1).

Les données obtenues étaient ainsi organisées sous forme de tenseurs tridimensionnels, comprenant 5 régions, 51 bandes de fréquences et 50 pas de temps. Différentes phases des essais ont été analysées afin d'évaluer la capacité à détecter la désactivation neuronale à différents moments du comportement : avant la reaction du singe (500 ms avant le Pellet Drop), au début du mouvement (250 ms avant et après la sortie de la main), et avant le début de la prise (500 ms avant le début de la saisie).

4.1.1 Période d'observation

Les résultats présentés dans le tableau 1 indiquent que, pour cette période, les meilleurs scores sont ex æquo pour le modèle CNN et le ViT, avec une accuracy et un F1-score de 77%. On observe par ailleurs qu'il n'existe pas de différence notable entre les valeurs de l'accuracy et du F1-score. À l'inverse, les performances les plus faibles sont obtenues avec le modèle le plus simple, le KNN.

Modèles	Precision	Recall	F1-score	Accuracy
SVM	64%	64%	64%	64%
Random Forest	78%	69%	72%	72%
KNN	52%	52%	52%	53%
Dense Model	53%	53%	52%	52%
CNN Model	77%	76%	77%	77%
ViT	77%	76%	77%	77%

TABLE 1 – Performances de classification des différents modèles pour distinguer l'activation et la désactivation neuronales pour la période d'observation

L'analyse de la matrice de confusion du modèle CNN montre que les erreurs se produisaient surtout lorsque le modèle prédisait qu'il n'y avait pas de désactivation, alors qu'en réalité l'activité neuronale était désactivée. Par ailleurs, l'examen de la dernière couche convolutionnelle du modèle CNN (voir la Figure 2) révèle que les informations les plus discriminantes pour déterminer la désactivation sont localisées juste avant le Pallet Drop entre -125ms et 0ms, et concernent principalement les bandes de fréquences Delta (0,5–4Hz), Theta (4-7Hz), Alpha (7-12Hz), Beta (12-16.5Hz) et Gamma (36.5-42.5Hz et 94.5-100.5 Hz).

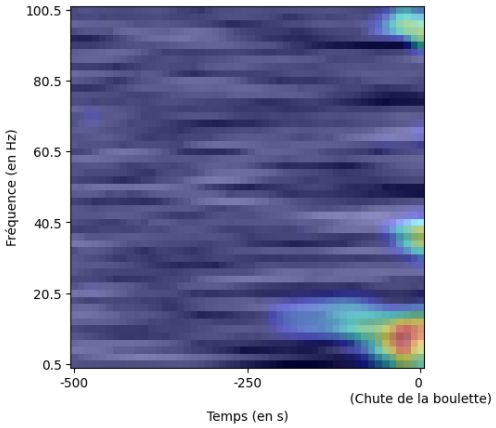


FIGURE 2 – Grad-CAM du modèle CNN pour la classe désactivée, 500 ms avant le dépôt de la boulette

4.1.2 Période de la sortie de la main

Les résultats présentés dans le tableau 2 montrent que, pour cette période, le modèle CNN obtient les meilleures performances, avec une accuracy et un F1-score de 78%.

Models	Precision	Recall	F1-score	Accuracy
SVM	72%	72%	72%	72%
Random Forest	78%	76%	77%	78%
KNN	53%	53%	53%	54%
Dense Model	68%	68%	68%	68%
CNN Model	78%	78%	78%	78%
ViT	76%	76%	76%	76%

TABLE 2 – Performances de classification des différents modèles pour distinguer l'activation et la désactivation neuronales au début de la prise

L'analyse de la dernière couche convolutionnelle du CNN modèle révèle que la période précédant la sortie de la main est la plus informative, principalement dans les basses bandes de fréquences : Delta, Theta et Alpha.

4.1.3 Période du début de la prise

Ici, en considérant les 500 ms précédant le début de la prise, nous observons de meilleurs résultats pour l'ensemble des modèles. Toutefois, ces performances ne dépassent pas celles obtenues sur d'autres périodes. Le meilleur modèle

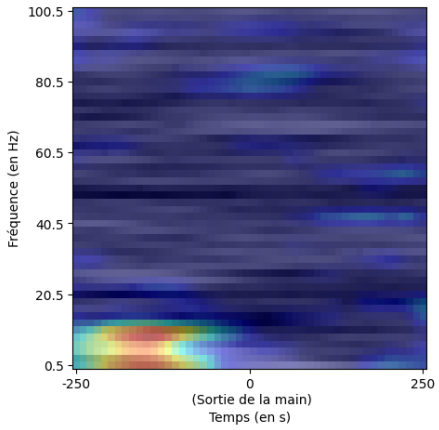


FIGURE 3 – Grad-CAM du modèle CNN pour la classe désactivée, 250ms avant et après la sortie de la main

reste le Random Forest, avec un F1-score de 76% et une accuracy de 78%.

Models	Precision	Recall	F1-score	Accuracy
SVM	69%	69%	69%	69%
Random Forest	79%	76%	76%	78%
KNN	50%	50%	50%	51%
Dense Model	73%	73%	73%	73%
CNN Model	74%	74%	74%	74%
ViT	76%	77%	76%	76%

TABLE 3 – Performances de classification des différents modèles pour distinguer l'activation et la désactivation neuronales pour la période du début de la prise

Si nous analysons la dernière couche convolutionnelle du CNN modèle, nous trouvons que les informations les plus importantes pour déterminer les classes se trouvent dans les basses fréquences de Delta, Theta et Alpha. Et, elles se concentrent surtout entre -375ms et -250ms avant le début de la saisie.

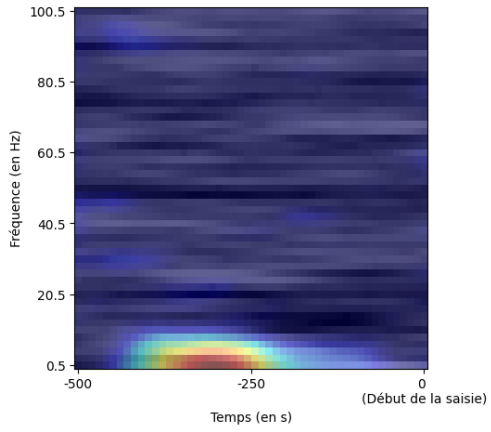


FIGURE 4 – Grad-CAM du modèle CNN pour la classe désactivée, 500ms avant le début de la saisie

4.2 Détection avec le "Power Density Spectral" des LFP

Dans cette partie, nous avons choisi de nous concentrer sur les régions lPMv et lPMd, car elles sont situées à proximité du site d'injection du muscimol. Nous avons analysé la classification selon les différentes phases de l'essai (Observation, Reaction, Reach and Grasp). Pour ces données, seuls des modèles interprétables pour ces données ont été utilisés, à savoir Decision Tree et Random Forest.

Models	Precision	Recall	F1-score	Accuracy
Decision Tree OBS	75%	74%	75%	75%
Decision Tree REACT	64%	64%	63%	63%
Decision Tree REACH	67%	66%	66%	68%
Decision Tree GRASP	65%	65%	65%	66%
Random Forest OBS	75%	75%	75%	76%
Random Forest REACT	71%	71%	71%	72%
Random Forest REACH	69%	68%	69%	70%
Random Forest GRASP	67%	67%	67%	68%

TABLE 4 – Performances des différents modèles utilisant les densités spectrales de puissance (PSD)

En analysant l'ensemble des résultats, on observe que les meilleures performances sont obtenues durant la période d'observation. Les scores se situent entre 65% et 75%, ce qui indique que les PSD présentent des changements notables suite à la désactivation neuronale.

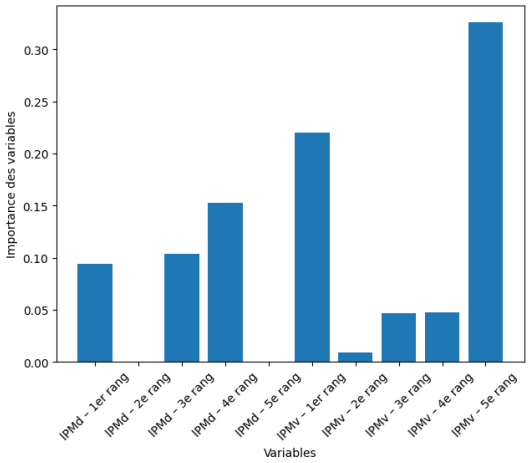


FIGURE 5 – Decision Tree : importance des positions dans les deux classements des zones

Dans la Figure 5, on peut identifier quelles positions dans le classement des bandes de fréquences, établi à partir des amplitudes maximales, influencent le plus la classification. Pour la zone lPMv, les rangs extrêmes — la bande présentant l'amplitude la plus élevée et celle présentant l'amplitude la plus faible — sont les plus déterminants. Pour la zone lPMd, les contributions majeures proviennent des troisième et quatrième bandes de fréquences. Un phénomène similaire est observé avec le modèle Random Forest, comme illustré dans la Figure 6.

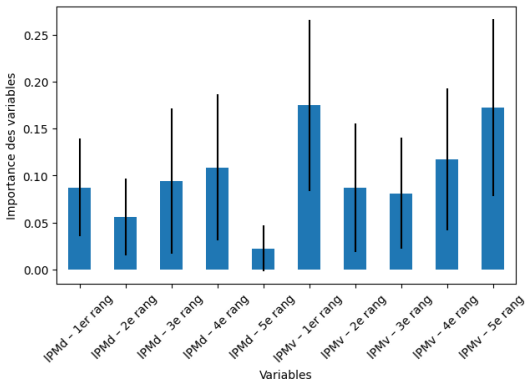


FIGURE 6 – Random forest : importance des positions dans les deux classements des zones

5 Conclusion

Cette étude montre que les méthodes d'apprentissage automatique appliquées aux spectrogrammes de LFP permettent de détecter automatiquement les activités neuronales associées à une inactivation cérébrale focale et réversible. Des informations discriminantes émergent dans les spectrogrammes pour identifier la désactivation, en particulier autour du début du mouvement, et se situent principalement dans les bandes de basses fréquences, telles que delta, theta et alpha. Ces observations sont cohérentes avec les études antérieures de [4, 5], qui indiquaient que l'essentiel de l'activité lors des mouvements de Reach-and-Grasp se concentre dans la bande Delta. D'autres signatures sont également mises en évidence dans les PSD, à travers des variations des amplitudes maximales sur les cinq bandes de fréquences. Ces différences sont particulièrement marquées dans la région lPMv, notamment pour les bandes de fréquences extrêmes (les plus basses et les plus hautes).

Par ailleurs, nous supposons que cette désactivation pourrait aussi être mise en évidence en comparant l'activité neuronale entre les deux hémisphères. Dans le même objectif d'affiner l'analyse, nous prévoyons également d'examiner l'évolution de nos prédictions en nous concentrant sur un seul type de mouvement plutôt que sur l'ensemble des essais ; par exemple, en focalisant l'analyse sur le bras droit, nous nous attendons à observer des différences plus prononcées qu'avec le bras gauche.

Remerciements

Ce travail est financé par une collectivité territoriale dans le cadre du projet CORN. Les données ont été acquises, transmises par l'Université de Montréal, Département de neurosciences, Faculté de médecine, Montréal.

Références

[1] Sarah Astill and Andrea Utley. Coupling of the reach and grasp phase during catching in children with developmental coordination disorder. *Journal of motor behavior*, 40 :315–23, 07 2008.

[2] Benjamin Baak, Otmar Bock, Anna Dovern, Jochen Saliger, Hans Karbe, and Peter H. Weiss. Deficits of reach-to-grasp coordination following stroke : Comparison of instructed and natural movements. *Neuropsychologia*, 77 :1–9, 2015.

[3] Anthony Bell and Terrence Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7 :1129–1159, 11 1995.

[4] Florian Chambellant, Ali Falaki, Ian Moreau-Debord, Robert French, Eleonore Serrano, Stephan Quessy, Numa Dancause, and Elizabeth Thomas. Variations in clustering of multielectrode local field potentials in the motor cortex of macaque monkeys during a reach-and-grasp task. *eneuro*, 11 :ENEURO.0047–24.2024, 09 2024.

[5] Ali Falaki, Stephan Quessy, and Numa Dancause. Differential modulation of local field potentials in the primary and premotor cortices during ipsilateral and contralateral reach to grasp in macaque monkeys. *Journal of Neuroscience*, 44(21), 2024.

[6] Mindy Levin. Interjoint coordination during pointing movements is disrupted in spastic hemiparesis. *Brain : a journal of neurology*, 119 (Pt 1) :281–93, 03 1996.

[7] Sheng Li, Mark L Latash, Guang H Yue, Vlodek Siemionow, and Vinod Sahgal. The effects of stroke and age on finger interaction in multi-finger force production tasks. *Clinical Neurophysiology*, 114(9) :1646–1655, 2003.

[8] Ian Moreau-Debord, Éléonore Serrano, Stephan Quessy, and Numa Dancause. Rapid and bihemispheric reorganization of neuronal activity in premotor cortex after brain injury. *The Journal of Neuroscience*, 41 :9112 – 9128, 2021.

[9] Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen. Fieldtrip : Open source software for advanced analysis of meg, eeg, and invasive electrophysiological data. *Computational intelligence and neuroscience*, 2011 :156869, 01 2011.

[10] The MathWorks Inc. Matlab.

[11] Thomas E. Twitchell. The restoration of motor function following hemiplegia in man. *Brain*, 74(4) :443–480, 12 1951.

[12] Nathan Whitmore and Shih-Chieh Lin. Unmasking local activity within local field potentials (lfps) by removing distal electrical signals using independent component analysis. *NeuroImage*, 132, 02 2016.

[13] Jing Xu, Adrian Haith, and John Krakauer. Motor control of the hand before and after stroke. *Clinical Systems Neuroscience*, pages 271–289, 12 2015.

Evaluation et Analyse explicative d'un modèle de prévision pluie-débit basé sur un Multilayer-Perceptron (MLP), dans le cas de la rivière Sisaony, à Madagascar

Hanitriniaina Marielle RAKOTOZANANY[1], Pierre NICOLLE[2], Josué RATOVONDRAHONA[1], Bob E. SAINT-FLEUR[2], Andry RAZAKAMANANTSOA[3], Samuel RAZANAKA[4], Thomas MAHATODY[1], Olivier PAYRASTRE[2]

[1] Université de Fianarantsoa, LIMAD, BP-1264 Fianarantsoa, Madagascar
[2] Univ Gustave Eiffel, GERS-LEE, F-44344 Bouguenais, France
[3] Univ Gustave Eiffel, GERS-GIE, F-44344 Bouguenais, France
[4] Centre National de Recherche pour l'Environnement, BP-1739 Antananarivo, Madagascar

Résumé

Cette étude évalue les performances d'un modèle pluie-débit basé sur un Multilayer Perceptron (MLP), complétée par une analyse d'eXplainable Artificial Intelligence (XAI) une approche fondée par la perturbation. Le modèle MLP prédit le débit instantanné horaire de la rivière Sisaony, Madagascar . Les résultats d'évaluation montrent que le MLP est performant, en particulier pour les horizons longs. L'analyse du modèle par perturbation montre également que le comportement du modèle est cohérent avec les principes physiques de la relation hydrologique pluie–débit.

Mots-clés

MLP, XAI, approche de perturbation, pluie-débit, rivière Sisaony.

Abstract

This study evaluates the performance of a rainfall-runoff model based on a Multilayer Perceptron (MLP), supplemented by an Explainable Artificial Intelligence (XAI) analysis using a perturbation-based approach. The MLP model predicts the hourly instantaneous discharge of the Sisaony River in Madagascar. The evaluation results show that the MLP performs well, particularly for long-term forecasts. Analysis of the model using perturbation theory also shows that the behavior of the model is consistent with the physical principles underlying the rainfall–discharge relationship.

Keywords

MLP, XAI, perturbation-based approach, rainfall–runoff, Sisaony River.

1 Introduction

Avec l'essor de l'intelligence artificielle, Deep learning s'est imposé comme des outils puissants capables d'extraire automatiquement des relations complexes à partir de grandes bases de données, notamment pour des phénomènes non linéaires. Il occupe une place importante dans la résolution de problèmes liés à l'environnement, mais ces modèles présentent comme limites une interprétabilité réduite[1, 2]. Dans le domaine de l'hydrologie, par exemple, les modèles de deep learning appliqués à la prévision hydrologique sont performants mais considérés comme des modèles "boîte noire"[3]. Dans cette étude, nous avons mis en place un modèle basé sur le deep learning, plus spécifiquement le Multilayer Perceptron (MLP), pour prévoir les débits de crue d'un cours d'eau. Pour maintenir une certaine transparence du modèle, nous avons combiné cette approche à l'utilisation de la technique XAI (Explainable Artificial Intelligence), l'approche de perturbation, pour interpréter le modèle afin de comprendre les paramètres qui influencent le plus le modèle de prédiction. Comme étude de cas, nous avons sélectionné une rivière, la Sisaony, délimitée par la carte à la figure 1, qui traverse la ville d'Antananarivo Madagascar, et qui permet de disposer des historiques d'observations horaires de précipitations et de débits passés. Compte tenu l'exposition de la ville aux risques d'inondation, Antananarivo dispose en effet d'un organisme chargé de la protection contre les inondations, l'Autorité pour la Protection contre les Inondations de la Plaine d'Antananarivo (APIPA), qui gère un système d'annonce de crue basé sur un réseau permanent d'observations pluviométriques et débitmétriques. L'APIPA ne dispose pas en revanche de modèle de prévision hydrologique, l'annonce de crues reposant principalement sur l'expérience des prévisionnistes, ainsi que sur l'analyse et l'observation des événements passés. Cette approche montre toutefois des limites dans une ville régulièrement touchée par les inondations, notamment en cas de rupture de digue.

Dans cette étude, les historiques de données d'observation de l'APIPA ont été mobilisées pour l'apprentissage et l'évaluation de modèles MLP prédisant les débits de crue de la Sisaony à plusieurs horizons (3h, 6h, 9h, 12h, 18h ou 24h). Les performances de ce modèle MLP ont été comparées à un modèle de prévision pluie-débit de type GRP [9] , très

utilisé pour la prévision des crues en France. L'association d'un modèle basé sur le deep learning et d'approches XAI a par ailleurs permis d'analyser la nature des dépendances apprises par le modèle, afin de vérifier s'il repose principalement sur une dynamique autorégressive ou s'il capture effectivement les relations non linéaires complexes entre les variables hydrométéorologiques.

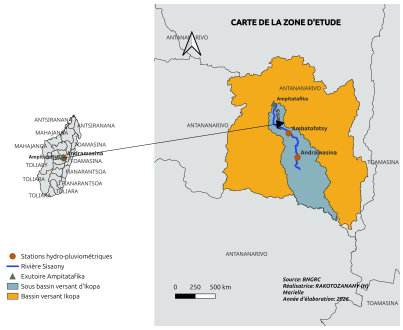


FIGURE 1 – Carte du bassin versant de la rivière Sisaony à Ampitatafika, montrant les stations hydro-pluviométriques situées en amont à Andramasina et Ambatofotsy, ainsi que la station hydrométrique à l'exutoire. La surface drainée du bassin est de 726 km^2

2 Méthodologie

2.1 Données

Un ensemble de données hydrométéorologiques horaires, sur la période 2002 à 2008, a été collecté et utilisé pour alimenter les modèles de prévision de débits hydrologiques. Ces données comprennent les variables suivantes : la précipitations P : issues des stations pluviométriques locales de l'APIPA et moyennées sur le bassin versant de la rivière Sisoany à Ampitatafika à l'aide de la méthode de polygone de Thiessen ; l'évapotranspiration potentielle ETP_1 calculée à partir des données de température extraites de la NASA POWER, en utilisant la méthode de Thornthwaite modifiée[6], l'évapotranspiration potentielle ETP_2 calculée à partir des données de température, extraites de la NASA POWER, en utilisant la méthode d'Oudin [5] ; le débit brut instantané Q mesuré au pas de temps horaire à la station d'Ampitatafika de l'APIPA sur la rivière Sisaony (726 km^2 de surface drainée). Ces données représentent la variable cible à prédire dans le modèle pluie-débit. Après vérification, ces données présentent quelques valeurs manquantes sur certaines périodes. Ces périodes de lacunes ont été reconstituées avec un modèle hydrologique traditionnel GR4H [7] car le résultat de la simulation de débit de ce modèle suit mieux la tendance des données de débits observés.

2.2 Développement du modèle MLP

2.2.1 Variables d'entrée utilisées

Pour estimer le débit $Q(t)$ à un instant t, nous considérons que le débit dépend des précipitations $P(t-i)$ et de l'évapotranspiration potentielle $ETP(t-i)$ observées sur un historique de données ($1 <= i <= p$), ainsi que du débit passé $Q(t-j)$ ($h <= j <= p$). p correspond à la fenêtre temporelle de données prise en compte par le modèle : une fenêtre temporelle glissante de 10 jours consécutifs (soit p=240 heures) a été utilisée ici. h correspond à l'horizon de prévision considéré, soit $h \in 3h, 6h, 9h, 12h, 18h, 24h$. Pour chaque horizon de prévision h, le modèle apprend une relation de la forme : $Q(t) = f(Q(t-j), P(t-i), ETP_1(t-i), ETP_2(t-i))$ où $P(t-i)$ et $ETP_n(t-i)$ (1). Les modèles utilisent donc les informations pluviométriques et d'ETP jusqu'à l'instant t. Pour réaliser de véritables prévisions avec ces modèles, il faut donc être en mesure d'effectuer des prévisions de pluie et d'ETP sur la période [t-h ;t-i]. Dans un premier temps, les données observées ont été utilisées ici, ce qui permet de s'affranchir des incertitudes associées aux prévisions météorologiques.

2.2.2 Découpage de données

Le tableau 1 présente en détail les périodes utilisées pour l'apprentissage et le test des données.

TABLE 1 – Périodes temporelles utilisées pour l'apprentissage et l'évaluation des modèles

Variables d'entrées	Variables cibles	Apprentissage/ Validation	Test
Précipitation, ETP Oudin, ETP Thornthwaite, Débit brut reconstitué avec GR4H	Débit brut reconstitué avec GR4H	Début : 11/01/2002 00 :00 Fin : 10/08/2007 13 :00	Début : 10/08/2007 14 :00 Fin : 31/12/2008 23 :00

2.2.3 Architecture et implémentation du modèle MLP

Un MLP est défini comme un réseau neuronal artificiel composé de trois (3) couches connectées : une couche d'entrée (input layer) qui reçoit les données d'entrée, une ou plusieurs couches cachées (hidden layers) qui contiennent un nombre variable de neurones connectés à tous les neurones de la couche précédente et de la suivante, et une couche de sortie (output layer) qui produit les valeurs de sortie [4]. Pour la création du modèle MLP, les valeurs suivantes d'hyperparamètres ont été retenus après une série de tests de sensibilité : trois (3) couches cachées (120-90-60 neurones), la fonction d'activation $ReLU$, la fonction d'optimisation $Adam$, la fonction coût MSE, le taux d'apprentissage 0.001, l'itération max 200. Pour chaque horizon de prévision h ($3h, 6h, 9h, 12h, 18h$ et $24h$), un modèle MLP distinct, partageant la même architecture et les mêmes hyperparamètres, a été entraîné séparément.

2.3 Le modèle GRP utilisé comme référence

Le modèle GRP est un modèle hydrologique de prévision, se servant des données des pluies disponibles sur un bassin versant pour calculer les débits à son exutoire. GRP est actuellement utilisé par une grande partie des Services de Prévisions de Crues (SPC) français [9]. GRP a une structure à réservoir construite sur trois modules : modèle d'accumulation et de fonte de la neige, le module de production et le modèle de transfert. Le critère de calage (apprentissage) pour GRP est l'erreur quadratique moyenne à l'horizon H

(RMSEH).

2.4 Critères utilisés pour l'évaluation et la comparaison des modèles

Deux critères ont été utilisés, pour l'apprentissage et le test des modèles : le critère de Nash-Sutcliffe (NSE) et le score de persistance. NSE est défini par la formule (2) : $NSE = 1 - \frac{\sum_{t=1}^{n}(Q_o^t - Q_m^t)}{\sum_{t=1}^{T}(Q_o^t - \overline{Q_o})}(2)$ où Q_o^t et Q_m^t sont respectivement les débits observés et simulés (ou prévus) à l'instant t, et $\overline{Q_o}$ est la moyenne des débits observés. Le critère de persistance est défini par la formule (3) suivante : $PI = \frac{\sum_{t=h+1}^{n}(Q_t^{obs} - Q_t^{prev})}{\sum_{i=1}^{n-H}(Q_t^{obs} - Q_{t-h}^{obs})}(3)$ avec Q_i^{prev} le débit prévu au pas de temps i et h l'horizon de prévision (exprimé en nombre de pas de temps). Une valeur (NSE ou persistance) proche de 1 indique une excellente correspondance entre les données simulées et observées, tandis qu'une valeur inférieure à 0 indique que le modèle est moins performant qu'une simple moyenne constante.

2.5 Analyse explicative du modèle pluie-débit MLP

L'importance des variables a été évaluée à l'aide d'une méthode de perturbation consistant à remplacer successivement chaque variable explicative soit par la valeur zéro, soit par sa moyenne. Cette approche permet d'analyser la sensibilité du modèle à différentes entrées en quantifiant la baisse de ses performances. L'analyse est réalisée à deux niveaux de granularité : l'importance par groupe de variables et l'importance individuelle, et ce pour plusieurs horizons de prévision (de 3 à 24 heures).

3 Résultats

3.1 Évaluation des modèles

Les tableaux 2 et 3 présentent respectivement les scores de NSE et de persistance obtenus par les deux modèles MLP et GRP, pour les différents horizons de prévision.

Le modèle MLP atteint des scores NSE très élevés, en particulier pour les horizons courts (3h à 6h), où les valeurs dépassent 0,95 en apprentissage et en test (jusqu'à 0,98). Cela traduit une excellente capacité à reproduire les débits observés et une forte stabilité du modèle. Pour les horizons intermédiaires (9h–12h), les scores restent élevés (0,84–0,94 en test), confirmant une bonne performance du modèle. En comparaison, le modèle GRP présente également de bonnes performances, mais légèrement inférieures à celles du MLP, notamment pour les horizons longs (HP 24h–18h), où les scores chutent en phase de test autour de 0,66–0,74, contre 0,73–0,78 pour le MLP.

L'analyse du score de persistance confirme que le MLP reste globalement plus performant que le GRP, notamment pour les horizons longs (0,32–0,47, en validation contre 0,26–0,33 pour le GRP). Toutefois, les performances déclinent rapidement lorsque l'horizon se raccourcit, avec des scores proches de zéro pour l'horizon 3h, indiquant qu'aucun des modèles n'arrive à surpasser efficacement la persistance sur cet horizon courts. Ceci conduit à relativiser fortement le pouvoir prédictif des modèles pour cet horizon.

TABLE 2 – Comparaison des score NSE des modèles MLP et GRP selon l'horizon de prévision

	MLP		GRP	
Horizon	Apprentissage	test	Apprentissage	test
3h	0.9814	0.9768	0.9865	0.9772
6h	0.9621	0.9479	0.9623	0.9366
9h	0.9482	0.8937	0.9376	0.8872
12h	0.9347	0.8517	0.9149	0.8334
18h	0.9126	0.7597	0.8824	0.7381
24h	0.9061	0.7348	0.8597	0.6633

TABLE 3 – Comparaison des scores de persistance des modèles MLP et GRP selon l'horizon de prévision

	MLP		GRP	
Horizon	Apprentissage	Test	Apprentissage	Test
3h	0.2977	0.0288	0.2441	0.0450
6h	0.4695	0.2635	0.3832	0.1044
9h	0.5959	0.2108	0.4702	0.1624
12h	0.6538	0.2900	0.5275	0.2023
18h	0.7124	0.3269	0.6050	0.2664
24h	0.7685	0.4723	0.6506	0.3301

3.2 Analyse explicative du modèle pluie-débit MLP

Les tableaux 4 et 5 présentent les résultats de l'analyse explicative du modèle pluie-débit basé sur MLP par l'approche de perturbation. La variable de débit passé (Q) apparaît comme la plus influente, avec des valeurs d'importance systématiquement plus élevées en valeur absolue, en particulier pour les horizons courts et intermédiaires, ce qui confirme son rôle structurant dans la dynamique du modèle pluie–débit.

TABLE 4 – Évaluation de l'importance des variables à l'aide d'une approche fondée sur la perturbation, consistant à remplacer chaque variable par zéro.

	Apprentissage			
Horizons	P	ETP1	ETP2	Q
3h	0,001	-0,017	-0.008	-0,381
6h	-0,005	-0,019	-0,047	-0,404
9h	-0,028	-0,0349	-0,027	-0,34
12h	-0,053	-0,060	-0,056	-0,343
18h	-0,015	0,007	0,002	-0,080
24h	-0,105	-0,168	-0,202	-0,176
	Test			
Horizons	P	ETP1	ETP2	Q
3h	-0,012	-0,05	-0,026	-0,504
6h	-0,037	-0,024	-0,024	-0,571
9h	-0,043	-0,058	0,024	-0,508
12h	-0,076	0,014	0,075	-0,533
18h	-0,015	0,054	0,05	-0,099
24h	-0,208	0,026	0,138	-0,412

TABLE 5 – Évaluation de l'importance des variables à l'aide d'une approche fondée sur la perturbation, consistant à remplacer chaque variable par sa moyenne.

	Apprentissage			
Horizons	P	ETP1	ETP2	Q
3h	0	0,008	0,005	-0,279
6h	-0,004	-0,004	-0,005	-0,327
9h	-0,023	-0,009	-0,007	-0,252
12h	-0,052	-0,026	-0,029	-0,215
18h	-0,172	-0,046	-0,055	-0,223
24h	-0,1	-0,044	-0,074	-0,144
	Apprentissage			
Horizons	P	ETP1	ETP2	Q
3h	-0,012	-0,027	-0,019	-0,35
6h	-0,033	-0,024	-0,016	-0,432
9h	-0,035	0,003	0,023	-0,328
12h	-0,061	0,012	0,018	-0,273
18h	-0,144	-0,002	0,009	-0,247
24h	-0,192	0,055	0,047	-0,179

Les précipitations (P) montrent une influence plus marquée aux horizons longs (24h et 18h), tandis que leur impact diminue progressivement lorsque l'horizon de prévision se rapproche de 3h. Les variables d'évapotranspiration (ETP1 et ETP2) présentent une contribution globalement plus faible et plus instable, avec des valeurs proches de zéro dans plusieurs configurations, ce qui suggère une influence secondaire dans la prédiction. La comparaison entre les deux méthodes de perturbation montre que le remplacement par la moyenne conduit à des amplitudes d'importance légèrement atténuées mais conserve les mêmes tendances globales, ce qui renforce la robustesse des conclusions sur le rôle relatif des variables dans le modèle MLP.

4 Conclusion

L'étude démontre que le MLP constitue une méthode robuste pour la prévision des débits hydrologiques à horizons multiples, dont les performances s'avèrent au moins équivalentes à celles d'un modèle pluie-débit GRP lorsque les jeux de données d'apprentissage utilisés portent sur plusieurs années hydrologiques consécutives. L'analyse des modèles MLP identifie bien le débit passé comme variable dominante. Ceci confirme que le MLP propose une interprétation des relations entre variables cohérente avec la physique de la relation pluie-débit . Comme perspectives, pour renforcer la capacité prédictive et capturer les dépendances temporelles plus complexes portant sur une fenêtre temporelle dépassant les 10jours, l'utilisation de modèles séquentiels tels que LSTM ou de modèles basés sur Transformer [8] pourrait être explorée avec une analyse explicative des modèles par XAI de type SHAP (Shapley Additive exPlanations). Par ailleurs, l'intégration de données pluviométriques issues de satellites permettrait de tester le pouvoir prédictif des modèles de deep learning dans des contextes de données moins favorables, en particulier dans les zones non couvertes par les réseaux pluviométriques au sol, et où les observations débitmétriques ne sont dispobnibles qu'à des pas de temps de 12 ou 24h.

Remerciements

Les auteurs remercient l'APIPA pour la fourniture des jeux de données nécessaires à ce travail. Ils remercient également le projet MADATLAS pour le financement de cette recherche, ainsi que Jean Donnée Rasolofoniaina pour son aide précieuse dans le développement du modèle.

Références

[1] C. Chen, Y. Liu, X. Sun, C. D. Cairano-Gilfedder, and S. Titmus. Automobile maintenance prediction using deep learning with gis data. *Procedia CIRP*, 81 :447–452, 2019.

[2] A. W. Kiwelekar, G. S. Mahamunkar, L. D. Netak, and V. B. Nikam. Deep learning techniques for geospatial data analysis. In G. A. Tsihrintzis and L. C. Jain, editors, *Machine Learning Paradigms : Advances in Deep Learning-based Technological Applications*, pages 63–81. Springer International Publishing, Cham, 2020.

[3] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 22(11) :6005–6022, 2018.

[4] M. Mohseni-Dargah, Z. Falahati, B. Dabirmanesh, P. Nasrollahi, and K. Khajeh. Machine learning in surface plasmon resonance for environmental monitoring. In *Artificial Intelligence and Data Science in Environmental Sensing*, pages 269–298. Academic Press, 2022.

[5] L. Oudin et al. Which potential evapotranspiration input for a lumped rainfall-runoff model ? part 2 : Towards a simple and efficient potential evapotranspiration model for rainfall-runoff modelling. *Journal of Hydrology*, 303(1–4) :290–306, 2005.

[6] A. R. Pereira and W. O. Pruitt. Adaptation of the thornthwaite scheme for estimating daily reference evapotranspiration. *Agricultural Water Management*, 66(3) :251–257, 2004.

[7] C. Perrin, C. Michel, and V. Andréassian. Modèles hydrologiques du génie rural (gr).

[8] Alain Josué Ratovondrahona, Hanitriniaina Marielle Rakotozanany, Thomas Mahatody, and Victor Manantsoa. Human like programming using spade bdi agents and the gpt-3-based transformer. *Proceedings of the AHFE International Conference on Human Interaction and Emerging Technologies*, 2023.

[9] F. Tilmant et al. Calage et application opérationnelle du modèle de prévision de crue grp - manuel d'utilisation (v2022.r3046), 2023. 93 pages, août 2023.

Human-centric annotation of multi-modal data: A framework perspective

Clément BELIVEAU[1, 2, 3, 4], Yann-Romain KECHABIA[4],
Cyril RAY [4], Maeve PETIT, Fabienne LAJONCHERE [3] , John PUENTES [1, 2]

[1]IMT Atlantique, Brest, FRANCE
[2]Lab-STICC CNRS UMR 6285, Brest, FRANCE
[3]Naval Group, CEMIS, Ollioules, FRANCE
[4]Ecole navale, IRENav, Lanveoc, FRANCE

clement.beliveau@naval-group.com

Abstract

Data annotation is a cognitively rich process shaped by perception, judgment, and variability. However, it is traditionally viewed as a mechanical and time-consuming task for humans, and is therefore often automated or outsourced. This paper considers annotation as a Human-Centric cognitive process and proposes a generic framework in which annotation supports humans in interpreting complex multimodal data and operational information. Through a taxonomy of human-in-the-loop semi-automatic pipelines, we show how AI-generated annotations can interact with humans. We apply this framework to a naval defense use case, specifically maritime surveillance activity. In this high-stakes environment, where operators must process continuous and heterogeneous information streams, annotation acts as a cognitive aid. It can ultimately become a decision-support tool that reduces cognitive load and preserves situational awareness, rather than remaining a simple data-labeling mechanism.

Keywords

Multi-modal data, Annotation, Data-Centric Artificial Intelligence, Human-Centric design, Human-In-The-Loop, Decision support.

1 Introduction

From finance to healthcare, from logistics to advertising, the modern world is awash in data, fueling the rapid advancement and integration of artificial intelligence (AI) across nearly every sector [25]. In this data saturated world, annotation is a foundational step in any machine learning pipeline, but from a cognitive ergonomics perspective it could also be seen as a form of cognitive externalization processes and serve as an efficient tool to help humans when dealing with complex and heterogeneous information [13, 22]. While often reduced to a mechanical labeling task, annotation is in reality shaped by human perception, judgment, and variability. In a Data-Centric AI paradigm [36], the quality of annotated data directly conditions its usage, making annotation an essential lever for system reliability. However, the human dimension of this process remains largely overlooked: who annotates, how, and under what cognitive conditions, are questions rarely addressed in the literature. Furthermore, as AI systems are increasingly deployed in operational settings, the boundary between annotation as a machine learning training step and annotation as a real-time decision-support tool becomes increasingly blurred. This raises a fundamental question: can annotation be reframed as a cognitive aid for humans confronted with complex, multi-modal, and ambiguous information, instead of merely as a resource for machine learning?

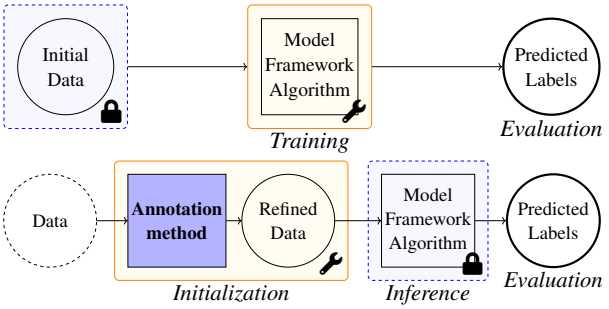


Figure 1: Model-Centric *(top)* and Data-centric *(bottom)* machine learning pipelines, 🔒 = frozen block – 🔧 = trainable block

Designed as a cognitive aid, data annotation requires to be implemented within a comprehensive, dynamic, and iterative improvement process, in which contextual nuances are swiftly captured. Regarding data annotation pipelines, currently two main paradigms prevail in AI: *Model-Centric* versus *Data-Centric* (Fig. 1). The Model-Centric one assumes that annotation is secondary, annotated data are available, data quality is taken for granted, and prioritizes algorithmic improvements through training techniques or architecture design. The Data-Centric paradigm shifts the focus to the quality, structure, and relevance of the data it-

self as the primary lever for enhancing model performance, to refine predictions by improving the model input [36]. Once a pipeline is trained and tuned with either paradigm, it is frozen and used for inference on new data (Fig. 2). This leads to the operator being a passive component of the pipeline. However, some critical sectors like defense, healthcare, or finance, cannot entirely remove humans from the decision process. For instance, in the defense domain, military operations increasingly rely on vast networks of sensors, technologies, and AI-based systems. Yet paradoxically, while defense operators are flooded with sensors data, they are often reluctant to share authority with autonomous or semi-autonomous decision-making systems. This tension raises a critical question: ***In high-stakes environments where human oversight is essential and data are overwhelming, what role can annotation play?***

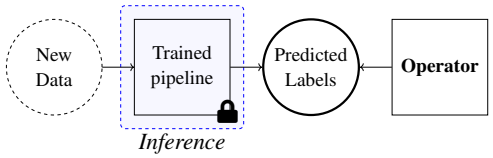


Figure 2: Inference phase where the operator uses the predicted labels

Hence, in this paper, we draw inspiration from the Data-Centric approach integrated with a Human-Centric paradigm, emphasizing the foundational yet often undervalued task of data **annotation** that precedes model training. Our work proposes a Human-Centered approach (Fig. 3) where annotation serves as a decision-support mechanism for humans, who remain central system components in the interpretation of complex multi-modal data and information. The proposed methodology is illustrated through a naval defense use-case in Section 4. More specifically it is applied to the activity of maritime surveillance deployed in semaphore stations and frigates. Rather than emphasizing system implementation, this paper adopts a conceptual perspective and investigates the interaction between human operators and an annotation framework, laying the groundwork for future instantiations and operational deployment.

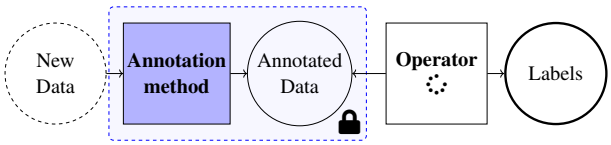


Figure 3: Position of the operator within a Human-Centered annotation paradigm

The article is organized as follows: Section 2 introduces a taxonomy of human-in-the-loop pipelines. The concept of data annotation, its use in the literature, and how it could be framed as a tool in a Human-Centered paradigm are presented in Section 3. The resulting concept is applied to the defense domain in Section 4, focusing on naval operators and instantiates the proposed multi-modal data annotation framework. Main resulting insights are discussed in section 5. Conclusions and perspectives are outlined in Section 6.

2 Human-in-the-loop for annotation

Annotation materializes the results from a succession of decisions and choices, aimed at summarizing insights on human perception at a given time, in order to train machine learning models [45]. Annotation can be fully manual, as with Amazon mechanical Turks [5] or crowd-sourcing [11, 27]. It can also be automatic, leveraging AI models trained on already annotated data [19, 37]. Finally, semi automatic ones combine human and AI to accomplish annotation [43]. Our framework moves away from a Model-Centric paradigm in which humans passively annotate datasets for machine learning. Instead, it promotes collaboration between humans and AI, where annotation becomes a tool to support human understanding. This approach aligns with hybrid intelligence research, which emphasizes the complementary roles of humans and AI in complex decision-making environments [10].
This section introduces the typology of Human-In-The-Loop (HITL) strategies for semi-automatic data annotation [43]. The goal is to understand how human and AI can interact, collaborate, and annotate together, and therefore, allow choosing the best HITL strategies for each use case and operator. Interactions between humans and several AI agents have not been previously studied in this particular context. We respectively describe as “Human” a given set of human operators and as “AI” a given set of models, within a simplified annotation process. We subdivide these strategies into five main types described below. Each one is illustrated by a schematic diagram. The legend of symbols is shown in Table 1.

Table 1: Legend and Symbols for annotation schemes

Legend	Description
H	Human
AI	Artificial Intelligence
A, A’	First and intermediate annotations
F	Final annotation

Symbols

◇	Annotation	(D)	Raw data
□	Annotation agent	⊕	Fusion component

2.1 Edge-Case

This pipeline is similar to a fully automatic one; the AI model processes the entire dataset, and the human acts as a judge. Instances where the model’s confidence falls below a predefined threshold are flagged for review. A human annotator then inspects and corrects these *edge-cases*, which often lie near the model’s decision boundary. This approach reduces cognitive load by filtering out trivial cases and concentrates human effort on the most uncertain or ambiguous

instances [27, 29]. However, AI model's errors in trivial examples might remain undetected [1], introducing biases in the annotated dataset (Fig. 4).

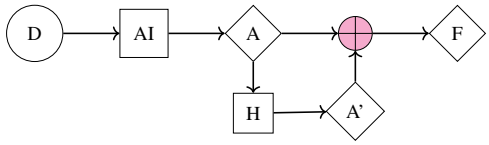


Figure 4: Block diagram of a semi-automatic edge-case pipeline (the human checks only automatic annotations that were flagged; automatic and human annotations are then fused)

2.2 Suggestion

The *suggestion* pipeline is conceptually closer to the manual approach; humans remain the primary annotators, while the AI can be summoned to suggest an annotation of a given instance. Operators can accept, modify, or reject these suggestions based on their judgment. This setup preserves human oversight and mitigates over-reliance on the model, leveraging AI assistance to ameliorate efficiency and consistency (Fig. 5).

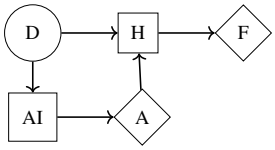


Figure 5: Block diagram of a semi-automatic suggestion pipeline (the final annotation is human - based on AI suggestions)

2.3 Iterative

In the *iterative refinement* pipeline, annotation is performed through alternating passes between the AI model and the human annotator. After each iteration, the model can be updated based on corrections provided by the human (active learning or reinforcement learning) [28] [23] [21]. This process continues until a predefined quality threshold is met [34]. The approach maintains expert involvement throughout the annotation, introducing the risk of human over-reliance on the model's suggestions, potentially limiting critical review (Fig. 6).

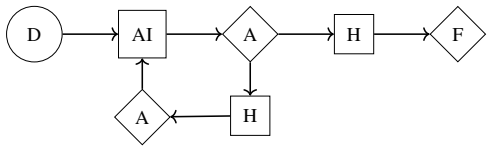


Figure 6: Block diagram of a semi-automatic iterative refinement pipeline (the final annotation is the result of a succession of human and AI annotations)

2.4 Few-shot

In this setting, a small part of the data is firstly annotated by humans, usually experts. These examples are then used to fine-tune or prompt in a *few-shot* setting to the model, which subsequently annotates the remaining data automatically. Additional human annotations may be requested if the model outputs fail to meet predefined quality thresholds. While this approach significantly reduces manual annotation efforts [15, 30], it remains susceptible to model severe inconsistencies and propagation of errors (Fig. 7).

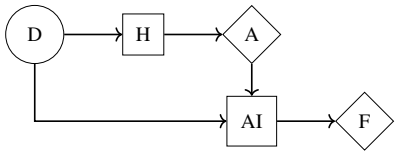


Figure 7: Block diagram of a semi-automatic few-shot pipeline (human annotations are used to automatically generate the final annotation)

2.5 Challenger

In the *challenger* pipeline, human and AI annotations are independently generated and compared to obtain the final annotation. Discrepancies trigger a dispute-resolution step, which may involve human review or additional decision logic. This design aims to mitigate human over-reliance on AI requiring the annotator to engage continually with each instance, while still benefiting from automated error detection and redundancy (Fig. 8).

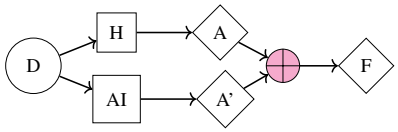


Figure 8: Block diagram of a semi-automatic challenger pipeline (human and AI annotate in parallel - the annotations are then fused)

3 Annotation properties and process

Annotation can be defined as a process through which humans add context, dimension, or information to existing data [19]. This section explores both its fundamental properties and the cognitive processes to derive annotations from raw data. On this basis, we introduce a generic framework in which annotation serves as a cognitive aid to support human interpretation and decision-making.

3.1 Fundamental properties

Annotation takes many forms depending of the context and goal, ranging from graphical to written representations. Initially formalized in textual contexts as "a note added to a text" [19], nowadays, annotation is everywhere in our dailylife:

- **Shopping mall**, a red dot "you are here" on a map or a colored arrow on a floor are graphical ones.
- **Television**, subtitles translating a movie or the crawling ticker during the news are written ones.
- **Restaurant**, reviews from previous customers or the number of stars they have assigned are both.

Varying from informal notes to formally structured labels, annotation is used to enrich data by adding contextual, structural, or semantic information, thus facilitating its understanding [16]. Annotation is intended for someone or something, a user or an AI model. Informal annotations can support human interpretation when exploring new data sources [35], whereas formal annotations typically consist of normalized labels that encode well-defined concepts. We identify four primary types of data that can be annotated: **textual, visual, numerical, and acoustic**, representing fundamental modalities to which any real-world data can be reduced.

3.1.1 Constructed from human perception

Annotation is based on how humans perceive input data. In this regard Gestalt principles and Cognitive Vision Theory (CVT) provide complementary perspectives on how humans perceive visual inputs. Gestalt principles explain how the human visual system organizes elements into coherent structures based on proximity, similarity, continuity, closure, and common fate [40]. CVT formalizes the layer of interpretation through which humans process and understand information [20] [44]. In spite of being initially formulated for visual stimuli, such concepts could transfer across various types of data [24].
Indeed, during annotation, humans establish a semantic interpretation of a situation that involves three levels [8]:

- **Low level**, corresponds to the perception of raw sensory features.
- **Mid level**, involves structural analysis and logical inference.
- **High level**, integrates abstract reasoning and prior knowledge.

Interpretation levels influence the cognitive effort during annotation. Specifically, high-level reasoning is typically an active, attention-demanding process [32], whereas low- and mid-level perception often occur unconsciously, governed by innate perceptual mechanisms [4]. In operational environments such as maritime surveillance, operators must rapidly shift between these levels to integrate heterogeneous signals and maintain a dynamic representation of the situation, reflecting situation awareness mechanisms that support perception, comprehension, and projection in complex, evolving environments [13, 42]. Understanding these perceptual mechanisms provides a foundation for designing annotation tools and workflows that optimize cognitive load, allowing operators to focus on high-value tasks.

3.1.2 Abstraction levels

According to examined works, we propose a classification of annotations into four hierarchical layers, each corresponding to a different level of abstraction:

1. **Surface layer**, refers to annotations based on raw, low-level features of the data. It is rarely treated as a distinct annotation layer in the literature [21].
2. **Structural layer**, captures the internal organization and structure of the data. Through global features, it identifies units and their spatial or temporal arrangement, often without interpreting their semantic content [26] [39].
3. **Semantic layer**, encodes standardized and widely understood meanings using shared conceptual frameworks. These annotations, based on common concepts, aim for consistency across systems and annotators [1, 20, 23].
4. **Interpretative layer**, represents subjective, high-level data interpretations, involving emotional, social, or contextual judgment. This annotation layer is somewhat recent, and unfolded initially from sentiment and social science fields, where human perception and affective response play a central role [45].

3.2 Annotation ~~as~~, a human process

The role humans play in the annotation process is an essential topic rarely discussed, which could help broaden AI capabilities by grasping how human intelligence works in this particular case [6].

3.2.1 Human strategies

The cognitive process deployed by humans to realize manual annotations is called a strategy. It represents the user's plan of action, formed by chains of tactics and moves. Annotation tactics are the steps, actions, and choices deployed by humans to move forward in order to carry out an annotation. Tactics are formed by annotation moves that are basic actions of thought performed by the annotator (zoom-in, zoom-out, compare, measure, define, etc.) [8]. An annotation strategy might include, for instance, a fine-tuning tactic, where the operator refines or adjusts an annotation to achieve greater accuracy afterward [2].
Considering these strategies when designing annotation tools might be useful to encourage or avoid certain tactics. Although, even if strategies vary between annotators, trends emerge. For instance, individuals are likely to prefer starting with less accurate tactics involving perception, rather than cognitively demanding ones. Also, individuals applying less accurate methods tend to be willing to continue refining their work [8, 34].

3.2.2 Human biases

Humans are inherently sensitive to cognitive, perceptual, and emotional biases, which can affect annotation quality and may be exacerbated under conditions of cognitive

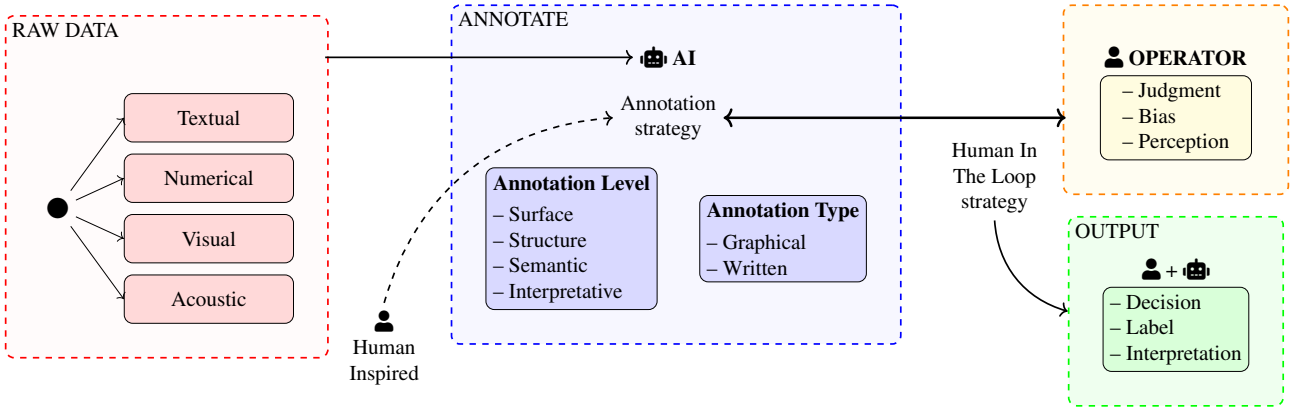


Figure 9: Proposed generic Human-Centered annotation framework for multi-modal data

overload [38]. Some biases, such as distrust or overconfidence, arise from the nature of the task itself, while others emerge from the structure and sequencing of the annotation process. Similar to surveys, annotation campaigns involve repeated human judgments under controlled conditions; in such settings, the order in which tasks are presented can induce contrast and assimilation effects [2].

The data being annotated can also generate variability in perception among annotators, potentially leading to confusion between certain labels [6]. This challenge is amplified by the heterogeneous and multi-modal nature of datasets, which often combine simple instances with complex ones of different modalities. Consequently, the difficulty of annotation can fluctuate dramatically, ranging from trivial to expert-level depending on the label schema. To mitigate such issues, adaptive labeling interfaces that dynamically adjust to the annotator's perceived task difficulty have been proposed [7].

How annotators perceive information is shaped by their prior experience and domain knowledge [11]. Individual's bias, expertise, or fatigue can influence the labels they assign [2, 9], with potential consequences for downstream model performance. Annotation is therefore not a purely technical task but a cognitively rich process shaped by perception, judgment, and variability.

By integrating all these components, we propose a generic Human-Centered annotation framework (Fig. 9), that can be tailored to specific use cases depending on the available data and the desired human-in-the-loop (HITL) configuration, as illustrated in Section 4 for a defense application. Especially suited for multi-modal data, this flexible framework searches to strengthen human decision-making, interpretation, and labeling capabilities. Based on contextual information an AI agent generates an annotation strategy designed to reflect human reasoning processes. The resulting annotation can thus become a cognitive artifact supporting operators' sense-making processes in complex environments, helping them externalize intermediate interpretations and stabilize evolving hypotheses during decision making [10, 22].

4 Application to a defense use case

Leveraging annotation not merely as a technical aid but as a cognitive support for humans remains largely unexplored both in hybrid-intelligence literature and in defense contexts where it may play a key role in supporting human decision-making in complex operational environments. This insight was reinforced by field observations conducted to model maritime surveillance activities. The pipeline of the current activity, presented in Fig. 10, is the result of a series of in situ observations and semi-structured interviews realized with a frigate crew and ten operators across four semaphores.

4.1 Multiplication of signals

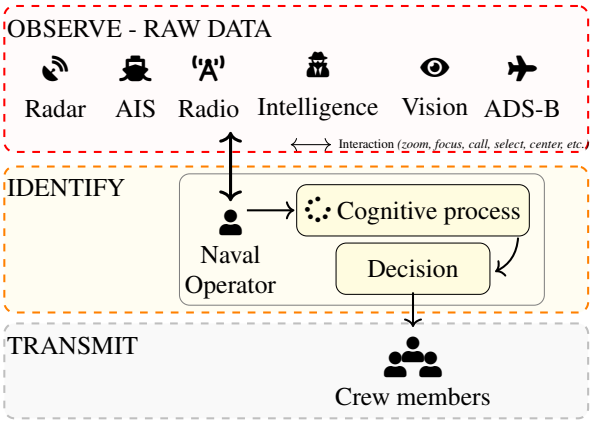


Figure 10: Current naval surveillance pipeline - the naval operator deploys cognitive processes to identify objects from all the collected information (*AIS - Automatic Identification System; ADS-B - Automatic Dependent Surveillance-Broadcast*)

A key element from field observations is that defense operators face a continuous flow of data from multiple sources; often incomplete, sometimes noisy, and frequently asynchronous [12]. In recent decades, the proliferation of sensors has led to an increase of data sources, resulting in significant cognitive burden for operators [41]. As presented in Fig 10, a naval operator may have to combine abruptly

radar detection, perform visual confirmations, monitor and engage in radio communication, coordinate with other services, and annotate observations, for each detected track in an observation-identification-transmission loop.

4.2 Ambiguity of signals

Naval operators constantly *collect* information and *correlate* fragmented signals. When data are incomplete, operators must *anticipate* developments and *complete* gaps using their expertise and knowledge. Furthermore, they are required to *prioritize* relevant tracks and signals, focusing on the most critical ones, to *make decisions* in real time, often under significant operational pressure.

In this context, information can be partial or ambiguous, which ergonomics literature refers to as situations of structural uncertainty [18]. Unlike purely procedural tasks, where each action logically follows from a clear signal [33], field observations have shown that maritime surveillance requires naval operators to interpret incomplete, unconfirmed, or contradictory signals, while remaining responsible for the potential consequences of their decisions, thus preventing the use of a rigid autonomous system.

Therefore, behind the seemingly simple task of identification, lie countless invisible actions that add up and might lead to a cognitive overload: **collect, correlate, anticipate, complete, prioritize, and make a decision**.

4.3 Human-Centric annotation

Consequently the Human-Centric annotation framework seems particularly suited to this defense context. The goal of annotation is not to replace human intervention but to enhance operators' work capacity, senses, and perception, with a purpose beyond merely recording events in a database. It must be a projection of the operator's cognitive processes, involving the selective identification of salient cues, the interpretation of signals based on task-related knowledge, and the intentional omission of elements deemed routine or unremarkable.

The Human-Centric annotation framework previously defined in Fig. 9, will be implemented alongside existing observation tasks described in Fig. 10, resulting in the dynamic collaboration presented in Fig. 11. In this setting, the naval operator can actively choose to use raw data, annotated ones, or a mixture of both to construct a decision. Consequently, the operator is active, and is not constrained by a decision from an automatic system. According to field observations, naval operators tend to reject imposed predictions from automatic systems they cannot control. Our Human-Centric annotation framework lets the operator take the final decision and stay in control. The annotation strategy is automatically constructed based on operationnal objective and context. Then each form of data can be annotated, radio communications would be classified and annotated as *acoustic* data, whereas AIS transmissions would be annotated as *numerical* data. Table 2 presents an analysis of possible annotations and demonstrates the application of the proposed annotation framework to the naval defense use case.

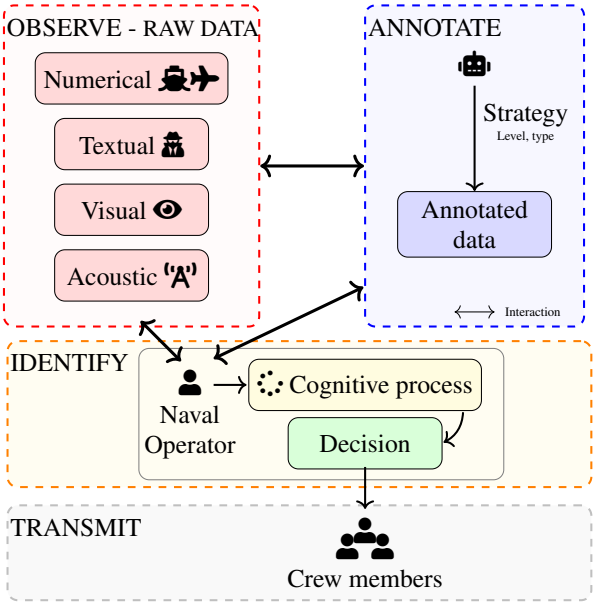


Figure 11: Proposed naval surveillance pipeline enhanced with an annotation module for each data type (textual, numerical, visual, and acoustic); the operator can use either annotations or collected information to identify objects

5 Implications and Future Directions

This section summarizes our reflections on enhancing human performance within a Human-Centric framework. An automatic system disconnected from a human operator can lead to overconfidence biases, where humans can reduce their involvement in an active analysis [31]. This phenomenon is amplified when automated interfaces leave little room for human interpretation or require scarce attention outside of critical alerts. A HITL strategy with an active engagement loop is crucial to mitigate human biases, either overconfidence or distrust, and avoid autopilot effect [17].

Additionally, gradual removal of responsibility leads to a significant decline in situational awareness, i.e., the ability to perceive, understand, and anticipate the dynamics of a changing environment [14]. This loss is precisely what needs to be avoided in the context of defense. An annotation framework where the operator is not directly providing an answer, but receives enhanced information, can be seen as a way to force human cognition [4], reducing overconfidence and distrust.

Based on observations of how maritime semaphore station and frigate operators annotate data, the *Suggestion*, *Iterative*, and *Challenger* HITL pipelines appear particularly well adapted as they preserve full human involvement.

5.1 Reframing Annotation as a Human-Centric Task

Throughout this study, the importance of human involvement in the annotation process has been repeatedly corroborated. Annotation is an intrinsically Human-Centric task, even though it is often perceived as a mechanical or aux-

Table 2: Analysis of possible annotations across modalities for naval surveillance data

Layer	**Task Name**	***[Annotation type]* - Task Description**
		Textual (Documentation - Intelligence)
Surface	Highlighting	*[Graphical]* Visually emphasize specific spans of a text.
Surface	Part-of-Speech Tagging	*[Written]* Assign part-of-speech tag (e.g., NOUN, VERB) to each token.
Surface	Lemmatization	*[Written]* Reduce inflected forms to their base form (lemmas).
Structural	Paragraph Extraction	*[Graphical]* Identify and extract text segments relevant to a query.
Semantic	Named Entity Recognition	*[Written]* Identify and label spans corresponding to entities.
Semantic	Topic Tagging	*[Written]* Assign one or more topic labels indicating thematic content.
Interpretative	Sentiment Analysis	*[Written]* Classify sentiment expressed in a sentence or document.
Interpretative	Relation Extraction	*[Written]* Identify and label semantic relationships between entities.
Interpretative	Summarization	*[Written]* Produce a condensed representation while preserving essential information.
Interpretative	Schema Generation	*[Graphical]* Transform structured or semi-structured text into an ordered representation (e.g., table, concept map).
		Visual (Camera)
Structural	Bounding Box	*[Graphical]* Delimit regions of interest in an image using rectangles.
Structural	Segmentation Mask	*[Graphical]* Assign a class label to each pixel for fine-grained object delineation.
Structural	Object Tracking	*[Graphical]* (Video) Assign identifiers to objects across frames to capture trajectories.
Semantic	Object Description	*[Graphical]* Provide structured metadata about objects (e.g., class, location, attributes).
Semantic	Classification	*[Written]* Assign labels to images or localize and label individual objects.
Semantic	Scene Detection	*[Written]* (Video) Identify shot boundaries or scene transitions.
Interpretative	Scene Graph Generation	*[Graphical]* Identify objects and their pairwise relationships (e.g., “man–riding–moto”).
Interpretative	Image Captioning	*[Written]* Automatically generate descriptive captions for images.
Interpretative	Intent Captioning	*[Written]* (Video) Generate descriptions explaining goals or intentions behind actions.
		Audio (Radio)
Surface	Subtitle	*[Written]* Transcribe spoken content into aligned text segments.
Semantic	Audio Classification	*[Written]* Assign semantic categories (e.g., genre, environment, speaker identity).
Interpretative	Emotion Classification	*[Written]* Identify and label the emotional state expressed in audio.
		Numeric (AIS – ADS-B – Radar)
Surface	Highlighting	*[Graphical]* Emphasize specific numeric ranges or values (e.g., outliers).
Structural	Region of Interest Selection	*[Graphical]* Select specific segments in a time series or numerical matrix.
Structural	Data Visualisation	*[Graphical]* Project data into a space emphasizing structures (e.g., clusters, outliers).
Interpretative	Trend Classification	*[Written]* Label time series segments according to their patterns (e.g., increasing, cyclic).
Interpretative	Anomaly Classification	*[Written]* Identify and categorize abnormal data points or sequences.

iliary step. In high-stakes domains such as defense, where sensor data are abundant and real-time decisions are crucial, manual annotation alone becomes impractical. Operators often lack the time to explicitly annotate; instead, they do what may be interpreted as implicit annotation.

We claim that designing annotation systems, specifically to support human operators, can enhance decision-making capacities and reduce cognitive overload, which implies shifting the paradigm from data labeling to Human-Centered interaction.

5.2 Towards Annotation Fusion: Managing Variability and Ambiguity

A challenge that was intentionally omitted of the core scope of this paper is the question of inter-annotator variability, and more broadly, multi-agent annotation. In complex, ambiguous, or cognitively demanding situations (as in defense scenarios), annotators production may diverge significantly, even working under shared guidelines.

This subject emerged as fundamental and should be explic-

itly modeled and incorporated. For instance, annotations could be contextualized based on annotators' profiles, or different opinions could be weighted according to expertise or cognitive style. On the AI side, this also raises the issue of AI-AI variability, where multiple AI systems produce different annotations.
A use case like this one, leads naturally to the concept of annotation fusion, where diverse annotations, from humans, AI, or both, are merged to create more robust and explainable datasets. Combining multiple annotations, potentially informed by annotators' reasoning process, offers a promising direction for mitigating ambiguity, while retaining the richness of multiple viewpoints.

5.3 Annotation for decision support

Finally, multiple-criteria decision-making (MCDM) methods are well-established and powerful decision-support tools for handling numerical data. In defense applications, such systems are already used to help operators ranking and interpreting sensor outputs, as well as in selecting appropriate strategies and behaviors [3]. However, operators struggle to handle visual or textual information. Annotation could serve therefore as a complementary mechanism to enrich MCDM approaches. Conversely, MCDM techniques could also be leveraged to weight and prioritize which annotations are the most relevant.

5.4 Implementation of a Human-Centric Annotation Framework

As previously discussed, annotation in naval environments is frequently conducted under time constraints, with operational realities compelling operators to adapt beyond prescribed procedures. Furthermore, the importance of contextual information may fluctuate according to geographic and meteorological conditions. As a result, rigid systems such as rule-based algorithms or ontology-driven approaches may lack the flexibility required for such dynamic settings. A Human-Centered annotation framework should accordingly be grounded in a system that facilitates seamless interaction between human and artificial agents, supporting adaptive collaboration with the data. In this regard, multi-agent systems appear particularly promising.

6 Conclusion

Surveillance within maritime semaphore station and frigate is complex tasks involving continuous observation, identification, and decision. Currently, operators must gather information from various sources, to be correlated before making decisions. In this process annotation is implicit and lacks automatization or normalization. There are at least three possible semi-automatic annotation approaches that could improve operators' decision capacity. These approaches require to design intelligent annotation systems, beyond prescribed theoretical procedures, to meet real operational constrains and needs. Annotation resulting of human-AI collaboration, could facilitate normalization and consistency within the resulting data, while being a cognitive aid for humans operating in high-stakes environments. Data enriched semantically with meaning and context, could lead to improved machine learning models, enabling efficient identification of normal and abnormal behaviors. Future work will investigate methods for merging human insights, AI suggestions, and task-specific heuristics into a cohesive annotation pipeline.

7 Acknowledgment

This work is funded by Agence Nationale de la Recherche et de la Technologie, CIFRE n°2024/1316, within the research program of Chaire NAIADE. Authors are very grateful to maritime semaphore station and frigate operators, who answered our questions about their sea and air surveillance missions.

References

[1] Shiva Agrawal, Savankumar Bhanderi, and Gordon Elger. Semi-automatic annotation of 3d radar and camera for smart infrastructure-based perception. *IEEE Access*, 12:34325–34341, 2024.

[2] Jacob Beck, Stephanie Eckman, Rob Chew, and Frauke Kreuter. Improving labeling through social science insights: results and research agenda. In *International Conference on Human-Computer Interaction*, pages 245–261. Springer, 2022.

[3] Dragan Bojanic, Mitar Kovač, Marina Bojanic, and Vladimir Ristic. Multi-criteria decision-making in a defensive operation of the guided anti-tank missile battery: An example of the hybrid model fuzzy ahp-mabac. *Decision Making: Applications in Management and Engineering*, 1(1):51–66, 2018.

[4] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. To trust or to think: cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making. *Proceedings of the ACM on Human-computer Interaction*, 5(CSCW1):1–21, 2021.

[5] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality data? *Perspectives on Psychological Science*, pages 3–5, 2016.

[6] Daniel N Cassenti, Vladislav D Veksler, and Frank E Ritter. Editor's review and introduction: Cognition-inspired artificial intelligence, 2022.

[7] Chia-Ming Chang, Yi He, Xusheng Du, Xi Yang, and Haoran Xie. Dynamic labeling: A control system for labeling styles in image annotation tasks. In *International Conference on Human-Computer Interaction*, pages 99–118. Springer, 2024.

[8] Jing Chen, Dan Wang, Iris Xie, and Quan Lu. Image annotation tactics: transitions, strategies and efficiency. *Information Processing & Management*, 54(6):985–1001, 2018.

[9] Jeffrey Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. In *Ethics of data and analytics*, pages 296–299. Auerbach Publications, 2022.

[10] Dominik Dellermann, Philipp Ebel, Matthias Söllner, and Jan Marco Leimeister. Hybrid intelligence. *Business & information systems engineering*, 61(5):637–643, 2019.

[11] Anca Dumitrache. Truth in disagreement: Crowdsourcing labeled data for natural language processing. 2019.

[12] Mica R Endsley. Situation awareness misconceptions and misunderstandings. *Journal of cognitive Engineering and Decision making*, 9(1):4–32, 2015.

[13] Mica R Endsley. From here to autonomy: lessons learned from human–automation research. *Human factors*, 59(1):5–27, 2017.

[14] Mica R Endsley. Toward a theory of situation awareness in dynamic systems. In *Situational awareness*, pages 9–42. Routledge, 2017.

[15] Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, et al. Llms accelerate annotation for medical information extraction. In *machine learning for health (ML4H)*, pages 82–100. PMLR, 2023.

[16] Jose Nuno Gomes-Pereira, Vincent Auger, Kolja Beisiegel, Robert Benjamin, Melanie Bergmann, David Bowden, Pal Buhl-Mortensen, Fabio C De Leo, Gisela Dionísio, Jennifer M Durden, et al. Current and future trends in marine image annotation software. *Progress in Oceanography*, 149:106–120, 2016.

[17] Jean-Michel Hoc. Towards ecological validity of research in cognitive ergonomics. *Theoretical issues in ergonomics science*, 2(3):278–288, 2001.

[18] Erik Hollnagel. *The ETTO principle: efficiency-thoroughness trade-off: why things that go right sometimes go wrong*. CRC press, 2017.

[19] Remi Kalir and Antero Garcia. Annotation guidelines. *Annotation*, 2019.

[20] Yusuke Kamoi, Yosuke Furukawa, Tatsuya Sato, Yuya Kiwada, and Tomohiro Takagi. Automatic image annotation based on visual cognitive theory. In *NAFIPS 2007-2007 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 239–244. IEEE, 2007.

[21] Hannes Kath, Thiago S Gouvêa, and Daniel Sonntag. A human-in-the-loop tool for annotating passive acoustic monitoring datasets. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 341–345. Springer, 2024.

[22] Gary Klein, Brian Moon, and Robert R Hoffman. Making sense of sensemaking 2: A macrocognitive model. *IEEE Intelligent systems*, 21(5):88–92, 2006.

[23] Jan-Christoph Klie, Richard Eckart De Castilho, and Iryna Gurevych. From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 6982–6993, 2020.

[24] Iuliia Kotseruba and John K Tsotsos. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1):17–94, 2020.

[25] Sushant Kumar, Sumit Datta, Vishakha Singh, Sanjay Kumar Singh, and Ritesh Sharma. Opportunities and challenges in data-centric ai. *IEEE Access*, 12:33173–33189, 2024.

[26] Sang-Heon Lee, Hae-Gwang Park, Ki-Hoon Kwon, Byeong-Hak Kim, Min Young Kim, and Seung-Hyun Jeong. Accurate ship detection using electro-optical image-based satellite on enhanced feature and land awareness. *Sensors*, 22(23):9491, 2022.

[27] Shixia Liu, Changjian Chen, Yafeng Lu, Fangxin Ouyang, and Bin Wang. An interactive method to improve crowdsourced annotations. *IEEE transactions on visualization and computer graphics*, 25(1):235–245, 2018.

[28] Zimo Liu, Jingya Wang, Shaogang Gong, Huchuan Lu, and Dacheng Tao. Deep reinforcement active learning for human-in-the-loop person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6122–6131, 2019.

[29] Ayodele Marvellous, Bamidele Matthew, Mauro Pezzè, Silvia Abrahão, and Birgit Penzenstadler. Human-in-the-loop ai engineering: Enhancing collaboration between developers and end users. 2025.

[30] Nick Pangakis and Sam Wolken. Keeping humans in the loop: human-centered automated annotation with generative ai. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 19, pages 1471–1492, 2025.

[31] Raja Parasuraman and Victor Riley. Humans and automation: Use, misuse, disuse, abuse. *Human factors*, 39(2):230–253, 1997.

[32] Jan L Plass, Roxana Moreno, and Roland Brünken. *Cognitive load theory*. Cambridge university press, 2010.

[33] Jens Rasmussen. Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE transactions on systems, man, and cybernetics*, (3):257–266, 2012.

[34] Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2121–2131, 2015.

[35] Mary B Ruvane. Defining annotations: a visual (re)-interpretation. *Proceedings of the American Society for Information Science and Technology*, 43(1):1–5, 2006.

[36] Lars Schmarje, Vasco Grossmann, Claudius Zelenka, Sabine Dippel, Rainer Kiko, Mariusz Oszust, Matti Pastell, Jenny Stracke, Anna Valros, Nina Volkmann, et al. Is one annotation enough?-a data-centric image classification benchmark for noisy and ambiguous label estimation. *Advances in Neural Information Processing Systems*, 35:33215–33232, 2022.

[37] Anastasios Stamoulakatos, Javier Cardona, Chris McCaig, David Murray, Hein Filius, Robert Atkinson, Xavier Bellekens, Craig Michie, Ivan Andonovic, Pavlos Lazaridis, et al. Automatic annotation of subsea pipelines using deep learning. *Sensors*, 20(3):674, 2020.

[38] Teodor Stoev, Kristina Yordanova, and Emma L Tonkin. Experiencing annotation: Emotion, motivation and bias in annotation tasks. In *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 534–539. IEEE, 2023.

[39] Jinhui Tang, Qiang Chen, Meng Wang, Shuicheng Yan, Tat-Seng Chua, and Ramesh Jain. Towards optimizing human labeling for interactive image tagging. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(4):1–18, 2013.

[40] Dejan Todorovic. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008.

[41] Christopher D Wickens. Multiple resources and mental workload. *Human factors*, 50(3):449–455, 2008.

[42] Christopher D Wickens, Sallie E Gordon, Yili Liu, and J Lee. *An introduction to human factors engineering*, volume 2. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[43] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381, 2022.

[44] Dengsheng Zhang, Md Monirul Islam, and Guojun Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, 2012.

[45] Yixin Zhu, Tao Gao, Lifeng Fan, Siyuan Huang, Mark Edmonds, Hangxin Liu, Feng Gao, Chi Zhang, Siyuan Qi, Ying Nian Wu, et al. Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense. *Engineering*, 6(3):310–345, 2020.

Session 4 : Perception, Vision & Capteurs

Self-Supervised Alignment of RGB-Infrared Representations for Embedded Perception

Abdelmalek Belghomari[1,2], Clara Barbanson[1], Frederic Jurie[2], Alexis Lechervy[2]

[1] Safran Electronics and Defense
[2] Université de Caen Normandie, GREYC

8 mars 2026

Résumé

La fusion d'images RVB-IR exige un alignement précis et des bases de données annotées. Pour s'en affranchir, nous proposons une approche auto-supervisée basée sur l'architecture JEPA. En prédisant les caractéristiques latentes de l'IR à partir d'un contexte RVB masqué, notre modèle projette les deux modalités dans un espace sémantique partagé. Les résultats préliminaires montrent que cet alignement constitue une base solide pour la perception embarquée, sans aucune intervention humaine.

Mots-clés

Apprentissage auto-supervisé, JEPA, Alignement de domaines, Multimodalité, Perception embarquée.

Abstract

RGB and IR image fusion requires precise alignment and annotated datasets. To eliminate this need for manual labeling, we propose a self-supervised approach using the Joint-Embedding Predictive Architecture (JEPA). By predicting IR latent features from masked RGB context, our model projects both modalities into a shared semantic space. Preliminary results show this alignment provides a solid foundation for embedded perception without any human intervention.

Keywords

Self-Supervised Learning, JEPA, Domain Alignment, Multimodality, Embedded Perception.

1 Introduction

Aligning visible (RGB) and infrared (IR) domains is a prerequisite for robust multispectral perception in defense and embedded systems. While RGB sensors excel in providing high-resolution textures and contextual details during the day, IR sensors capture thermal signatures that remain highly reliable in degraded visual environments, such as nighttime, dense fog, or camouflage scenarios. However, designing an effective fusion strategy that respects the strict power and latency constraints of embedded hardware remains a significant challenge.

Existing fusion strategies are typically categorized by their integration stage and architectural backbone. Early and middle fusion methods concatenate inputs or intermediate representations, but require extensive paired data for precise alignment. Late fusion processes modalities independently ; while effective on popular benchmarks, the dual-backbone overhead severely limits embedded deployment. Regarding backbones, Transformers (e.g., CAFF-DINO [4]) utilize cross-attention for state-of-the-art accuracy, though their quadratic complexity hinders real-time use. Alternatively, State-Space Models (SSMs) like Mamba [2] and its vision variants [6] offer linear complexity, but their efficacy for multimodal fusion remains an open empirical question.

Consequently, several key limitations emerge from these dominant approaches in the embedded context : *(1) Annotation dependency :* Most early and middle fusion methods require dense labeled RGB-IR pairs, which are costly and difficult to acquire in operational defense conditions. *(2) Computational cost :* Cross-attention and dual-backbone architectures are computationally expensive, making them challenging to run at real-time frame rates on embedded SoCs or FPGAs. *(3) Scene generalization :* Supervised models trained on specific benchmarks like LLVIP [5] often fail to generalize to new operational environments without extensive retraining.

To address these challenges, we argue that self-supervised learning (SSL) offers a practical path toward efficient cross-modal alignment. In this paper, we propose a research direction based on the Joint-Embedding Predictive Architecture (JEPA) [1] as a zero-annotation framework to learn a shared latent space, providing a lightweight and scene-agnostic foundation for real-time embedded perception.

2 Proposed Methodology

The primary motivation for adopting an SSL framework lies in the critical scarcity of annotated multimodal data. By leveraging unlabelled raw sensor data, we shift the bottleneck from expert human annotation to the simple acquisition of raw data. To this end, we propose a self-supervised pretraining strategy based on JEPA [1], adapted specifically for cross-modal alignment.

2.1 Cross-Modal JEPA Formulation

JEPA learns representations by predicting the latent representation of a target view from a context view, *entirely in*

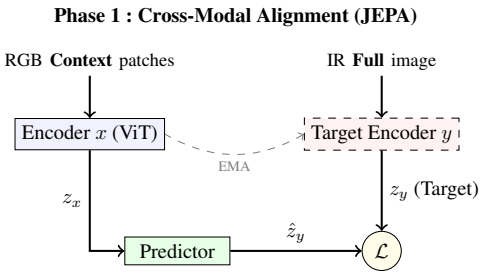


FIGURE 1 – Cross-modal Latent Alignment via JEPA

feature space. This is a critical distinction from masked autoencoders (MAE) [3] : predicting in latent space forces the model to capture high-level semantic structures rather than low-level pixel statistics, which is ultimately better suited for downstream detection tasks.

We adapt JEPA to the cross-modal RGB-IR setting through three main components : *The Context Encoder* processes patches of the RGB image. Crucially, a large portion of these RGB tokens are masked. The encoder's objective is to learn a compact visible-spectrum representation from the remaining visible context. *The Target Encoder* receives the corresponding, strictly aligned IR image. The weights of the target encoder are an exponential moving average (EMA) of the context encoder. This prevents representation collapse and provides stable cross-modal latent targets. *The Predictor* acts as the alignment mechanism. Taking the encoded RGB context and the positional embeddings of the masked tokens, it is trained to predict the exact latent features of the corresponding masked regions in the IR domain. The entire framework is optimized by minimizing the L_2 distance between the predictor's output and the target encoder's representation.

2.2 Methodological Advantages for Embedded Systems

Our proposed formulation directly addresses embedded limitations : *(1) Zero-annotation fusion :* It relies solely on co-registered RGB-IR pairs, completely eliminating the need for bounding boxes or semantic maps during pretraining. *(2) Lightweight design :* By using a lightweight Vision Transformer (ViT-Tiny) with approximately 5.7M parameters, the backbone remains strictly compatible with embedded hardware targets like low-power edge devices. *(3) Domain Agnosticism :* The SSL objective relies on structural cross-modal correlation rather than class labels, theoretically allowing the network to be trained on diverse unlabelled operational datasets (desert, jungle, urban) without manual retagging.

3 Preliminary Proof of Concept & Future Work

To validate our proposed methodology, we trained the cross-modal JEPA on the LLVIP dataset [5], which contains over 15,000 aligned image pairs recorded in challenging low-light conditions. We utilized a ViT-Tiny backbone with a patch size of 16 and a resolution of 224×224, optimizing the architecture over 450 epochs to ensure full convergence of the self-supervised representations.

To evaluate the learned representation, we froze the pretrained encoder and attached a DETR (DEtection TRansformer) head. Because DETR relies heavily on rich, global feature sets for its bipartite matching algorithm, it serves as an excellent probe for feature quality. At epoch 450, our model reached a stable localization plateau ($L_1 \approx 0.051$, $IoU \approx 67.5\%$), proving that the cross-modal self-supervised features capture the spatial geometry necessary for object detection. However, a classification error of 60% indicates that capturing fine-grained semantic distinctions remains an ongoing challenge.

Open Research Directions : Building on these preliminary insights, several methodological paths remain open : *(1) Information Saturation :* The model likely saturates the semantic information available in static LLVIP pairs. We plan to implement aggressive cross-modal data augmentations to synthetically expand the training distribution. *(2) SSM Alternatives :* Investigating whether State-Space Models (e.g., VMamba) can replace ViT within the JEPA framework to further reduce inference complexity for extreme embedded targets. *(3) Transferability :* Evaluating whether this scene-agnostic pretraining inherently transfers better to unobserved operational environments compared to standard supervised baselines.

4 Conclusion

We proposed a self-supervised methodology using JEPA to establish a shared RGB-IR semantic space without manual labels. By predicting IR latent features from masked RGB contexts, the framework intrinsically aligns both modalities. Preliminary validation confirms stable geometric feature extraction using a lightweight ViT-Tiny backbone. Future work will refine semantic classification via cross-modal augmentations and explore sub-quadratic SSM backbones to satisfy extreme embedded perception constraints.

Références

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

[2] Albert Gu and Tri Dao. Mamba : Linear-time sequence modeling with selective state spaces. In *ICLR*, 2024.

[3] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[4] Kevin Helvig, Baptiste Abeloos, and Pauline Trouvé-Peloux. CAFF-DINO : Multi-spectral object detection transformers with cross-attention features fusion. In *CVPR*, 2024.

[5] Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. LLVIP : A visible-infrared paired dataset for low-light vision. In *ICCV Workshops*, 2021.

[6] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunnan Liu. VMamba : Visual state space model. In *NeurIPS*, 2024.

CAESAR++: Uncertainty-Driven Contextual Reasoning for Trustworthy and Explainable Road Object Detection

Anh-Thu Mai[1,2], Marina Nicolas[1], Patricia Ladret[2], Alice Caplier[2]

[1] STMicroelectronics, Grenoble, France
[2] Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, France

anh-thu.mai@st.com

Résumé

Cet article présente CAESAR++, une approche pour la détection d'objets routiers qui combine la prédiction conforme, un raisonnement contextuel adaptatif et des cartes de saillance à double couleur. CAESAR++ calibre d'abord les incertitudes de classification et de localisation au moyen d'une procédure conforme en deux étapes, puis agrandit dynamiquement la fenêtre de contexte autour de chaque détection en fonction de son niveau d'incertitude, et produit enfin des explications au niveau de l'objet qui distinguent les indices sensoriels locaux des indices contextuels. Les expériences montrent des gains constants en précision, en calibration de l'incertitude et en stabilité des explications, sans réentraîner les modèles de base.

Mots-clés

Intelligence artificielle explicable, détection d'objets routiers, prédiction conforme, raisonnement contextuel.

Abstract

This paper introduces CAESAR++, a framework for road object detection that combines conformal prediction, adaptive contextual reasoning, and dual-color saliency maps. CAESAR++ first calibrates classification and localization uncertainty using a two-step conformal procedure, then enlarges the context window around each detection in proportion to its uncertainty, and finally produces object-wise explanations that disentangle bottom-up sensory evidence from top-down contextual cues. Experiments indicate consistent improvements in detection accuracy, uncertainty calibration, and explanation stability without retraining the base models.

Keywords

Explainable artificial intelligence, road object detection, conformal prediction, contextual reasoning.

1 Introduction

Reliable perception is a central requirement for autonomous vehicles and advanced driver-assistance systems. In real-world urban traffic, detectors must recognize and localize a variety of objects despite occlusions, adverse weather, cluttered backgrounds and small apparent sizes. Recent deep learning detectors have made considerable progress [1], yet they often remain overconfident and brittle under challenging conditions in real-world scenarios [2].

Two aspects are particularly critical for trustworthy perception. First, the system should provide well-calibrated uncertainty on both class labels and bounding boxes so that downstream modules can take risk-aware actions. Second, the system should expose human-understandable explanations that reveal which visual cues drive each detection, in order to support debugging and user trust [3].

Existing approaches usually address these aspects separately. Uncertainty estimation methods, including Bayesian approximations and ensembles [4, 5], improve calibration but rarely show how to act on the estimated uncertainty. Conformal prediction provides distribution-free coverage guarantees [6, 7], and recent work has adapted it to detection [8, 9]. However, these methods are often limited to reporting coverage statistics, without using uncertainty to guide refinement or explanation.

Conversely, explanation techniques for object detectors [10] usually rely on local visual features and fixed context. Current systems often either waste computation on easy cases or leave false detections unresolved.

This leads to a persistent gap between reliability and explainability in safety-critical settings, where these two capabilities are not fully unified within a single framework. This work proposes CAESAR++, which extends our previous context-aware explanation framework CAESAR [11] into an integrated system that:

— produces statistically valid uncertainty estimates on labels and bounding boxes;

— uses these estimates to adapt the amount of contextual information considered for each detection;

— generates dual-color saliency maps that clearly separate local features from contextual reasoning;

Although the proposed methodology is mainly designed for urban road scenes with complex interactions and challenging conditions, we also evaluate its generalization on diverse datasets under controlled out-of-context settings to demonstrate its robustness beyond the target domain.

This paper is a conference-length version of a manuscript submitted to *Neurocomputing* (Elsevier), currently under review after revision.

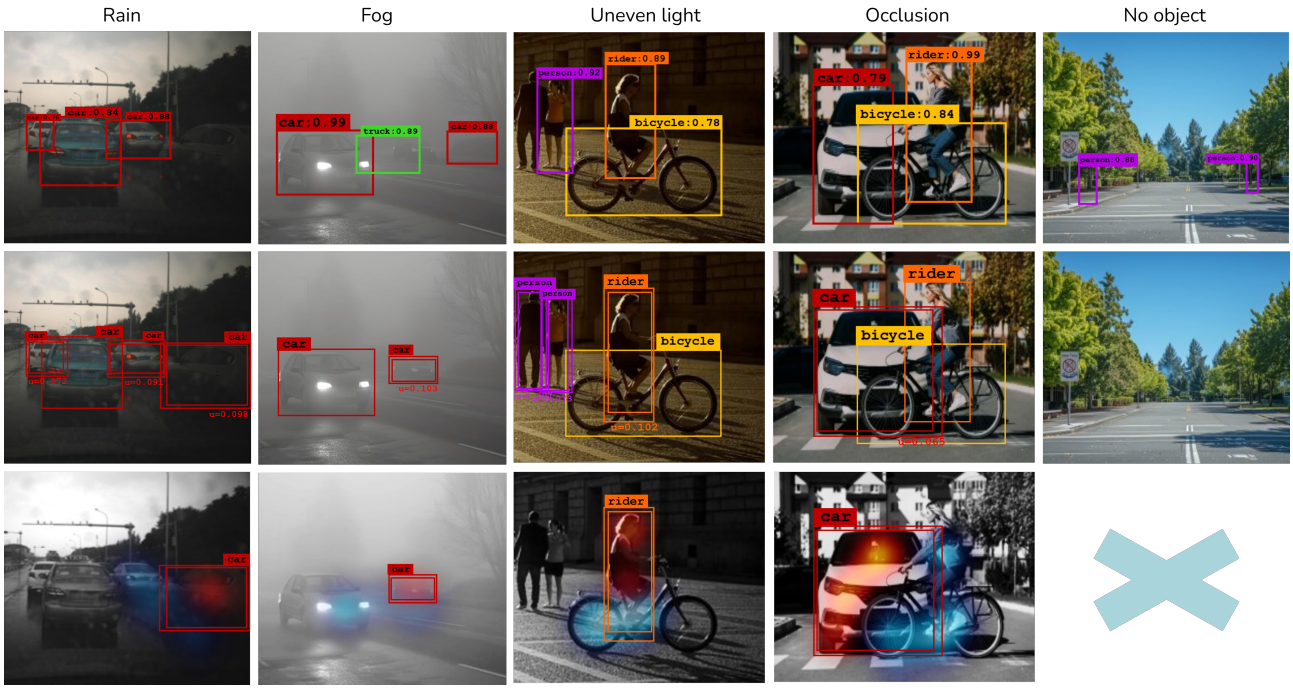


Figure 1: Overview of CAESAR++ main contributions. Top row: baseline detections in challenging urban conditions, with missed or mislocalized objects. Middle row: corresponding refined detections produced by adaptive contextual reasoning. Bottom row: dual-color saliency maps where red highlights sensory evidence and blue highlights contextual cues.

Contributions. The main contributions are as follows:

— A detector-agnostic pipeline that combines a two-step conformal prediction procedure [9] with context-driven refinement for road object detection.

— An end-to-end contextual reasoning mechanism that adaptively expands the spatial window around each detection proportionally to its calibrated uncertainty.

— A dual-color saliency scheme based on Grad-CAM++ [12, 13] that disentangles bottom-up cues from top-down context, providing instance-wise explanations with improved visual clarity.

Figure 1 illustrates consistent improvements obtained by CAESAR++ over a baseline detector in challenging scenes. Overall, the framework enhances detection performance while also making the decision process more transparent.
The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the CAESAR++ framework. Section 4 reports the main experimental results. Section 5 provides additional insights and implications. Section 6 concludes the paper and outlines future work.

2 Related work

Uncertainty in object detection. Uncertainty estimation is essential to detect overconfident errors in safety-critical applications. Bayesian formulations and dropout-based approximations [4] offer principled tools to estimate epistemic and aleatoric uncertainty, while deep ensembles provide strong empirical calibration [5]. Conformal prediction offers an alternative that yields finite-sample coverage guarantees under mild assumptions [6, 7]. Recent works adapt conformal prediction to object detection [8, 9], with separate calibration of classification and localization. These studies mainly focus on coverage metrics and interval width, and do not exploit uncertainty to steer context usage or explanation.

Explainability for detectors. Gradient-based explanation methods such as Grad-CAM and Grad-CAM++ [12, 13], and perturbation-based approaches such as RISE [14], have become standard for visualizing the internal reasoning of convolutional networks. Several extensions build object-specific saliency maps for detectors [15, 16, 17], and human attention has been proposed as an additional signal [18]. These works considerably improve instance-level interpretability, but largely treat all detections in the same manner and use fixed context ranges. As a result, explanations can become noisy on ambiguous cases and waste computation on very confident detections.

Contextual reasoning and computation strategy. Human observers strongly rely on scene context when local evidence is insufficient [19]. Modern detectors incorporate context via attention mechanisms and relation modules [20, 21], which improves robustness in cluttered scenes. In parallel, dynamic computation strategies adjust the amount of processing to the difficulty of the input [22, 23]. However, most context modules are integrated into the detector architecture and require retraining. CAESAR++ instead provides a detector-agnostic, post-hoc mechanism that triggers context expansion and refinement.

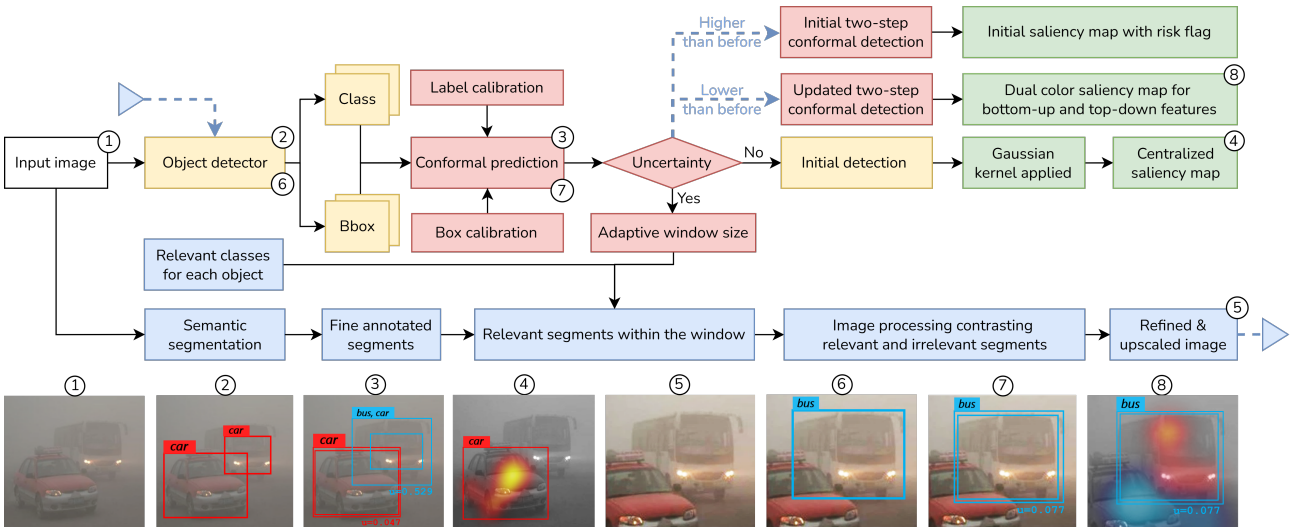


Figure 2: Overall CAESAR++ pipeline showing the successive stages from initial detection (yellow) to uncertainty estimation (red), contextual refinement (blue), and post-hoc explainability (green). The detector first generates predictions; conformal calibration then produces uncertainty scores that guide adaptive context expansion using semantic segmentation; the refined patch is reprocessed, and dual-color saliency maps are finally generated. Numbered images illustrate corresponding steps.

3 CAESAR++ framework

Figure 2 summarizes the end-to-end workflow of our proposed method. The following subsections provide a detailed description of each main component in turn.

3.1 Context modeling from segmentation

CAESAR++ builds on the context computation strategy introduced in our prior work CAESAR [11]. To model the typical contextual environment associated with each road object category, we employ a lightweight semantic segmentation network tailored to our use case. Specifically, we use DeepLabV3+ [24] with a MobileNetV2 backbone [25], pretrained on MS COCO and fine-tuned on Cityscapes [26]. The model is evaluated on selected images from TJU-DHD-Traffic [27], BDD100K [28], and Pascal VOC [29].

For each instance of a road object class, we define a local window centered at its centroid and count the distinct semantic classes intersecting this window. The resulting counts are normalized over the dataset, and for each road object category we retain the most frequent context classes as well as co-occurring road objects. Table 1 reports the resulting context sets. Road objects themselves are also considered as potential context for other road objects, reflecting the structured nature of traffic scenes [19]. This design deliberately focuses on meaningful in-domain context modeling, rather than on out-of-distribution scenarios beyond the scope of this work.

3.2 Two-step conformal calibration

For each image in a calibration set, the base detector outputs candidate bounding boxes and class probabilities. Based on the procedure adopted by [9], CAESAR++ derives nonconformity scores for labels and bounding boxes and computes empirical quantiles that will be used at test time.

Table 1: Relevant context classes for each road object category. Context categories are indicated in italics and co-occurring road objects are underlined.

Road Object	Relevant Classes
Person	*road*, *sidewalk*, *pole*, person, rider, bicycle
Rider	*road*, *sidewalk*, *wall*, rider, bicycle, motorcycle
Car	*road*, *traffic sign*, *pole*, person, rider, car
Truck	*road*, *traffic light*, *building*, car, truck, bus
Bus	*road*, *traffic sign*, *building*, car, truck, bus
Tram	*road*, *fence*, *vegetation*, person, car, bicycle
Motorcycle	*road*, *wall*, *building*, person, rider, motorcycle
Bicycle	*road*, *sidewalk*, *wall*, person, rider, bicycle

For labels, the nonconformity score is defined as one minus the predicted probability of the true class. The empirical quantile at level $(1 - \alpha)$ defines a threshold; at test time, the conformal label set for a given detection contains all classes whose complement probability does not exceed this threshold, to yield marginal coverage guarantees on labels.

For bounding boxes, we use conformalized quantile regression [30], where each coordinate is predicted with lower and upper quantiles, and the nonconformity score is defined as the largest deviation outside this predictive interval. The corresponding quantile then determines a symmetric expansion of the predicted box, yielding a conformal prediction set for localization.

To drive context adaptation, CAESAR++ introduces a normalized uncertainty score u in $[0, 1]$ by aggregating label and box nonconformity, which is then used to derive the adaptive threshold for context expansion. Low values indicate confident detections, whereas high values correspond to ambiguous cases.

Figure 3: From left to right: baseline detections, refined detections with CAESAR++, and dual-color explanations for individual objects. Red indicates sensory evidence, blue indicates contextual support. Uncertainty scores omitted for clarity.

3.3 Adaptive context selection

Given the conformal outer box and the normalized uncertainty score of a detection, CAESAR++ determines how much surrounding context should be incorporated. Confident detections are left unchanged, whereas uncertain ones are reprocessed with a larger contextual region.

The context window is expanded using a simple linear rule that progressively enlarges the region from the outer box toward the full image as uncertainty increases. This choice provides a direct and monotonic relationship between uncertainty and context expansion, while remaining easy to interpret and free of additional hyperparameters. We also evaluated nonlinear mappings, but we retained the linear formulation for its simplicity and stable behavior.

To avoid unnecessary reprocessing, CAESAR++ further relies on a data-driven certainty criterion derived from the recent distribution of uncertainty scores [31]. Detections whose uncertainty falls below a low adaptive threshold are processed directly with the original box, while the remaining detections undergo contextual enhancement. Within the selected region, semantic masks preserve the relevant classes listed in Table 1 and suppress irrelevant areas before the refined patch is fed back to the detector.

3.4 Dual-color saliency maps

To explain individual detections, CAESAR++ constructs two complementary saliency maps per object using Grad-CAM++ [12, 13] as the baseline approach.

A bottom-up map is obtained by applying Grad-CAM++ to the original detection without context expansion. It captures the direct sensory evidence for the prediction. A second map is computed for the refined detection after context refinement. The difference between the refined and original maps highlights top-down contextual cues.

Bottom-up saliency is visualized in red and top-down contribution in blue. For detections classified as certain, only the bottom-up component is displayed, smoothed by a Gaussian kernel [32] to enhance spatial coherence. For uncertain detections, both components are shown, providing a clear view of how context modifies the decision.

As illustrated in Fig. 3, this representation improves interpretability over the initial detector output. It offers a compact and informative view of the interplay between appearance and context, helping to understand both the detection update and the underlying decision mechanism.

4 Results

4.1 Experimental setup

We evaluate CAESAR++ on three datasets: TJU-DHD-Traffic [27], BDD100K [28] and Pascal VOC 2012 [29]. Each contributes 5000 validation images. TJU-DHD-Traffic contains diverse driving scenes with strong environmental variations. BDD100K covers dense urban environments. Pascal VOC includes a broader set of scenes and objects, which is less specific to driving but useful to test generalization across various scenarios.

We consider six detectors representing one-stage, two-stage and transformer-based paradigms, including YOLOv8m [33], IA-YOLO [34], EfficientDet-D4 [35], L-SSD [36], RT-DETR-R50 [37], and Faster R-CNN [38]. CAESAR++ is applied as a post-processing module without retraining.

Detection performance is measured with mAP@0.5:0.95 and mAP@0.5, false negative rate (FNR), false discovery rate (FDR), F1-score and frames per second (FPS). Uncertainty quality is assessed through label and box coverage and mean prediction interval width [7]. Explanations are evaluated with Deletion and Insertion metrics [14], the Energy-Based Pointing Game (EBPG) [39], and Average Stability (AvgS) for robustness [40].

Table 2: Detection results comparison on TJU-DHD-Traffic validation set with and without CAESAR++ (95% CI). Metrics reported: mAP@0.50:0.95 (↑), mAP@0.5 (↑), False Negative Rate (FNR, ↓), False Discovery Rate (FDR, ↓), F1-Score (↑), and FPS (↑). Better results are **bolded**.

Detector brief description	Method	mAP@0.50:0.95	mAP@0.5	FNR ↓	FDR ↓	F1-Score ↑	FPS ↑
One-stage, anchor-based	(1) YOLOv8m	50.8 ± 1.4	73.4 ± 1.5	8.2 ± 0.8	7.8 ± 0.5	78.2 ± 1.7	58.1 ± 0.6
	(1) + CAESAR++	$\mathbf{53.7 \pm 0.7}$	$\mathbf{79.1 \pm 0.8}$	$\mathbf{4.8 \pm 0.4}$	$\mathbf{4.2 \pm 0.3}$	$\mathbf{82.9 \pm 0.7}$	52.8 ± 1.8
One-stage, anchor-based, image adaptive processing	(2) IA-YOLO	52.2 ± 1.2	75.1 ± 1.4	5.1 ± 0.8	5.4 ± 0.7	80.5 ± 1.7	79.7 ± 1.2
	(2) + CAESAR++	$\mathbf{54.8 \pm 0.8}$	$\mathbf{80.3 \pm 0.7}$	$\mathbf{2.7 \pm 0.3}$	$\mathbf{2.2 \pm 0.2}$	$\mathbf{84.7 \pm 0.8}$	75.8 ± 2.1
One-stage, EfficientNet architecture	(3) EfficientDet-D4	52.5 ± 1.1	75.8 ± 0.9	7.2 ± 0.7	4.9 ± 0.5	80.8 ± 1.2	35.8 ± 0.5
	(3) + CAESAR++	$\mathbf{55.4 \pm 0.7}$	$\mathbf{81.2 \pm 0.5}$	$\mathbf{4.3 \pm 0.4}$	$\mathbf{2.1 \pm 0.3}$	$\mathbf{84.1 \pm 0.5}$	32.1 ± 1.1
One-stage, anchor-based, lightweight	(4) L-SSD	46.9 ± 1.0	71.4 ± 0.8	8.7 ± 0.6	4.9 ± 0.5	76.9 ± 0.8	99.9 ± 0.7
	(4) + CAESAR++	$\mathbf{50.4 \pm 0.6}$	$\mathbf{77.1 \pm 0.4}$	$\mathbf{5.4 \pm 0.4}$	$\mathbf{2.8 \pm 0.2}$	$\mathbf{80.3 \pm 0.4}$	94.3 ± 1.6
Transformer-based, real-time end-to-end	(5) RT-DETR-R50	51.2 ± 0.9	73.8 ± 0.8	6.9 ± 0.6	4.8 ± 0.4	81.4 ± 1.0	68.3 ± 1.6
	(5) + CAESAR++	$\mathbf{53.5 \pm 0.5}$	$\mathbf{77.0 \pm 0.4}$	$\mathbf{4.3 \pm 0.4}$	$\mathbf{2.2 \pm 0.3}$	$\mathbf{84.2 \pm 0.6}$	66.1 ± 2.1
Two-stage, region proposal networks	(6) Faster R-CNN	53.4 ± 0.4	75.4 ± 0.5	4.7 ± 0.4	4.2 ± 0.3	82.3 ± 0.7	9.8 ± 0.2
	(6) + CAESAR++	$\mathbf{56.6 \pm 0.2}$	$\mathbf{80.9 \pm 0.3}$	$\mathbf{2.1 \pm 0.2}$	$\mathbf{1.3 \pm 0.2}$	$\mathbf{86.2 \pm 0.4}$	7.4 ± 0.8

Table 3: Accuracy gains in mAP@0.50:0.95 after applying CAESAR++ on YOLOv8m across object sizes and eight road object classes in cross-dataset validation. Object sizes are stratified following the COCO challenge. Values are mean ± standard deviation in percentage points.

Train	Test	Small	Medium	Large	Person	Rider	Car	Truck	Bus	Tram	M.cycle	Bicycle
TJU-DHD	PascalVOC	5.1±0.6	3.0±0.4	1.1±0.3	4.3±0.5	3.9±0.6	3.4±0.4	1.8±0.4	1.7±0.3	0.7±0.3	5.2±0.7	5.4±0.8
TJU-DHD	BDD100K	5.9±0.5	3.8±0.4	1.4±0.2	5.2±0.4	4.5±0.5	3.8±0.3	2.0±0.3	1.9±0.2	0.8±0.2	6.3±0.6	6.4±0.6
PascalVOC	TJU-DHD	3.6±0.7	2.3±0.5	0.9±0.3	3.6±0.5	3.5±0.6	3.1±0.4	1.5±0.5	1.4±0.4	0.3±0.1	4.7±0.7	4.6±0.8
PascalVOC	BDD100K	4.0±0.7	2.5±0.5	0.9±0.3	3.8±0.6	3.7±0.5	3.3±0.4	1.6±0.5	1.4±0.4	0.4±0.1	4.8±0.7	4.8±0.7
BDD100K	TJU-DHD	5.5±0.6	4.2±0.4	1.3±0.2	5.0±0.4	4.4±0.5	3.8±0.3	1.9±0.3	1.8±0.3	0.8±0.2	6.1±0.6	6.2±0.6
BDD100K	PascalVOC	4.7±0.6	2.8±0.4	1.1±0.2	3.9±0.5	3.9±0.6	3.4±0.4	1.8±0.5	1.7±0.3	0.5±0.3	5.1±0.7	5.3±0.7

4.2 Detection performance

Table 2 presents results on TJU-DHD-Traffic for a subset of detectors, with and without CAESAR++. In all cases, CAESAR++ improves mAP and reduces both FNR and FDR. The gain in mAP@0.5:0.95 reaches about three percentage points for some detectors, and the reduction in FNR and FDR often exceeds fifty percent relative.

An important trend is that FDR often decreases nearly as much as FNR. This suggests that CAESAR++ is effective at suppressing uncertain false alarms by incorporating semantically relevant context, while still reducing missed detections. In safety-critical road perception, this is a desirable trade-off: fewer false positives improve reliability, and fewer false negatives better support safe navigation.

From a deployment perspective, the computational cost remains moderate because only uncertain detections are reprocessed. As a result, the FPS drop is limited for the one-stage and transformer-based detectors, whereas Faster R-CNN remains the computational bottleneck in absolute terms due to its inherently heavier design. The additional computation depends on the proportion of uncertain detections. Detectors with stronger baselines generate fewer uncertain cases and therefore incur smaller slowdowns. In most configurations CAESAR++ maintains real-time performance of the base models, for example more than 70 FPS with IA-YOLO.

Cross-dataset experiments with YOLOv8m (Table 3) show that the largest gains are achieved for small objects and for classes such as pedestrians, riders, bicycles and motorcycles, which are both critical for safety and particularly difficult to detect. This is consistent with the fact that small instances provide limited appearance evidence inside the original bounding box, so neighboring structure becomes more valuable for disambiguation.

4.3 Visual explanations

Figure 3 shows qualitative examples with YOLOv8m. The baseline detector misses several objects and sometimes misplaces bounding boxes. After CAESAR++ refinement, detections are more accurate, and the dual-color visual explanations reveal which parts of the image support each decision. Red regions capture the object appearance itself, whereas blue regions highlight contextual structures such as road, sidewalks or nearby vehicles. This separation is particularly useful when local appearance is weak, because it makes explicit how contextual information helps resolve ambiguity caused by occlusion, clutter, or low contrast.

Table 4 quantitatively compares CAESAR++ with three object-specific peer explainers [15, 16, 17] across detectors. The reported metrics capture complementary aspects of explanation quality. Insertion measures how much the detector score increases as salient pixels are progressively revealed, Deletion measures how quickly the score decreases as they are removed, EBPG evaluates spatial alignment with the ground-truth bounding box, and AvgS assesses the stability of explanations under perturbations.

Table 4: Comparison of object-specific explainers on TJU-DHD-Traffic with YOLOv8m, Faster R-CNN and RT-DETR-R50. Metrics: Deletion (Del., ↓), Insertion (Ins., ↑), EBPG (↑), AvgS (↓). The best result for each metric is in **bold**.

Method	YOLOv8m				Faster R-CNN				RT-DETR-R50			
	Del.	Ins.	EBPG	AvgS	Del.	Ins.	EBPG	AvgS	Del.	Ins.	EBPG	AvgS
D-RISE	0.22	0.54	0.51	0.13	0.21	0.57	0.51	0.13	0.24	0.54	0.47	0.15
D-CLOSE	**0.19**	0.55	0.55	0.11	0.21	0.66	**0.78**	0.16	0.22	0.56	0.55	0.12
D-MFPP	0.22	0.58	**0.64**	0.09	**0.18**	0.63	0.69	0.09	**0.18**	0.57	**0.59**	0.10
CAESAR++	0.21	**0.65**	0.62	**0.08**	0.17	**0.69**	0.71	**0.07**	0.19	**0.64**	0.56	**0.09**

CAESAR++ achieves the best Insertion and AvgS results, showing that its explanations consistently identify decision-relevant regions and remain reliable under small input changes. Its Deletion scores are also competitive, which indicates that the highlighted areas are not only visually meaningful but also influential for the detector response. The slightly lower EBPG values are consistent with the design of CAESAR++. Unlike other methods that restrict explanations to the object interior, CAESAR++ is explicitly context-aware and may attribute part of the decision to surrounding structures when they help disambiguate the detection. This is not a limitation of the method, but a reflection of the fact that road-object recognition often depends on scene context, especially for uncertain, partially occluded, or small instances.

As a result, CAESAR++ provides explanations that are less box-constrained but more informative and interpretable, since they expose both what the detector directly sees and how context changes the final decision.

5 Discussion

CAESAR++ is particularly effective in urban driving scenes, where the semantic relationships learned during context modeling remain representative of the scene at test time. In this setting, the uncertainty score serves as a meaningful indicator of ambiguity, the adaptive window retrieves relevant surrounding cues, and the explanation branch makes the contribution of context explicit. This behavior matches the intended design of the framework, which is to exploit regular scene structure in order to improve both detection reliability and interpretability, while remaining fully compatible with the underlying detector.

Its behavior is less favorable when the scene departs from the semantic prior encoded during context construction. In out-of-distribution or unusual situations, uncertainty may still increase, but the contextual evidence available in the enlarged window may be incomplete or poorly aligned with the expected traffic pattern. In such cases, the refinement stage is triggered on weaker contextual support, and the improvement in detection or explanation quality may be limited. This is an inherent consequence of the method's design, which targets structured road scenes rather than open-world conditions with unsupported semantics.

A second potential limitation arises in sparse scenes or in configurations where informative cues lie outside the adaptive crop. In such cases, the method may still detect ambiguity correctly, but the expanded region does not necessarily provide enough additional semantic evidence to justify the extra computation. This leads to a diminishing return effect, where refinement is correctly triggered but the contextual gain remains limited. For this reason, in latency-sensitive deployment scenarios, contextual reprocessing should be applied selectively, for instance by capping the number of refined detections or by restricting the mechanism to the object classes for which context is most informative and operationally relevant.

Regarding the balance between error types, the framework lowers both false negatives and false discoveries, but not always to the same extent. In real-world perception, this trade-off should be interpreted in light of the downstream application: missed detections are generally more critical than occasional false alarms, especially for vulnerable road users. As a result, the operating point should be chosen according to the target system requirements, with priority given to recall when safety is the primary concern. The latency overhead introduced by CAESAR++ is therefore acceptable when a moderate increase in computation is compatible with the application, but it should be managed carefully in real-time deployments.

6 Conclusion and perspectives

This paper presented CAESAR++, a detector-agnostic framework for road object detection that jointly addresses uncertainty quantification, adaptive contextual reasoning and visual explainability. By coupling a two-step conformal prediction with context-aware refinement and dual-color saliency maps, CAESAR++ improves detection accuracy, calibrates uncertainty and yields robust, plausible, and interpretable explanations. Experiments across multiple detectors and datasets show consistent gains, especially for small and difficult objects, with reasonable cost.

Future work will focus on reducing computational overhead through lighter segmentation backbones, single-pass refinement without re-detection, and more selective contextual expansion. We also plan to explore online recalibration and adaptive preprocessing to reduce reliance on segmentation in atypical scenes, while extending the framework to stronger domain shifts and related tasks such as instance segmentation and tracking.

Acknowledgments

Most of the computations presented in this paper were performed using the GRICAD infrastructure (https://gricad.univ-grenoble-alpes.fr), which is supported by the Grenoble research community.

References

[1] Narges Saeedizadeh, Seyed Mohammad Jafar Jalali, Burhan Khan, and Shady Mohamed. Cutting-edge deep learning methods for image-based object detection in autonomous driving: In-depth survey. *Expert Systems*, 42(4), 2025.

[2] Tirupathamma Mudavath and Anooja Mamidi. Object detection challenges: Navigating through varied weather conditions—a comprehensive survey. *Journal of Ambient Intelligence and Humanized Computing*, 16(2):443–457, 2025.

[3] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models, 2017.

[4] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

[5] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30:6405–6416, 2017.

[6] Harris Papadopoulos, Volodya Vovk, and Alex Gammerman. Conformal prediction with neural networks. In *19th IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pages 388–395, 2007.

[7] Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2022.

[8] Leo Andeol, Thomas Fel, Florence de Grancey, and Luca Mossina. Confident object detection via conformal prediction and conformal risk control: An application to railway signaling. In *Symposium on Conformal and Probabilistic Prediction with Applications*, volume 204, pages 36–55. PMLR, 2023.

[9] Alexander Timans, Christoph-Nikolas Straehle, Kaspar Sakmann, and Eric Nalisnick. Adaptive bounding box uncertainties via two-step conformal prediction. In *European Conference on Computer Vision*, pages 363–398, 2025.

[10] Ruoxi Qi, Guoyang Liu, Jindi Zhang, and Janet Hui-Wen Hsiao. Do saliency-based explainable ai methods help us understand ai's decisions? the case of object detection ai. In *Annual Meeting of the Cognitive Science Society*, volume 46, Rotterdam, the Netherlands, 2024.

[11] Anh-Thu Mai, Marina Nicolas, Patricia Ladret, and Alice Caplier. Robust road object detection with caesar: Context-aware explanations via semantic attribution and refinement. In *IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5, 2025.

[12] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[13] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*, pages 839–847, 2018.

[14] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models, 2018.

[15] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I. Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11438–11447, 2021.

[16] Van Binh Truong, Truong Thanh Hung Nguyen, Vo Thanh Khang Nguyen, Quoc Khanh Nguyen, and Quoc Hung Cao. Towards better explanations for object detection. In *Asian Conference on Machine Learning*, volume 222 of *Proceedings of Machine Learning Research*, pages 1385–1400, 2024.

[17] Alain Andres, Aitor Martinez-Seras, Ibai Laña, and Javier Del Ser. On the black-box explainability of object detection models for safe and trustworthy industrial applications. *Results in Engineering*, 24:103498, 2024.

[18] Guoyang Liu, Jindi Zhang, Antoni B. Chan, and Janet H. Hsiao. Human attention guided explainable artificial intelligence for computer vision models. *Neural Networks*, 177:106392, 2024.

[19] Aude Oliva and Antonio Torralba. The role of context in object recognition. *Trends in Cognitive Sciences*, 11(12):520–527, 2007.

[20] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018.

[21] Xinfang Zhong, Wenlan Kuang, and Zhixin Li. Adaptive graph reasoning network for object detection. *Image and Vision Computing*, 151, 2024.

[22] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *European Conference on Computer Vision*, pages 420–436, 2018.

[23] Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S. Davis, Kristen Grauman, and Rogerio Feris. Blockdrop: Dynamic inference paths in residual networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8817–8826, 2018.

[24] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, pages 833–851, 2018.

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[26] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[27] Yanwei Pang, Jiale Cao, Yazhao Li, Jin Xie, Hanqing Sun, and Jinfeng Gong. Tju-dhd: A diverse high-resolution dataset for object detection. *IEEE Transactions on Image Processing*, 30:207–219, 2021.

[28] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2633–2642, 2020.

[29] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. Pascal visual object classes 2012, 2025.

[30] Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. Conformalized quantile regression. In *Advances in Neural Information Processing Systems*, pages 3543–3553, 2019.

[31] Fabian Küppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1322–1330, 2020.

[32] Saumya Jetley, Naila Murray, and Eleonora Vig. End-to-end saliency mapping via probability distribution prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5753–5761, 2016.

[33] Rejin Varghese and M. Sambath. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6, 2024.

[34] Wenyu Liu, Gaofeng Ren, Runsheng Yu, Shi Guo, Jianke Zhu, and Lei Zhang. Image-adaptive yolo for object detection in adverse weather conditions. In *AAAI Conference on Artificial Intelligence*, pages 1792–1800, 2022.

[35] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10778–10787, 2020.

[36] Huilin Wang, Huaming Qian, Shuai Feng, and Wenna Wang. L-ssd: Lightweight ssd target detection based on depth-separable convolution. *Journal of Real-Time Image Processing*, 21(2), 2024.

[37] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection, 2023.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[39] Jianming Zhang, Saeed Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *International Journal of Computer Vision*, pages 1084–1102, 2018.

[40] David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods, 2018.

Approche hybride pour la détection du levé de stylo à partir de vidéos : preuve de concept pour une analyse complémentaire de l'écriture

Lauren Sismeiro[1], Rémy Plastre[2], Binbin Xu[1], Frédéric Puyjarinet[1], Gérard Dray[1]

[1] EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Alès, France
[2] IMT Mines Alès, Alès, France

lauren.sismeiro@mines-ales.fr, remy.plastre@mines-ales.org

Résumé

L'analyse dynamique de l'écriture est essentielle pour l'évaluation de la dysgraphie, mais les tablettes graphiques ne capturent que les mouvements proches de la surface. Nous proposons une preuve de concept visant à détecter les levés de stylo à partir de vidéos en vue de dessus. L'approche repose sur une chaîne de traitement hybride combinant suivi de pointe du stylo, descripteurs cinématiques multi-échelles et classification supervisée. Évaluée en Leave-One-Video-Out, la méthode atteint un score F_2 de 0.805*, suggérant que l'analyse vidéo constitue un complément pertinent et peu coûteux aux dispositifs existants.*

Mots-clés

Analyse de l'écriture manuscrite, Vision par ordinateur, Suivi de stylo, Cinématique, Dysgraphie, Santé numérique.

1 Introduction

Le diagnostic de la dysgraphie repose notamment sur l'échelle BHK [1], fondée sur l'analyse statique de l'écriture, sans accès à la dynamique du geste, pourtant essentielle pour caractériser le contrôle moteur. Les tablettes numériques capturent ces informations, mais restent limitées aux mouvements proches de la surface [2], excluant les levées de grande amplitude potentiellement informatives [4]. Nous proposons d'explorer la vision par ordinateur comme modalité complémentaire. À partir de vidéos en vue de dessus, nous évaluons dans une approche de preuve de concept, la capacité d'une chaîne de traitement hybride à détecter les états de contact (*Pen-Down*) et de levé (*Pen-Up*). Une version étendue de ce travail est disponible [5].

2 Méthode

2.1 Collecte de données et annotation

Cinq vidéos d'écriture manuscrite ont été extraites de YouTube[1], couvrant divers styles (cursive, script), stylos (pointes fines/épaisses, encres bleue/noire) et supports (interlignes variables). Le jeu de données ainsi créé comprenait 13 507 images après échantillonnage à 30 fps, d'une résolution de 1080p. Chaque frame a été annotée de façon binaire par 3 évaluateurs indépendants en *Pen-Down* ou *Pen-Up*, en excluant les images non informatives, atteignant un accord inter-juges de $89\,\%$ et un coefficient Kappa de Fleiss de $0,78$. Après agrégation via soft labelling, on comptait $28,7\,\%$ d'images étiquetées *Pen-Up*.

1. DorufaVSArt, `https://www.youtube.com/@DorufaVSArt`

2.2 Métrique d'évaluation

Le score F_2 dans la détection des évènements *Pen-Up* a été choisi comme la métrique prioritaire, afin de privilégier des modèles sensibles avec un rappel élevé plus adaptés en contexte de dépistage.

2.3 Chaîne de traitement en quatre étapes

L'approche proposée, illustrée Figure 1, repose sur une chaîne de traitement hybride séparant explicitement la localisation de la pointe du stylo et l'inférence de son état.

1. Suivi de la pointe du stylo. La position de la pointe (u, v) a été estimée image par image à l'aide du modèle de détection d'objets YOLOv11m, selon un protocole Leave-One-Video-Out. Le modèle a atteint une erreur médiane de 3,28 px (P95 : 7,44 px), avec 77,7 % des prédictions < 5 px et 99,2 % < 10 px, permettant d'obtenir une trajectoire fiable, y compris lors de mouvements rapides ou de levés importants comme l'illustre la Figure 2.

2. Extraction de caractéristiques cinématiques. À partir des coordonnées (u, v), 147 descripteurs cinématiques ont été extraits, comprenant notamment des caractéristiques locales multi-échelles (fenêtres glissante de 3 à 16 images) de linéarité, de variabilité angulaire et de variations de vitesse, ainsi que des descripteurs globaux (inclinaison moyenne, vitesse normalisée) tenant compte du style d'écriture.

3. Classification supervisée. Random Forest, HistGBM, LightGBM ainsi qu'une approche par Ridge stacking ont été entraînés et évalués selon le protocole Leave-One-Video-Out, afin de maximiser la F_2 des probabilités $P_{\text{Pen-Up}}$ par image, incluant une optimisation des hyperparamètres via la librairie Optuna (totalisant 100 essais par modèle).

4. Post-traitement événementiel. Les probabilités frame par frame ont été converties en segments temporels (évènements *Pen-Up* et *Pen-Down*) grâce à un post-traitement combinant : **(i)** un lissage par hystérésis pour stabiliser les transitions, **(ii)** un filtrage morphologique pour supprimer les détections bruitées, et **(iii)** un alignement cinématique.

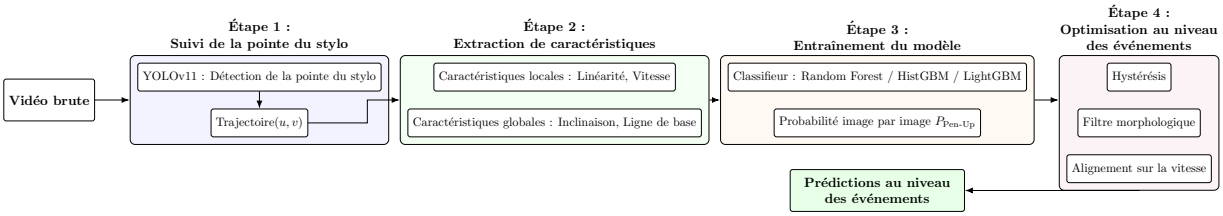


FIGURE 1 – Chaîne de traitement hybride pour la détection du contact du stylo en vidéos d'écriture manuscrite vues de dessus.

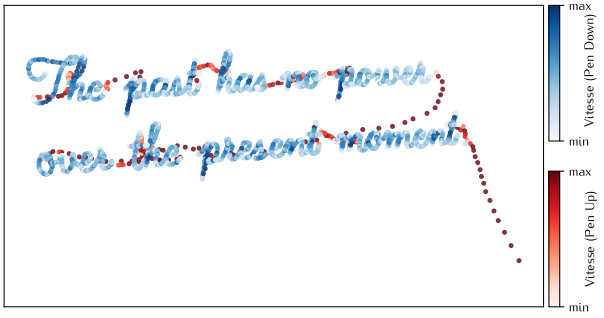


FIGURE 2 – Visualisation de la trajectoire du stylo détectée par Yolov11m (Bleu : *Pen-Down*, Rouge : *Pen-Up*, vitesse codée en intensité)

3 Résultats

L'évaluation a été réalisée au niveau événementiel, afin de détecter des segments de levé de stylo plutôt que des prédictions image par image. Plusieurs tolérances temporelles de décalage entre prédiction et annotation ont ainsi été testées (0 à 20 frames). Les meilleures performances ont été obtenues via les modèles de type gradient boosting. À une tolérance de 12 images ($\approx$ 400 ms, durée considérée comme cliniquement pertinente pour les pauses en écriture [3]), le modèle LightGBM a atteint un score F_2 de $0,805$ avec un rappel de $0,880$. La Figure 3 illustre l'évolution du rappel et du score F_2 en fonction de la tolérance temporelle. L'augmentation de cette tolérance améliorait les performances, en particulier à faible valeur (< 5 frames). Les gains observés entre 10 et 12 images étaient quant à eux limités, suggérant un plateau de performance.

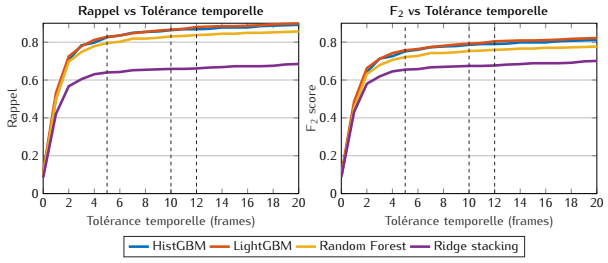


FIGURE 3 – Rappel et Score F_2 en fonction de la tolérance pour les quatre modèles évalués.

4 Discussion

Cette étude démontre la faisabilité de la détection des états *Pen-Up* à partir de vidéos en vue de dessus, comme complément potentiel aux tablettes numériques, en se basant sur des descripteurs cinématiques interprétables, plus adaptés à un contexte clinique, et pouvant se révéler de potentiels indicateurs du geste d'écriture pour les professionels de santé.

Limites et perspectives. Le jeu de données réduit limite la généralisabilité des résultats. Par ailleurs, l'utilisation d'une seule caméra ne permet pas d'estimer directement la hauteur du stylo, les états *Pen-Up* étant inférés à partir de la projection dans le repère de la caméra. Des approches multi-vues pourraient permettre d'accéder à cette information.

5 Conclusion

Ce travail démontre la faisabilité d'une analyse cinématique de l'écriture manuscrite par vidéo. En permettant le suivi de la trajectoire au-delà de la surface d'écriture, cette approche ouvre la voie à une caractérisation des dynamiques scripturales, avec des applications en évaluation de la dysgraphie.

Remerciements

Les auteurs remercient le créateur DorufaVSArt pour l'autorisation d'utilisation des vidéos, ainsi que Romain Sebire pour sa contribution à l'annotation des images.

Références

[1] M. Charles, R. Soppelsa, and J.-M. Albaret. *BHK – Échelle d'évaluation rapide de l'écriture chez l'enfant*. Éditions Centre de Psychologie Appliquée, 2004.

[2] Jean-Claude Gilhodes, Elie Fabiani, Marieke Longcamp, Jean-Luc Velay, and Jérémy Danna. Chapitre 4. traiter des données de langage écrit recueillies avec tablette graphique. In *Introduction aux statistiques en sciences du langage*, pages 117–136. Dunod, 2023.

[3] Mariona Pascual, Olga Soler, and Naymé Salas. In a split second : Handwriting pauses in typical and struggling writers. *Frontiers in Psychology*, 13, 2023.

[4] Sara Rosenblum, Shula Parush, and Patrice L. Weiss. The in Air phenomenon : Temporal and spatial correlates of the handwriting process. *Perceptual and Motor Skills*, 96(3) :933–954, June 2003.

[5] Lauren Sismeiro, Remy Plastre, Binbin Xu, Frederic Puyjarinet, and Gerard Dray. Detecting Pen-In-Air states from video : A proof-of-concept toward complementary handwriting analysis. *arXiv preprint arXiv :2606.02342*, 2026.

Reliability-Aware Fusion for Semantic Segmentation under Sensor Degradation and Failures

Abdelhak Benamirouche[1,*], Lucas Deregnaucourt[2,*], Mihreteab Negash Geletu[1]
Hind Laghmara[2], Remi Boutteau[2], Jean-Philippe Lauffenburger[1]

[1] IRIMAS-UR7499, Université de Haute-Alsace, Mulhouse, France.
[2] INSA Rouen Normandie, Univ Rouen Normandie, Université Le Havre Normandie, Normandie Univ, LITIS UR 4108, F-76000 Rouen, France.
[*] Equal contribution.

Résumé

La segmentation sémantique dans des scénarios de conduite réels est particulièrement difficile en raison de la dégradation des capteurs, des pannes et des conditions environnementales changeantes. Bien que la fusion multimodale soit une solution couramment utilisée, de nombreuses approches existantes traitent toutes les modalités de manière équivalente, en ignorant leur fiabilité variable selon les classes sémantiques et les conditions. Dans cet article, nous présentons ReCoLaF (Reliability-aware Conflict-guided Late Fusion), un nouveau cadre de fusion profonde pour la segmentation sémantique multimodale en présence d'incertitude. ReCoLaF ajuste de manière adaptative la contribution de chaque modalité de capteur grâce à une stratégie de pondération en deux étapes : un module de fiabilité appris, spécifique aux classes, qui estime la pertinence de chaque modalité pour différentes classes sémantiques, ainsi qu'un ajustement basé sur le conflit qui mesure les incohérences locales entre modalités au niveau du pixel. La fusion est formulée dans le cadre de la théorie de l'évidence de Dempster-Shafer, offrant une approche mathématiquement fondée pour gérer l'incertitude et produire des prédictions robustes. Nous évaluons ReCoLaF sur les jeux de données DeLiVER (synthétique) et MUSES (réel) dans diverses conditions météorologiques et configurations de capteurs dégradés. ReCoLaF obtient systématiquement de meilleures performances moyennes en présence de défaillances de capteurs, mettant en évidence l'intérêt de modéliser conjointement la fiabilité sémantique et l'accord inter-modalités pour une fusion robuste dans des scénarios de conduite complexes.

Mots-clés

Segmentation sémantique, Fusion multimodale, Dégradation des capteurs, Théorie de Dempster-Shafer, Conduite autonome.

Abstract

Semantic segmentation in real-world driving scenarios is particularly challenging due to sensor degradation, failures, and changing environmental conditions. While multimodal fusion is a common solution, many existing approaches treat all modalities equally, ignoring their varying reliability across semantic classes and conditions. In this paper, we present ReCoLaF (Reliability-aware Conflict-guided Late Fusion), a novel deep fusion framework for multimodal semantic segmentation under uncertainty. ReCoLaF adaptively adjusts the contribution of each sensor modality through a two-stage weighting strategy: a learned class-wise reliability module that estimates how relevant each modality is for different semantic classes, and a conflict-based adjustment that measures local inconsistencies between modalities at the pixel level. The fusion is formulated within the Dempster-Shafer theory of evidence, providing a mathematically grounded approach to handle uncertainty and make robust predictions. We evaluate ReCoLaF on the DeLiVER (synthetic) and MUSES (real-world) datasets under diverse weather conditions and degraded sensor configurations. ReCoLaF consistently achieves higher average performance under sensor failures, highlighting the benefit of jointly modeling semantic reliability and inter-modality agreement for robust fusion in complex driving scenarios.

Keywords

Semantic segmentation, Multimodal fusion, Sensor degradation, Dempster-Shafer theory, Autonomous driving.

1 Introduction

Semantic segmentation is a critical component of environment perception in autonomous driving systems. However, performance can be significantly degraded in real-world scenarios due to both external and internal factors. Externally, the driving scene is often unstructured, with occlusions, object truncations, and variable weather conditions such as rain, fog, or snow. Illumination changes from day to night or reflections further complicate scene understanding. Internally, perception sensors have inherent limitations: cameras provide rich color and texture but

are sensitive to lighting; LiDAR and radar are robust to illumination but have lower resolution; event cameras handle motion and lighting variations well but lack color information. Moreover, sensor failures or degradation can occur during deployment, making the need for robust perception ever more pressing.

To overcome these challenges, multimodal sensor fusion is widely used to integrate complementary information and to improve the robustness of perception systems. Recent deep learning-based fusion architectures show high results. They combine modalities at various stages using operations such as concatenation, addition, ensembles or mixtures of experts [8]. These methods have been applied to object detection, road segmentation, and semantic scene understanding [1, 21]. More recently, evidential deep learning has been introduced to model uncertainty explicitly in multimodal fusion [20, 10]. Based on evidence theory, these models provide extended mechanisms for uncertainty handling and more informed fusion strategies.

In real-world driving scenes, different modalities are not equally informative for all object classes. For instance, LiDAR may perform better for detecting structures, while RGB excels at recognizing signs or road markings. Moreover, in degraded settings—such as fog or partial sensor failure—some modalities become unreliable or even detrimental. To address this, robust fusion should dynamically adjust modality contributions based on their relevance and consistency. In this work, we propose an evidential deep fusion framework for semantic segmentation that adaptively fuses multiple modalities by jointly modeling both modality-specific class-level reliability and inter-modality conflict.

The organization of this paper is as follows. Section 2 reviews related work in semantic and multimodal perception, with a focus on recent approaches to evidential fusion and sensor-aware modeling. Section 3 introduces the fundamentals of evidence theory as the underlying framework for our evidential reasoning. Section 4 presents our proposed architecture, ReCoLaF, detailing its four main components: evidential encoder-decoders, sensor reliability estimation, conflict-based adjustment, and evidence fusion. Section 5 describes the experimental setup, datasets, and implementation details, followed by a thorough evaluation of our method across various sensor combinations and conditions. We conclude the paper in Section 6 with a summary of contributions and directions for future research.

2 Related work

Robust and accurate scene understanding can be critical under diverse and often challenging environmental conditions. Single-modality sensors, such as RGB cameras, LiDAR, and radar, each offer unique advantages —color and texture, precise depth, and weather resilience, respectively— but also exhibit limitations when faced with low visibility, adverse weather, or sensor degradation. To address these challenges, **multimodal perception** integrates complementary information from multiple sensors, enabling perception systems to compensate for the weaknesses of individual modalities and improving performance in tasks such as detection, segmentation, localization, and navigation. Early efforts focused on enhancing LiDAR-based 3D detection with RGB data, supported by benchmark datasets like KITTI [9] and nuScenes [4]. However, these lacked coverage of adverse conditions. Fusion strategies have evolved from fixed dual-modality combinations to flexible RGB-X fusion frameworks capable of handling arbitrary modality sets. Recent methods such as CMNeXt [23] employ modular attention-based architectures, while approaches like StitchFusion [13] explore large-scale pre-trained backbones and modality-specific encoding.

While these works enhance perception in challenging environments, they typically fuse sensor inputs uniformly and do not explicitly model the reliability or contextual relevance of each modality. To address this, Huang et al. [11] proposed a fusion framework under the Dempster-Shafer theory that introduces contextual discounting based on class-specific reliability estimation for each modality. This approach adjusts the influence of each sensor depending on its expected semantic reliability, allowing more informed evidence fusion. Other approaches have explored how inter-sensor disagreement can guide adaptive fusion. Deregnaucourt et al. [5] introduced ECoLaF, a conflict-guided fusion framework that discounts modality contributions based on local disagreement with other modalities. ECoLaF demonstrated strong robustness in degraded conditions by dynamically adjusting trust in each sensor at the pixel level. In parallel, CAFuser [3] explored environmental condition-aware fusion mechanisms, enabling models to adapt to external factors such as fog, rain, or night-time illumination, rather than relying solely on the sensor signal itself.

In this work, we propose a novel and robust evidential fusion framework that unifies the strengths of both contextual and conflict-guided discounting. Specifically, we estimate sensor reliability per class to discount mass functions semantically, and subsequently apply a conflict-guided discounting step to account for disagreement across modalities at the spatial level. This two-stage trust modeling mechanism allows our model to reason both about the expected utility of each sensor and its consistency with others, enabling robust and adaptive multimodal segmentation under uncertain or degraded conditions.

3 Evidence Theory Basics

Evidence theory, or Dempster-Shafer theory, is a formalism for representing, reasoning and making decision under uncertainty [19]. Two major uncertainty types can be distinguished: it is known to be aleatory when due to process randomness and can not be reduced. When uncertainty rises from a lack of knowledge, it is epistemic. This form of uncertainty can be limited by acquiring additional information.

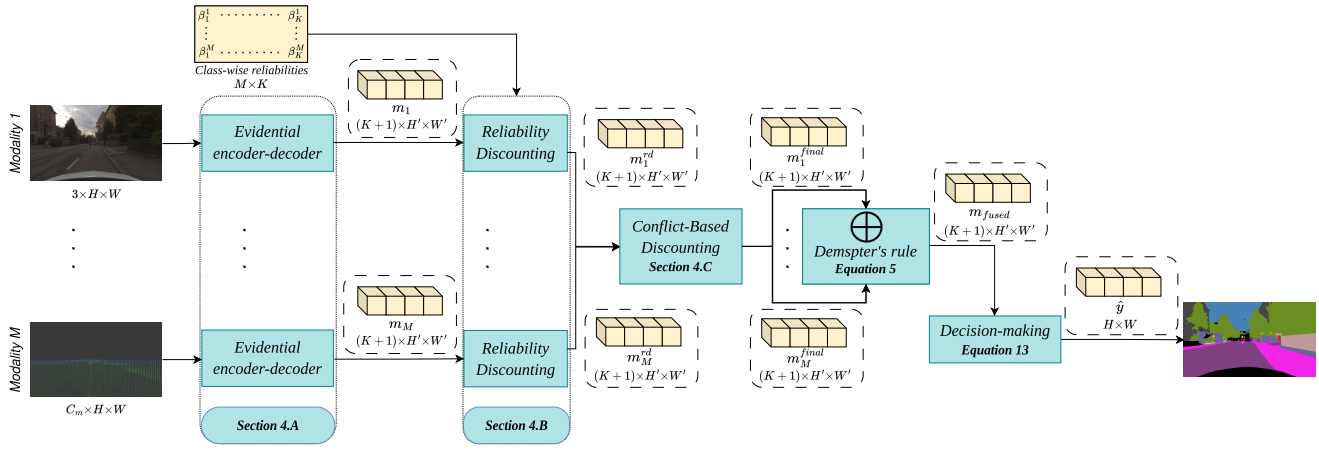


Figure 1: **ReCoLaF architecture.** Each modality is associated to an independent evidential encoder-decoder, which outputs mass functions. The mass functions of each modality are first discounted on the basis of their reliability and then on the basis of their respective conflict. The discounted mass functions are then fused and converted into probabilities to make a decision.

3.1 General definitions

Let $\Omega = \{\omega_1, \omega_2, \omega_3, ..., \omega_n\}$ be the *Frame of Discernment* (FoD). Ω is the finite set of mutually exclusive and exhaustive elements called *singletons*. Singletons are of single cardinality. A *Basic Belief Assignment* is a mass function $m : 2^\Omega \rightarrow [0, 1]$ that satifies the following constraints:

$$m(\emptyset) = 0 \tag{1}$$

$$\sum_{A \in 2^\Omega} m(A) = 1 \tag{2}$$

where 2^Ω is the power set of Ω defined as follows:

$$2^\Omega = \{\emptyset, \{\omega_1\}, \ldots, \{\omega_n\}, \{\omega_1, \omega_2\}, \{\omega_1, \omega_3\}, \ldots, \Omega\} \tag{3}$$

For clarity and coherence purpose, m will be called a mass function in the following sections. For any subset $A \in 2^\Omega$, $m(A)$ is bounded between 0 and 1. The quantity $m(A)$ measures the belief that one commits exactly to hypothesis A (i.e., the true answer to a certain question is in A), and it can not be assigned to any proper subset of A. If $m(A) > 0$, A is called a *focal set (or element)* of m.

3.2 Mass function discounting

A source of evidence may not be reliable or its associated support can be inaccurate. In this situation discounting the support given by the mass function is relevant [19]. Consider $1 - \alpha$, the discounting factor, i. e. α the degree of trust in the evidence with $0 < \alpha < 1$. The discounted mass function ${}^\alpha m(A)$ is given as:

$$\begin{aligned} {}^\alpha m(A) &= \alpha \cdot m(A) \quad \forall A \subset \Omega \\ {}^\alpha m(\Omega) &= (1 - \alpha) + \alpha \cdot m(\Omega) \end{aligned} \tag{4}$$

Discounting can be used to lower the effect of sources which are not fully trusted before evidence combination (see Section 3.3).

3.3 Evidence Combination and Conflict Management

Two mass functions m_1 and m_2 representing independent pieces of evidence (e.g., predictions from two different sensors) on a common frame Ω can be combined by Dempster-Shafer's rule (DS) [19] defined as:

$$(m_1 \oplus m_2)(A) = \frac{1}{1 - \kappa} \sum_{B \cap C = A} m_1(B) m_2(C) \tag{5}$$

for all $A \in 2^\Omega$, $A \neq \emptyset$, and $(m_1 \oplus m_2)(\emptyset) = 0$. The DS rule is commutative and associative. The constant k is called the conflict degree between the mass functions and is given as:

$$\kappa = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \tag{6}$$

If k=0, the mass functions m_1 and m_2 are non-conflicting (i.e., each focal set of m_1 intersects all focal sets of m_2). If k=1, the pieces of evidence are logically contradictory (i.e., total conflict) and their combination through the DS rule is impossible.

3.4 Decision in evidence theory

Once the sources of evidence are combined, a probabilistic decision rule is required to select the final class $\omega_k \in \Omega$. In this work, we adopt the DSmP transformation [6], which redistributes the uncertainty toward singletons while minimizing the entropy of the resulting probability distribution.
The DSmP transformation for a given singleton ω_k is defined as:

$$\text{DSmP}_\varepsilon(\omega_k) = \sum_{A \in 2^\Omega} m(A) \frac{\sum_{a \in \{\omega_k \cap A\}} m(a) + \varepsilon \cdot |\omega_k \cap A|}{\sum_{a \in A} m(a) + \varepsilon \cdot |A|} \tag{7}$$

The parameter $\varepsilon > 0$ moderates the impact of the focal elements cardinality in the uncertainty redistribution.

4 Reliability-Aware Fusion Framework

Figure 1 illustrates the overall architecture of the proposed adaptive evidential fusion framework. The model is composed of four main stages: modality-specific evidential encoder-decoders, sensor reliability estimation, conflict-based discounting, and evidential fusion through Dempster-Shafer theory. We detail each component of the architecture in the following subsections.

4.1 Evidential Encoder-Decoder (per modality)

Let $\mathcal{M}=\{1,\dots,M\}$ be the set of modalities and $\Omega=\{\omega_1,\dots,\omega_K\}$ the FoD for a K-class segmentation task. For each modality $m \in \mathcal{M}$, an encoder-decoder produces a mass function map $m_m^{\text{raw}}(i,j) \in \mathbb{R}^{K+1}$ for each pixel (i,j) by applying a softmax activation function to ensure that the mass functions sum to one.

Following prior work in evidential deep learning [20, 5], we restrict the set of focal elements to only the singletons and the full frame Ω. That is, for each class $\omega_k \in \Omega$, a mass value is assigned to $\{\omega_k\}$, and the remaining mass is assigned to Ω, which captures total uncertainty. This simplifies the mass function representation while retaining the expressiveness needed for uncertainty modeling.

4.2 Class-wise Sensor Reliability

We assign to each modality a vector of class-specific reliability scores:

$$\boldsymbol{\beta}_m = (\beta_1^m, \beta_2^m, \dots, \beta_K^m) \in [0,1]^K \tag{8}$$

Each β_k^m quantifies how reliable modality m is for predicting class ω_k. These scores are used to discount each of the raw mass functions per class, resulting in the discounted mass function m_m^{rd}:

$$m_m^{\text{rd}}(\{\omega_k\}) = \beta_k^m \cdot m_m^{\text{raw}}(\{\omega_k\}) \tag{9}$$

$$m_m^{\text{rd}}(\Omega) = 1 - \sum_{k=1}^{K} m_m^{\text{rd}}(\{\omega_k\}) \tag{10}$$

The reliability scores for each modality $\boldsymbol{\beta}_m$ are implemented as trainable parameters and are learned jointly with the rest of the network during training. Importantly, during inference time, they are fixed, not input-dependent. This design reflects a global semantic prior that captures the average reliability of each modality for each class.

4.3 Conflict-based Adjustment

To account for sensor disagreements, we apply a conflict-based discounting mechanism. Following the ECoLaF framework [5], we measure the level of disagreement between modalities using the Jousselme distance [12] and derive a conflict score Conf_m for each modality. This score is then converted into a conflict-based reliability score α_m [18].

The final mass function m_m^{final} is given by the discounting formula:

$$\begin{cases} m_m^{\text{final}}(\omega_k) & = \alpha_m \cdot m_m^{\text{rd}}(\omega_k), \qquad \omega_k \in \Omega \\ m_m^{\text{final}}(\Omega) & = 1 - \alpha_m + \alpha_m \cdot m_m^{\text{rd}}(\Omega) \end{cases} \tag{11}$$

This step adjusts the influence of each sensor locally based on how much it agrees with others.

4.4 Multimodal Fusion and Class Definition

The refined mass functions from all modalities are fused using Dempster's rule $\oplus$ (Eq. 5):

$$m_{\text{fused}}(A) = \left(\bigoplus_{m=1}^{M} m_m^{\text{final}} \right)(A) \tag{12}$$

For decision making, according to (7), we choose the class with highest DSmP as the final prediction:

$$\hat{y} = \arg\max_k \text{DSmP}_\varepsilon(\omega_k) \tag{13}$$

Following the recommendations in [6], we choose $\varepsilon = 0.001$ to obtain a probability function with an entropy as low as possible while avoiding numerical instabilities.

In our architecture of combining the model-based approach of Dempster-Shafer theory with deep learning, the reliability weighting is positioned ahead of the conflict discounting (see Fig. 1). In this arrangement, semantically relevant mass functions are obtained first by class-based rectification (i.e., class-specific reliability value β_k^m, (Eq. 8)). Then, they are discounted based on their inter-class disagreement. If the order were the opposite, the conflict among sources could be calculated based on less refined mass functions. Therefore, the proposed order of computations is chosen for maintaining information quality.

5 Experiments

5.1 Datasets and Implementation Details

A quantitative validation has been performed on two multimodal datasets designed for robust semantic segmentation under challenging driving conditions: the **DeLiVER** [24] synthetic dataset, and **MUSES** [2], a non-synthetic real-world dataset. Both datasets provide diverse driving scenarios and multimodal sensor data.

DeLiVER comprises paired images from four sensor modalities—RGB, Depth, Event, and LiDAR—captured under diverse weather conditions and simulated sensor failure scenarios, including motion blur, over-exposure, under-exposure, LiDAR jitter, and low-resolution Event data. It is designed to study the semantic segmentation of road scenes across 25 classes. The dataset includes 3983, 2005, and 1897 front-view image pairs for training, validation, and testing, respectively.

MUSES is a dataset dedicated to the analysis of real-world road scenes under adverse weather conditions, specifically fog, rain, and snow. It comprises 1500, 250, and 750 paired images across four modalities—RGB, Event, LiDAR, and RADAR—for training, validation, and testing,

RGB	Depth	Event	LiDAR	CMNeXt [24]	CAFuser [3]	ECoLaF [5]	ReCoLaF
✓				20.62	24.69	31.44	**35.62**
	✓			40.29	**43.52**	38.77	42.06
		✓		2.82	1.74	1.87	**3.77**
			✓	2.76	1.44	2.40	**3.13**
✓	✓			52.96	**54.05**	49.23	49.89
✓		✓		20.37	26.29	31.44	**35.69**
✓			✓	20.79	26.04	31.72	**35.55**
	✓	✓		40.46	**43.95**	38.77	42.48
	✓		✓	40.29	**43.52**	38.86	42.05
		✓	✓	2.81	1.58	2.03	**4.14**
✓	✓	✓		53.11	**54.44**	49.23	49.90
✓	✓		✓	52.88	**53.93**	49.25	49.90
✓		✓	✓	20.54	27.37	31.72	**35.66**
	✓	✓	✓	40.39	**43.78**	38.86	42.58
✓	✓	✓	✓	53.01	**53.87**	49.25	49.90
mean				30.94	33.35	32.35	**34.82**

(a) DeLiVER

RGB	Event	LiDAR	Radar	CMNeXt [24]	CAFuser [3]	ECoLaF [5]	ReCoLaF
✓				49.82	63.92	65.02	**66.56**
	✓			2.65	**4.51**	3.15	3.99
		✓		2.65	16.79	19.49	**33.90**
			✓	7.56	**8.34**	3.64	4.23
✓	✓			52.55	65.33	65.02	**66.56**
✓		✓		66.90	**68.57**	66.43	68.29
✓			✓	61.66	65.10	65.02	**66.56**
	✓	✓		2.62	16.83	19.49	**33.90**
	✓		✓	7.71	**8.39**	3.64	4.23
		✓	✓	9.94	15.18	20.20	**34.04**
✓	✓	✓		66.64	**68.76**	66.43	68.29
✓	✓		✓	62.09	65.26	65.02	**66.56**
✓		✓	✓	71.06	**72.88**	66.42	68.29
	✓	✓	✓	10.82	17.59	20.20	**34.04**
✓	✓	✓	✓	71.06	**72.88**	66.42	68.29
mean				36.38	42.02	41.04	**45.85**

(b) MUSES

Table 1: Performances comparison of using different modalities in mIoU(%). Each row represents a test-time inference scenario where a subset of modalities is available (✓) and the others are disabled (i.e., replaced with zero-filled tensors to simulate sensor failure). ✓ indicates the available modalities at test time, while others are disabled (zero-filled) to simulate sensor failures. Bold values represent the best performances to the nearest rounding.

respectively. As ground truth annotations are not available for the original test set, we re-split the training set into a new training and validation subset, and repurpose the original validation set as the new test set. This results in 1250, 250, and 250 images for training, validation, and testing, respectively, while keeping the same balance between the day, night, clear, fog, rain and snow images as the original dataset.

Implementation details. All experiments are performed on a A100 GPU. The models are trained with an initial learning rate of 6×10^{-5}. The optimizer is AdamW [17] with epsilon $1e^{-8}$ and weight-decay 0.01 over 200 epochs. For all experiments, the learning rates are scheduled with a polynomial strategy with power 0.9 including 10 warm-up epochs.

The data augmentation includes random horizontal flips, random scaled crops, gaussian blur and random color jitter. The proposed architecture ReCoLaF is built with Segformer [22] encoder-decoders with an MiT-B2 backbone [22]. All Dempster-Shafer-based modules are fully differentiable and integrated end-to-end with standard backpropagation. It is nevertheless noticeable that only the class-wise reliability estimation module contains trainable parameters.

5.2 Experimental Setup

We evaluate the robustness, efficiency, and interpretability of ReCoLaF under challenging multimodal perception conditions. We compare against recent state-of-the-art fusion methods, analyse the estimated reliability scores, and report model efficiency in terms of FLOPs, parameters, and inference time.For both DeLiVER and MUSES datasets, we adopt the protocol introduced by [14], supported by [15], and followed by [5], where all sensor modalities are used during training, and sensor failure scenarios are simulated at inference time.

To simulate sensor failures, we selectively disable one or more modalities during inference by replacing the corresponding input with zero-filled tensors. This strategy, shown to be simple, reproducible, and effective for evaluating fusion robustness in prior work [15], allows us to assess the model's robustness without requiring modality-specific dropout training. For each degraded configuration, we report the mean Intersection over Union (mIoU) [7, 16] over all semantic classes. A modality is considered "available" if its input is provided during inference. Otherwise, it is zero-filled. Results across all tested configurations are presented in Table 1.

5.3 Robustness Analysis

Results on DeLiVER. ReCoLaF outperforms CMNeXt and ECoLaF across most degraded configurations and achieves competitive performance compared to CAFuser. In full-modality settings (all four sensors), CAFuser achieves the highest mIoU (53.87%), followed closely by ReCoLaF (49.90%) and ECoLaF (49.25%). However, ReCoLaF shows stronger robustness in challenging configurations. For instance, in the *RGB + Event* configuration, ReCoLaF scores 35.69%, surpassing ECoLaF (31.44%), CAFuser (26.29%), and CMNeXt (20.37%). Overall, these results highlight the strong dependency of both CMNeXt and CAFuser to the Depth modality. This can be explained by the fact that this modality is never degraded during the training phase and is not realistically impacted by adversarial weather conditions due to the synthetic nature of the dataset. Therefore, the Depth modality is always very informative whereas the RGB modality can be impacted by over-exposure, under-exposure or motion blur during training, encouraging the models to strongly rely on the Depth modality. On average across all configurations, ReCoLaF achieves 34.82% mIoU, outperforming ECoLaF (32.35%), CMNeXt (30.94%), and CAFuser (33.35%).

Results on MUSES. ReCoLaF continues to demonstrate strong robustness against sensor failure under real-world adverse weather conditions. While CAFuser

obtains once more the highest performance in the full-modality setting (72.88%), it shows a strong dependency on the RGB modality, whereas CMNeXt interestingly shows a dependency on the *RGB+LiDAR* and the *RGB+Radar* combinations. Under partial modality configurations, ReCoLaF consistently surpasses other methods. For instance, in the *Event+LiDAR* setup, ReCoLaF achieves 33.90%, compared to 19.49% (ECoLaF), 16.83% (CAFuser), and 2.62% (CMNeXt). In the *RGB+Radar* configuration, ReCoLaF reaches 66.56%, slightly outperforming ECoLaF (65.02%), CAFuser (65.10%), and CMNeXt (61.66%). On average across all configurations, ReCoLaF scores 45.85% mIoU, compared to 41.04% (ECoLaF), 36.38% (CMNeXt), and 42.02% (CAFuser). Regarding the LiDAR modality, there is a clear difference between DeLIVER and MUSES in terms of performances. This may be explained by the fact that the real-world LiDAR images from the MUSES dataset are more dense than the synthetic ones from the DeLIVER dataset, making the classical transformer-based encoder-decoders more effective to extract information from these real-world images.

The presented results confirm that ReCoLaF provides a strong balance between accuracy in favorable conditions and robustness in degraded ones. While CAFuser performs well when all modalities are available, its performance drops more steeply under sensor failures, even though modality-drop was used during training. In contrast, ReCoLaF explicitly limits over-reliance on any single modality by modeling class-specific sensor reliability and incorporating conflict-guided fusion. This design may result in slightly lower peak performance in full-modality settings, but it leads to improved robustness and stability under adverse sensing conditions. Such a trade-off is an intentional design choice, aligned with the requirements of safety-critical autonomous driving systems, where resilience to sensor degradation and failures is often more critical than maximizing accuracy under ideal conditions.

5.4 Sensor Reliability Analysis

To better understand how our model estimates class-specific reliability across semantic classes, we analyze the learned per-modality, per-class reliability scores. Figure 2 presents a radar chart visualization of the estimated reliability scores for four modalities: RGB, Depth, LiDAR, and Event. We observe that Depth consistently shows the highest reliability for structural and planar classes such as *Building*, *Wall*, *Fence*, and *Vegetation*, reflecting its strong geometric representation. In contrast, RGB performs well on texture-rich and visually distinctive classes, such as *Road lines*, *Cars*, and *Traffic signs*. Event cameras show moderate reliability across all classes but do not dominate in any particular category. They tend to follow RGB in shape but with slightly lower scores, indicating their utility in spread but less impactful alone. LiDAR reflects lower reliability in several classes, especially *Traffic signs*, *Wall*, and *Sky*. This may be due to its sparse nature or limitations in vertical resolution for capturing elevated or fine-detailed features. These findings confirm that our class-specific reliability estimation module captures meaningful sensor-class relationships, effectively enabling the fusion framework to weigh sensor contributions adaptively based on semantic content.

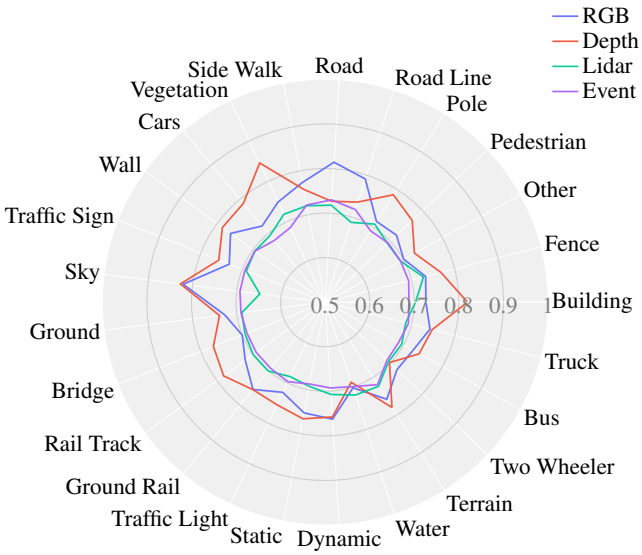


Figure 2: Estimated class-specific reliability across the four sensor modalities of the DeLIVER dataset: RGB, Depth, LiDAR, and Event.

5.5 Ablation study

To evaluate the effectiveness of our two-stage discounting mechanism, we conduct an ablation study comparing ReCoLaF with and without discounting on DeLiVER. Specifically, we remove both the semantic reliability estimation and the conflict-based adjustment, resulting in a version where all modalities contribute equally during fusion. To this end, we train the two models separately.
We compare the following two variants:

- **ReCoLaF w/o discounting:** all mass functions are fused without any reliability-based weighting or conflict-guided discounting. This is equivalent to applying uniform fusion in the Dempster-Shafer framework (i.e., $\beta_k^m = 1$ and $\alpha_m = 1$ in Equations (9) and (11), respectively).
- **ReCoLaF (full):** includes both class-specific reliability estimation and conflict-guided discounting.

As shown in Table 2, removing both discounting stages results in a performance drop of over 6 points in average mIoU. The ablation of the discounting mechanism makes the model highly dependent on the Depth modality, leading to the same lack of robustness as the probabilistic models CMNeXt and CAFuser. This highlights the importance of jointly modeling sensor reliability and inter-modality inconsistency during fusion, particularly under sensor failure conditions.

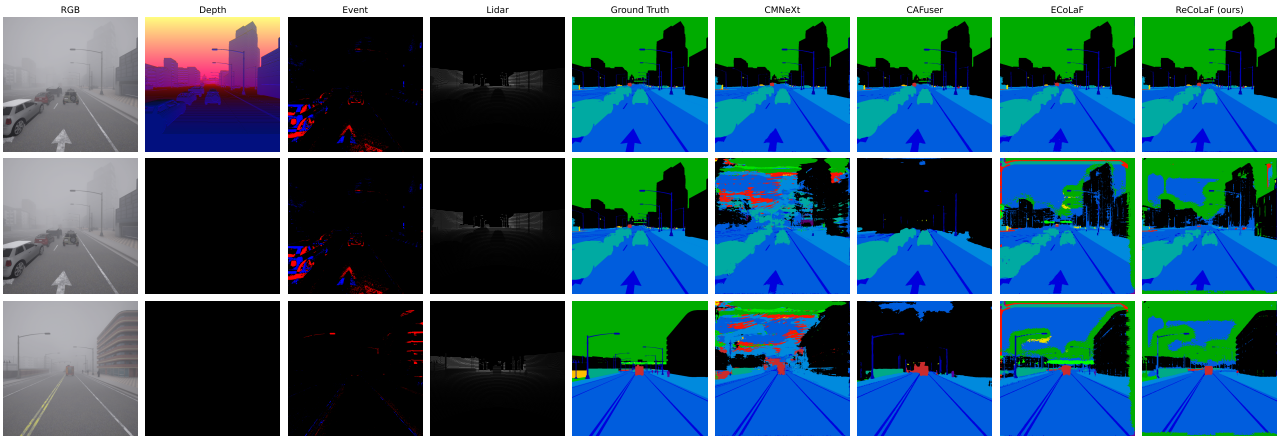


Figure 3: Qualitative segmentation results under degraded conditions (fog and missing depth). Row 1 shows the full-modality scene (RGB, Depth, Event, LiDAR), Row 2 is the same scene under simulated Depth failure (zero-filled), and Row 3 shows a different scene also with missing Depth. From left to right: sensor inputs, ground truth, and predictions from CMNeXt, CAFuser, ECoLaF, and ReCoLaF. ReCoLaF consistently provides cleaner and more accurate segmentations, especially in the presence of sensor failure.

RGB	*Depth*	*Event*	*LiDAR*	ReCoLaF w/o discounting	Full ReCoLaF
✓				12.67	**35.62**
	✓			41.70	**42.06**
		✓		1.99	**3.77**
			✓	1.99	**3.13**
✓	✓			49.81	**49.89**
✓		✓		12.67	**35.69**
✓			✓	12.67	**35.55**
	✓	✓		41.70	**42.48**
	✓		✓	41.69	**42.05**
		✓	✓	1.99	**4.14**
✓	✓	✓		49.81	**49.90**
✓	✓		✓	49.81	**49.90**
✓		✓	✓	12.67	**35.66**
	✓	✓	✓	41.69	**42.58**
✓	✓	✓	✓	49.81	**49.90**
mean				28.18	**34.82**

Table 2: Ablation study on DeLiVER: effect of discounting.

5.6 Qualitative Analysis

To further assess the robustness of the proposed framework under degraded sensing conditions, Figure 3 presents qualitative segmentation results from the DeLiVER dataset under fog and simulated Depth modality failure. The first row shows a scene where all four modalities available. In this full-modality configuration, all methods produce reasonably accurate segmentation maps that are sufficiently clean for perception in autonomous systems. The second and third rows simulate sensor failure by removing the Depth input, which is the modality the models are most dependent on. In both cases, CMNeXt produces highly degraded predictions with strong visual artifacts. CAFuser shows some resilience but provides substantial misclassifications across large areas of the image, particularly in the upper regions of the image, where it often confuses the sky with buildings. ECoLaF partially mitigates the issue thanks to its conflict-guided fusion, but still shows significant artifacts and inconsistencies. ReCoLaF, in contrast, maintains structurally consistent predictions and accurate labeling even in the absence of Depth. This illustrates its ability to effectively adjust the contribution of the remaining modalities based on their reliability and mutual agreement. The results confirm the advantage of our two-stage fusion strategy, which combines learned semantic reliability scores with conflict-guided fusion which provide strong resilience to sensor-level uncertainty and failure.

Overall, these qualitative results support our claim that ReCoLaF is more resilient to sensor failures than attention-based (CMNeXt) or condition-aware (CAFuser) fusion approaches, making it a promising choice for deployment in real-world autonomous driving systems.

5.7 Real-Time Inference and Model Efficiency

Table 3 reports FLOPs, parameters count, and inference time for all evaluated models on DeLiVER. ReCoLaF shares the same architectural complexity as ECoLaF, with nearly equivalent computational cost and parameter count, and is significantly more frugal than CAFuser (which requires condition prediction). CMNeXt is the most lightweight, but achieves lower average mIoU under sensor failures. These results demonstrate that ReCoLaF offers a favorable trade-off between robustness and computational cost, making it suitable for real-time autonomous driving applications. Moreover, the number of parameters in the ReCoLaF architecture can be greatly reduced by adopting a shared backbone strategy as in CAFuser.

	CMNeXt	CAFuser	ECoLaF	ReCoLaF
GFLOPs	65.42	699.12	157.12	157.63
# Params (M)	58.73	75.01	103.16	103.16
Inference time (s/img)	0.17	0.38	0.23	0.24

Table 3: FLOPs, parameters and inference time comparison on the DeLIVER dataset.

6 Conclusion

In this paper, we presented ReCoLaF, a novel fusion framework for robust multimodal semantic segmentation in autonomous driving. ReCoLaF combines class-specific sensor reliability estimation with conflict-guided adjustment to adaptively fuse heterogenous sensor modalities under uncertain or degraded conditions. Grounded in Dempster-Shafer theory, our two-stage fusion strategy first discounts each modality based on learned per-class semantic reliability, then further adjusts its contribution based on disagreement with other modalities at the pixel level.

Experiments on the DeLiVER and MUSES datasets demonstrate that ReCoLaF consistently improves robustness compared to strong baselines, including middle fusion approaches with cross-attention mechanism and conflict-only evidential methods, particularly under sensor degradation and failure scenarios. While some competing methods achieve higher performance when all modalities are available, ReCoLaF is intentionally designed to favor robustness and stability under adverse conditions, reflecting a trade-off that is well aligned with the requirements of safety-critical autnomous driving systems. These results highlight the importance of explicitly modeling both the reliability and agreement of sensor modalities to achieve robust scene understanding in autonomous driving systems.

In the current formulation, class-wise sensor reliability scores are learned as global parameters and remain fixed during inference. This design choice provides stable semantic priors while keeping the fusion process tractable and interpretable, and is complemented by a spatially adaptive conflict-based mechanism that locally adjusts modality contributions. Future work will investigate dynamic and pixel-level reliability estimation to better capture region-specific and context-dependent sensor degradation. In addition, parameter-efficient architectures, such as shared backbone designs, will be explored to improve scalability to larger numbers of modalities and higher-resolution inputs, as well as to enhance real-time performance for practical autonomous driving applications.

acknowledgement

The authors are supported by the French National Research Agency (ANR) under grants HAISCoDe, INARI, and EviDeep. This project was provided with computing AI and storage resources by GENCI at IDRIS thanks to the grant 2024-AD011014391 on the supercomputer Jean Zay's A100 partition along with computing resources of CRIANN (Normandy, France).

References

[1] Alireza Asvadi, Luis Garrote, Cristiano Premebida, Paulo Peixoto, and Urbano J Nunes. Multimodal vehicle detection: fusing 3d-lidar and color camera data. *Pattern Recognition Letters*, 115:20–29, 2018.

[2] Tim Brödermann, David Bruggemann, Christos Sakaridis, Kevin Ta, Odysseas Liagouris, Jason Corkill, and Luc Van Gool. Muses: The multi-sensor semantic perception dataset for driving under uncertainty. In *European Conference on Computer Vision*, pages 21–38. Springer, 2024.

[3] Tim Brödermann, Christos Sakaridis, Yuqian Fu, and Luc Van Gool. Cafuser: Condition-aware multimodal fusion for robust semantic perception of driving scenes. *IEEE Robotics and Automation Letters*, 10(4):3134–3141, 2025.

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[5] Lucas Deregnaucourt, Hind Laghmara, Alexis Lechervy, and Samia Ainouz. A conflict-guided evidential multimodal fusion for semantic segmentation. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 1373–1382, February 2025.

[6] Jean Dezert and Florentin Smarandache. A new probabilistic transformation of belief mass assignment. *CoRR*, abs/0807.3669, 2008.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[8] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[9] Jannik Fritsch, Tobias Kühnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1693–1700, 2013.

[10] Mihreteab Negash Geletu, Dănuţ-Vasile Giurgi, Thomas Josso-Laurain, Maxime Devanne, Mengesha Mamo Wogari, and Jean-Philippe Lauffenburger. Evidential deep learning-based multi-modal environment perception for intelligent vehicles. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2023.

[11] Ling Huang, Su Ruan, Pierre Decazes, and Thierry Denœux. Deep evidential fusion with uncertainty quantification and reliability learning for multimodal medical image segmentation. *Information Fusion*, 113:102648, 2025.

[12] Anne-Laure Jousselme, Dominic Grenier, and Éloi Bossé. A new distance between two bodies of evidence. *Information fusion*, 2(2):91–101, 2001.

[13] Bingyu Li, Da Zhang, Zhiyuan Zhao, Junyu Gao, and Xuelong Li. Stitchfusion: Weaving any visual modalities to enhance multimodal semantic segmentation, 2024.

[14] Yupeng Liang, Ryosuke Wakaki, Shohei Nobuhara, and Ko Nishino. Multimodal material segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19800–19808, 2022.

[15] Chenfei Liao, Kaiyu Lei, Xu Zheng, Junha Moon, Zhixiong Wang, Yixuan Wang, Danda Pani Paudel, Luc Van Gool, and Xuming Hu. Benchmarking multimodal semantic segmentation under sensor failures: Missing and noisy modality robustness, 2025.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[18] Arnaud Martin, Anne-Laure Jousselme, and Christophe Osswald. Conflict measure for the discounting operation on belief functions. In *2008 11th International Conference on Information Fusion*, pages 1–8, 2008.

[19] Glenn Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.

[20] Zheng Tong, Philippe Xu, and Thierry Denœux. Fusion of evidential cnn classifiers for image classification. In *International Conference on Belief Functions*, pages 168–176. Springer, 2021.

[21] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, 128(5):1239–1285, 2020.

[22] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[23] Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *CVPR*, 2023.

[24] Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1136–1147, 2023.

Session 5 : Détection d'anomalies & Incertitude

Temporal Conditional Normalizing Flows for Data Augmentation in Remaining Useful Life Prediction under Data Scarcity

Guillaume Prevost[1], Esteban Cabanillas[1], Jérôme Boutet[1], Cornel Ioana[2]

[1] Univ. Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France

[2] Gipsa-Lab, Univ. Grenoble-Alpes, Saint-Martin d'Hères, France

guillaume.prevost@cea.fr

Résumé

La prédiction de la durée de vie résiduelle (RUL) des roulements souffre d'un manque de données industrielles, limitant la généralisation des modèles supervisés. Nous proposons un modèle TCNF (Temporal Conditional Normalizing Flow), un modèle génératif conditionné sur la RUL et un contexte temporel encodé par GRU, capable de générer des trajectoires de dégradation complètes et cohérentes. Une couche de normalisation affine apprise est intégrée directement dans le flot, garantissant bijectivité et calcul exact du log-déterminant. Sur le jeu de données XJTU-SY, TCNF améliore le RMSE par rapport à la référence sans augmentation et surpasse les approches comparatives GAN et VAE.

Mots-clés

Normalizing flows, augmentation de données, durée de vie résiduelle.

Abstract

Predicting the remaining useful life (RUL) of bearings suffers from a lack of industrial data, limiting the generalization of supervised models. We propose a temporal conditional normalizing flow model (TCNF), a generative model conditioned on RUL and a temporal context encoded by GRU, capable of generating complete and consistent degradation trajectories. An affine normalization layer is integrated directly into the flow, ensuring bijectivity and exact log-determinant computation. On the XJTU-SY dataset, TCNF improves the RMSE compared to the baseline without augmentation and outperforms the comparative GAN and VAE approaches.

Keywords

Normalizing flows, data augmentation, remaining useful life.

1 Introduction

The predictive maintenance (PdM) field has become a major strategic lever for industry, enabling reduced maintenance costs, improved equipment availability and prevention of critical failures, with significant economic gains [1][2]. At the core of PdM, Remaining Useful Life (RUL) estimation aims to predict the time remaining before system failure from sensor measurements such as vibrations, currents or temperatures. In rotating mechanical systems, and in particular bearings, vibration signals are widely exploited due to their high sensitivity to degradation mechanisms [3] [4].

Despite recent advances in deep learning methods for RUL estimation [5][6], their industrial deployment remains severely limited by the scarcity of relevant data. Run-to-failure data, required to model the complete evolution of degradation up to failure, are costly to acquire, time-consuming to collect, and rarely available in large quantities [7]. This constraint is even more pronounced in embedded or real industrial settings, where equipment is designed precisely to avoid failure situations. As a result, datasets used for RUL model training are typically small, imbalanced and heterogeneous, leading to overfitting and limited generalization.

To address this data scarcity, several recent works have explored manual data augmentation techniques [8][9], or generative model-based approaches such as Generative Adversarial Networks (GAN) [10][11] or Variational Autoencoders (VAE) [12][13]. Although these approaches have shown promising potential for enriching training sets, they exhibit important limitations in the context of predictive maintenance. GANs frequently suffer from training instability and mode collapse, while VAEs rely on likelihood approximations that can degrade the statistical fidelity of generated data. Moreover, most of these approaches generate samples independently, without guaranteeing the temporal coherence of degradation trajectories or explicit conditioning on health state, limiting their application to fault classification [14][15].

Yet in RUL applications, system degradation is inherently a temporal and progressive phenomenon. Many works have shown that RUL can be modelled according to a two-phase dynamics, comprising an initial healthy operating phase followed by an accelerated degradation phase leading to failure [16]. Generating realistic synthetic data should therefore respect not only the marginal distributions of measured features, but also their temporal evolution conditioned on RUL. Few existing works explicitly address this problem within a rigorous probabilistic framework.

In this context, normalizing flow models [17][18] appear as a particularly well-suited alternative for data generation in predictive maintenance. These generative models rely on bijective and differentiable transformations that enable exact probability density estimation through likelihood maximization. Unlike GANs and VAEs, normalizing flows offer increased training stability, clear probabilistic interpretability, and coherent sample generation through model inversion. However, their application to the generation of temporally conditioned degradation trajectories remains largely unexplored in the literature.

In this paper, we propose a novel data augmentation approach for RUL estimation based on a Temporal Conditional Normalizing Flow, specifically designed for predictive maintenance scenarios under limited data constraints. The proposed model incorporates explicit RUL conditioning as well as temporal context encoding through a recurrent network, enabling the generation of complete, temporally coherent degradation trajectories. To guarantee model bijectivity and compatibility with normalizing flows, a learned, invertible affine normalization layer is introduced as a replacement for conventional normalization methods. The generated synthetic data are then used to enrich the training set of a RUL prediction model.

The main contributions of this paper are as follows:

— the proposal of a RUL-conditioned Normalizing Flow, capable of modelling the distribution of vibration features throughout the entire lifetime of a bearing,

— the integration of a recurrent temporal context enabling the generation of coherent and realistic degradation trajectories,

— the introduction of a learned, bijective and invertible affine normalization layer, specifically adapted to the constraints of flow-based models.

The remainder of this paper is organized as follows. Section 2 describes the proposed approach in detail. Section 3 presents the experimental protocol, while Sections 4 and 5 discuss the obtained results. Finally, Section 6 concludes the paper and outlines future research directions.

2 Proposed methodology

2.1 Overview of the framework

Our approach relies on a three-stage pipeline. In the first stage, d features are extracted from raw vibration signals and concatenated into a feature vector at each time step. These vectors, from the start of system life to the end of life, form degradation trajectories on which the generative model is trained. In the second stage, the model is used to generate new complete synthetic trajectories, which are added to the training set of a RUL prediction model. Fig 1. Shows the overall pipeline of the proposed approach. A RUL prediction regressive model is then trained on the augmented set.

The objective of the generative model is to learn the conditional distribution of the feature vectors observed at each time step, given the current RUL value and the recent observation history. Formally, we seek to model:

$$p(x_t | y_t, x_{\{t-1:t-k\}})$$

where $x_t \in \mathbb{R}^d$ is the feature vector at time t, $y_t \in [0,1]$ is the normalized RUL, and $x_{\{t-k:t-1\}}$ represents the k previous observations. A complete degradation trajectory is then generated by sequentially sampling from this distribution, from $y = 1.0$ down to $y = 0.0$. This work proposes to use a normalizing flow model conditioned on RUL and temporal context to generate complete degradation trajectories.

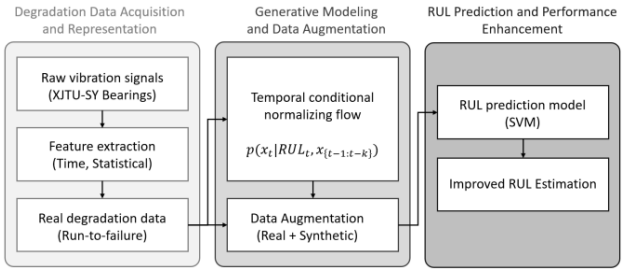


Figure 1. Overall pipeline of the proposed method. The pipeline consists of three stages: (1) degradation data acquisition and representation, (2) generative modelling and data augmentation, (3) RUL prediction.

2.2 Extracted features

The features extracted from vibration signals are chosen to capture different aspects of degradation. We retain classical time-domain descriptors from the literature: root mean square (RMS), zero-crossing rate (ZCR), crest factor, skewness, peak-to-peak amplitude, mean and kurtosis. These descriptors are widely used in the bearing health monitoring literature and enable the capture of phenomena of different natures: overall signal energy, impulsiveness, and non-stationarities [19].

2.3 RUL modelisation

In line with common practice in the predictive bearing maintenance literature, RUL is modelled as a two-phase function: a healthy phase followed by a degradation phase [16]. This model, often referred to as a piecewise linear RUL model [20], provides a more accurate representation of the physical dynamics of degradation by explicitly distinguishing the nominal operating phase from the active degradation phase [21].

Formally, for a bearing with total lifetime T, the normalized RUL at time t is defined by:

$$y_t = \begin{cases} 1, & t < t_{FPT} \\ 1 - \dfrac{t - t_{FPT}}{T - t_{FPT}}, & t \geq t_{FPT} \end{cases} \quad (1)$$

where t_{FPT} (First Predictable Time) denotes the instant from which degradation becomes measurable. This formulation has the advantage of limiting the influence of noise during the healthy phase and of introducing structural constraints (monotonicity and boundedness between 0 and 1) consistent with the physics of the system [22]. In this study, the t_{FPT} values are set in accordance with the proposals of Yin et al. [23].

2.4 Temporal Conditional Normalizing Flow

2.4.1 General structure

The Temporal Conditional Normalizing Flow (TCNF) consists of a succession of conditioned affine coupling layers, preceded by the learned normalization layer. The core idea of normalizing flows is to learn a bijective and differentiable mapping between a complex data distribution and a simple base distribution, here a standard Gaussian, such that both exact sampling and exact likelihood evaluation are tractable. The model transforms a data vector x into a latent vector z by:

$$z = f_K \circ f_{K-1} \circ \dots \circ f_1 \circ Norm(x)$$

where each f_i is an affine coupling layer. The exact log-likelihood is computed via the change-of-variables theorem:

$$log\ p(x) = log\ p_z(z) + \sum_{i=1}^{K} log\ |det\ J_{f_i}| + log\ |det\ J_{Norm}| \quad (2)$$

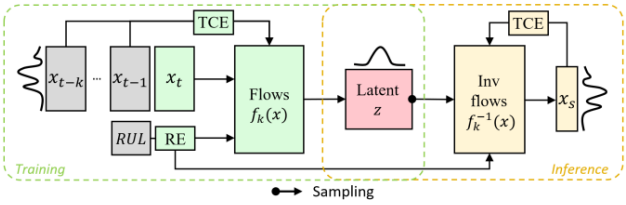


Figure 2. Architecture of the TCNF during training and inference phases. The TCE (Temporal Context Encoder) encodes the sliding history $x_{t-k}, \dots, x_t$ via a gated recurrent unit (GRU), while the RUL Encoder (RE) projects the current RUL value into an embedding space.

2.4.2 Learned affine normalization

Adequate data preprocessing is crucial for training normalizing flows. A traditional normalization scheme has the drawback of being a transformation external to the model, non-differentiable with respect to its parameters, and whose statistics must be computed and stored separately. We instead propose a learned affine normalization layer, integrated directly as the first layer of the flow.

Let $x \in \mathbb{R}^d$ be an input vector. The transformation is defined by:

$$z = \frac{(x - \mu)}{\sigma} \quad (3)$$

where μ and $\sigma = exp(log\_\sigma)$ are learnable parameters initialized from the empirical statistics of the training set. The constraint $\sigma > 0$ is guaranteed by the log_σ parameterization. The associated log-Jacobian determinant is:

$$log\ |det\ J| = -\sum_{i=1}^{d} \log\ (\sigma_i) \quad (4)$$

This formulation offers several decisive advantages over standard normalization: bijectivity is guaranteed by construction, the behavior is identical during training and inference, and the log-determinant is computable explicitly and exactly, which is a fundamental requirement of normalizing flows. Furthermore, the parameters are initialized from training set statistics and can be refined through backpropagation during training. Note that a version without log-determinant computation is also available to normalize the history before passing it to the temporal encoder, avoiding an unnecessary computation in this context.

2.4.3 Affine coupling layer

Each coupling layer splits the input vector into two parts via a binary mask that alternates between layers. The masked part x_1 is passed unchanged, while the unmasked part x_2 is transformed by an affine transformation whose parameters depend on both x_1 and the condition c:

$$x_2' = x_2 \times exp\big(s(x_1, c)\big) + t(x_1, c) \quad (5)$$

where s and t are neural networks with two hidden layers and ReLU activations, and c is the condition vector integrating the RUL embedding and the temporal context. The inverse transformation is trivial and the log-determinant equals the sum over the unmasked dimensions of $s(x_1, c)$ guaranteeing computation in linear time.

2.4.4 RUL Encoding

The scalar RUL value $y_t \in [0, 1]$ is projected into a higher-dimensional space. This allows the model to learn a non-linear representation of the RUL condition, more expressive than a simple concatenation of the raw scalar value.

2.4.5 Temporal context encoder (TCE)

To capture degradation dynamics and ensure the temporal coherence of generated trajectories, we introduce a Temporal Context Encoder (TCE) based on a GRU network. At each time step t during generation, the k most recent observations are normalized (via the learned affine normalization without log-det) and passed to the GRU, whose final hidden state is projected into a context vector $c_t \in \mathbb{R}^{d_{ctx}}$:

$$c_t = MLP(GRU(Norm(x_{t-k}, \dots, x_{t-1})))$$

This context is concatenated with the encoded RUL vector to form the complete condition for each coupling layer. In the absence of history (beginning of a trajectory), the context is set to zero, allowing the model to generate a coherent first point from the learned distribution.

2.5 Training

The model is trained by maximizing the exact log-likelihood (or equivalently minimizing the NLL). The training objective writes:

$$\mathcal{L} = -\left(\frac{1}{N}\right)\sum_{t=1}^{N} log\ p(x_t\ |y_t, x_{\{t-k:t-1\}}) \quad (6)$$

Temporal sequences are constructed by extracting sliding windows of length k over each bearing trajectory. To strengthen model robustness and enable trajectory generation in the absence of temporal context (e.g. at the beginning of a trajectory), a fraction of training examples is processed without temporal context (context set to zero). In our experiments, this ratio is set to 0.2.

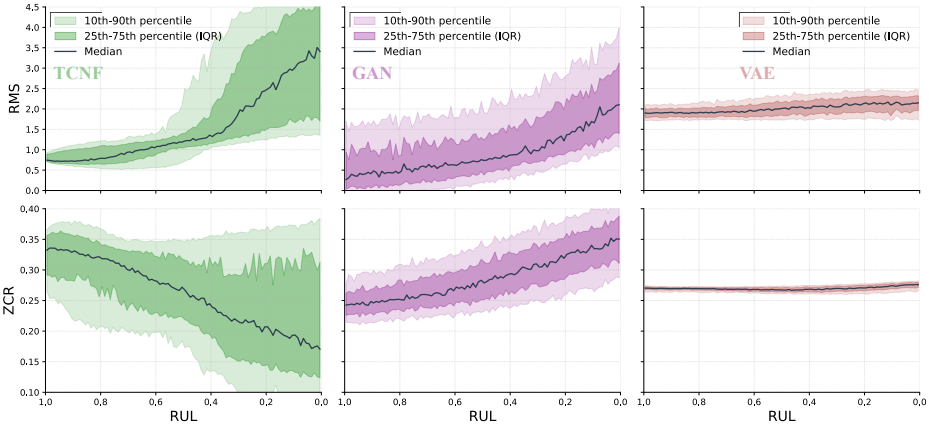


Figure 4. Mean generated trajectories for RMS and ZCR by TCNF (green), GAN (purple) and VAE (brown). Solid curves represent the median over 200 generated trajectories. Shaded areas indicate the 25–75% and 10–90% intervals.

3 Experimental validation

3.1 XJTU-SY dataset

The method is validated on the XJTU-SY experimental dataset, developed by Xi'an Jiaotong University in collaboration with Changxing Sumyoung Technology. This dataset provides 15 run-to-failure bearing experiments under three different load and rotation speed conditions (5 experiments per condition), capturing a wide diversity of lifetimes and fault types (inner race, outer race, cage and ball defects).

Vibration signals are sampled at 25.6 kHz, with 32,768 samples acquired every minute. A 5-fold cross-validation protocol is employed to assess model robustness and generalization ability. At each iteration, bearings are split into training and test sets with no overlap. Hyperparameters are kept identical across all folds to ensure fair comparison.

3.2 Comparative approaches

We compare TCNF against two generative baselines representative of the state of the art in data augmentation: a Generative Adversarial Network (GAN) and a Variational Autoencoder (VAE). Both models are equally conditioned on RUL and trained on the same dataset. The configuration without augmentation (training on real data only) serves as the absolute reference (Baseline).

4 Synthetic data quality

The quality of synthetic data generated by TCNF is assessed along two complementary axes: (i) analysis of the statistical consistency of conditional distributions, and (ii) analysis of the temporal coherence of complete generated trajectories.

4.1 Conditional distribution analysis

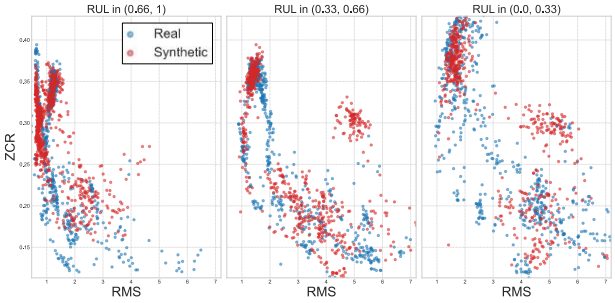


Figure 3. Joint RMS–ZCR distribution for three RUL intervals (high, intermediate, low). Blue points correspond to real training data, red points to synthetic data generated by TCNF.

Fig. 3 illustrates the joint distribution of the RMS and ZCR features for three RUL intervals (high, medium and low). Real data are shown in blue, while synthetic data generated by TCNF are shown in red.

The model faithfully reproduces the multivariate and non-linear structure of the real distribution. To quantify this agreement, we compute the Maximum Mean Discrepancy (MMD) between real and synthetic distributions within each RUL bin, averaged over 10 repetitions to account for generation stochasticity. TCNF achieves a mean MMD of 0.15 (± 0.06), outperforming both GAN (0.22 ± 0.01) and VAE (0.42 ± 0.01). The higher variance of TCNF's MMD across repetitions reflects its broader generative diversity, consistent with the wider confidence intervals observed in trajectory analysis. As RUL decreases, the distribution spreads and deforms, reflecting the increase in vibration variability and the emergence of impulsive behaviors typical of bearing degradation.

TCNF correctly captures:

- the progressive drift of distribution centroids,
- the increase in dispersion towards end of life,
- non-Gaussian structures and inter-feature correlations.

This ability to model a complex distribution conditioned on time confirms the benefit of joint conditioning on both RUL and temporal context.

4.2 Temporal coherence of generated trajectories

Fig. 4 compares the synthetic trajectories generated by TCNF, GAN and VAE. For each model, 200 complete trajectories are generated, and the median along with interquartile (25–75%) and extended (10–90%) intervals are shown for the RMS and ZCR features.

TCNF is distinguished by:

- realistic non-linear dynamics,
- a relatively stable healthy phase and a progressive transition towards an accelerated degradation phase,
- and increasing variability towards end of life.

Conversely, GAN produces smoother and more monotonic trajectories. Inter-trajectory variability is lower and the transition between healthy and degraded phases appears less coherent. VAE generates trajectories close to zero, with strongly reduced variability.

These observations confirm that the integration of the Temporal Context Encoder (TCE) enables TCNF to capture realistic degradation dynamics while preserving inter-trajectory diversity.

4.3 Contitioned extrapolation

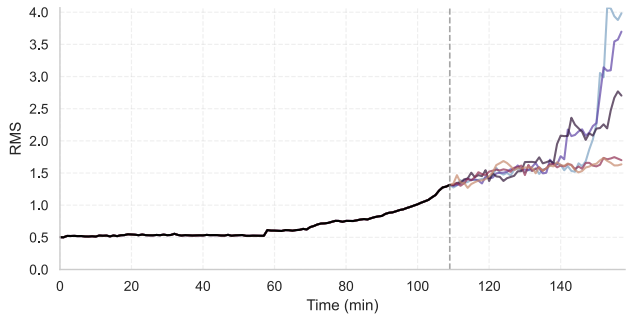


Figure 5. Example of conditioned extrapolation. The left part corresponds to a real partial trajectory (RMS). Colored curves represent five synthetic continuations generated by TCNF.

An important advantage of the proposed model lies in the ability to initialize the TCE from a real partial history. The model can thereby extrapolate an existing trajectory by generating its continuation conditioned on the current state.

Figure 5 presents an extrapolation example from a real trajectory (black line, left part). Five synthetic continuations are generated. It can be observed that:

- the extrapolated trajectories remain consistent with the initial dynamics,
- dispersion naturally increases during the degradation phase,
- and some trajectories exhibit a marked acceleration, reflecting the intrinsic uncertainty of end-of-life behavior.

This capability opens the way to targeted augmentation strategies, in particular for enriching data in the terminal phase (low RUL), which is often under-represented in industrial datasets.

5 Impact on RUL prediction

To evaluate the contribution of synthetic data, we train two regressive models to predict RUL with and without synthetic data.

Support Vector Machine (SVM). The SVM is widely used in the literature for RUL prediction and fault classification [24][25]. The input vector is augmented with an Exponentially Weighted Moving Average (EWMA) at two temporal scales, informing the model about the trend of each feature's evolution at different time scales.

Recurrent Neural Network (RNN). A recurrent neural network using a gated recurrent unit GRU architecture, with greater non-linear modelling capacity than the SVM, and also widely used in predictive maintenance for time series analysis [26][27].

Performance is evaluated according to three complementary criteria. RMSE measures the overall quadratic error:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{y}_t - y_t)^2} \tag{7}$$

Mean absolute error (MAE) measures the error in absolute value:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|\hat{y}_t - y_t| \tag{8}$$

The NASA score asymmetrically penalizes early predictions (underestimation of remaining RUL, risk of unwarranted maintenance) and late predictions (overestimation, risk of unanticipated failure), the latter being more critical in an industrial context:

$$Score = \sum_{t=1}^{T}\begin{cases}\exp\left(-\frac{\hat{y}_t - y_t}{13}\right) - 1, & \hat{y}_t < y_t \\ \exp\left(\frac{\hat{y}_t - y_t}{10}\right) - 1, & \hat{y}_t \geq y_t\end{cases} \tag{9}$$

To limit the impact of the stochastic nature inherent to generation, each experiment is repeated 10 times and reported results correspond to the means over these 10 repetitions. In each configuration, 7 synthetic trajectories are generated (corresponding to approximately 70% additional synthetic data). Table 1 presents the RUL prediction results for the four evaluated configurations.

Table 1. SVM RUL prediction results.

	RMSE	MAE	Nasa-score
Baseline	0.222	0.172	0.015
GAN	0.219	0.168	0.015
VAE	0.232	0.188	0.016
TCNF	**0.205**	**0.157**	**0.014**

Table 2. RNN RUL prediction results.

	RMSE	MAE	Nasa-score
Baseline	0.179	0.124	0.022
GAN	0.167	0.115	0.022
VAE	0.172	0.115	0.024
TCNF	**0.159**	**0.099**	**0.020**

TCNF improves all metrics for both prediction models. For the SVM, RMSE decreases by 7.7%, MAE by 8.7% and the NASA score by 6.7%. Gains are even more pronounced for the RNN, with RMSE dropping from 0.179 to 0.159 (−11.2%). These improvements, consistent across all cross-validation folds, indicate that synthetic trajectories bring genuinely useful information, regardless of the capacity of the downstream model.

The comparison between baselines is instructive. GAN provides modest but systematic gains on both models. VAE, on the other hand, slightly improves RNN performance but degrades SVM performance relative to the no-augmentation baseline, suggesting that its variational regularization produces trajectories insufficiently representative of the real variability of degradation.

The fact that TCNF is the only model to consistently improve the NASA score — the most demanding metric due to its asymmetric nature — is relevant: it suggests that the temporal coherence ensured by the TCE leads to better representation of end-of-life dynamics, where prediction errors are most costly from an industrial standpoint.

A further advantage of the approach lies in the ability to generate trajectories specifically targeting the degradation phase ($RUL < 1$), allowing rebalancing of the natural imbalance between the healthy phase (often the majority) and the degraded phase. By setting a short t_{FPT} during generation, it is possible to produce synthetic trajectories with an extended degradation phase, particularly enriching the representation of the most critical health states.

6 Conclusion

This paper introduced the Temporal Conditional Normalizing Flow (TCNF), a generative approach dedicated to data augmentation for remaining useful life prediction in industrial data scarcity settings. By combining a conditional normalizing flow, explicit RUL encoding and a GRU-modelled temporal context, the model generates complete, temporally coherent and statistically faithful degradation trajectories.

Results obtained on the XJTU-SY dataset show that synthetic data produced by TCNF systematically improve prediction performance (RMSE, MAE and NASA score), in contrast to the GAN and VAE-based comparative approaches. These gains confirm the value of exact probabilistic modelling and explicit temporal conditioning for degradation data generation.

Beyond quantitative performance, the model enables targeted generation of critical degradation phases and conditioned extrapolation of partial trajectories, opening promising avenues for more refined augmentation strategies in predictive maintenance.

This work highlights the potential of temporal conditional normalizing flows as a robust and relevant tool for improving the generalization of RUL prediction models in constrained industrial settings.

7 References

[1] J. Guo, Z. Li, and M. Li, ‘A Review on Prognostics Methods for Engineering Systems’, *IEEE Trans. Rel.*, vol. 69, no. 3, pp. 1110–1129, Sep. 2020, doi: 10.1109/TR.2019.2957965.

[2] M. Haarman, M. Mulders, and C. Vassiliadis, ‘Predictive maintenance 4.0: predict the unpredictable.’, *PwC and Mainnovation*, *4*, 2017.

[3] A. Althubaiti, F. Elasha, and J. A. Teixeira, ‘Fault diagnosis and health management of bearings in rotating equipment based on vibration analysis – a review’, *J. vibroeng.*, vol. 24, no. 1, pp. 46–74, Feb. 2022, doi: 10.21595/jve.2021.22100.

[4] D. Jung, Z. Zhang, and M. Winslett, ‘Vibration Analysis for IoT Enabled Predictive Maintenance’, in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, San Diego, CA, USA: IEEE, Apr. 2017, pp. 1271–1282. doi: 10.1109/ICDE.2017.170.

[5] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, ‘Machinery health prognostics: A systematic review from data acquisition to RUL prediction’, *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, May 2018, doi: 10.1016/j.ymssp.2017.11.016.

[6] S. Khan and T. Yairi, ‘A review on the application of deep learning in system health management’, *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, Jul. 2018, doi: 10.1016/j.ymssp.2017.11.024.

[7] J. Zhang, D. Zhang, M. Yang, X. Xu, W. Liu, and C. Wen, ‘Fault Diagnosis for Rotating Machinery with Scarce Labeled Samples: A Deep CNN Method Based on Knowledge-Transferring from Shallow Models’, in *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*, Hangzhou: IEEE, Oct. 2018, pp. 482–487. doi: 10.1109/ICCAIS.2018.8570515.

[8] T. Peng, C. Shen, S. Sun, and D. Wang, ‘Fault Feature Extractor Based on Bootstrap Your Own Latent and Data Augmentation Algorithm for Unlabeled Vibration Signals’, *IEEE Trans. Ind. Electron.*, vol. 69, no. 9, pp. 9547–9555, Sep. 2022, doi: 10.1109/TIE.2021.3111567.

[9] M. Hu, C. Wang, C. Zhuang, and Y. Wang, ‘Bearing fault diagnosis method based on data augmentation and

MCNN-LSTM', in *2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*, Chongqing, China: IEEE, Feb. 2023, pp. 663–671. doi: 10.1109/ITNEC56291.2023.10082598.

[10] T. Koenig, L. Cadau, F. Wagner, and M. Kley, 'A generative adversarial network-based data augmentation approach with transient vibration data', *Procedia Computer Science*, vol. 225, pp. 1340–1349, 2023, doi: 10.1016/j.procs.2023.10.122.

[11] S. Shao, P. Wang, and R. Yan, 'Generative adversarial networks for data augmentation in machine fault diagnosis', *Computers in Industry*, vol. 106, pp. 85–93, Apr. 2019, doi: 10.1016/j.compind.2019.01.001.

[12] M. S. Rathore and S. P. Harsha, 'Non-linear Vibration Response Analysis of Rolling Bearing for Data Augmentation and Characterization', *J. Vib. Eng. Technol.*, vol. 11, no. 5, pp. 2109–2131, Jul. 2023, doi: 10.1007/s42417-022-00691-w.

[13] L. Wang, Q. Qu, Y. Wang, D. S.-H. Wong, and Y. Zheng, 'Data Augmentation Integrated with Feature-Enhanced Convolutional Neural Network for Imbalanced Fault Diagnosis in Rolling Bearings', in *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, Kaifeng, China: IEEE, May 2024, pp. 1204–1209. doi: 10.1109/DDCLS61622.2024.10606651.

[14] S. Sun, H. Ding, H. Huang, Z. Zhao, D. Wang, and W. Xu, 'A Novel Cross-Domain Data Augmentation and Bearing Fault Diagnosis Method Based on an Enhanced Generative Model', *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–9, 2024, doi: 10.1109/TIM.2024.3390242.

[15] X. Yang, T. Ye, X. Yuan, W. Zhu, X. Mei, and F. Zhou, 'A Novel Data Augmentation Method Based on Denoising Diffusion Probabilistic Model for Fault Diagnosis Under Imbalanced Data', *IEEE Trans. Ind. Inf.*, vol. 20, no. 5, pp. 7820–7831, May 2024, doi: 10.1109/TII.2024.3366991.

[16] N. Li, Y. Lei, J. Lin, and S. X. Ding, 'An Improved Exponential Model for Predicting Remaining Useful Life of Rolling Element Bearings', *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7762–7773, Dec. 2015, doi: 10.1109/TIE.2015.2455055.

[17] D. Rezende and S. Mohamed, 'Variational Inference with Normalizing Flows', in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, Jun. 2015, pp. 1530–1538. Accessed: Feb. 25, 2026. [Online]. Available: https://proceedings.mlr.press/v37/rezende15.html

[18] M. Russell and P. Wang, 'Normalizing Flows for Intelligent Manufacturing', in *Volume 2: Manufacturing Equipment and Automation; Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability*, New Brunswick, New Jersey, USA: American Society of Mechanical Engineers, Jun. 2023, p. V002T09A004. doi: 10.1115/MSEC2023-101281.

[19] Y. Wang, J. Zhao, C. Yang, D. Xu, and J. Ge, 'Remaining useful life prediction of rolling bearings based on Pearson correlation-KPCA multi-feature fusion', *Measurement*, vol. 201, p. 111572, Sep. 2022, doi: 10.1016/j.measurement.2022.111572.

[20] C. Yin, Y. Li, Y. Wang, and Y. Dong, 'Physics-guided degradation trajectory modeling for remaining useful life prediction of rolling bearings', *Mechanical Systems and Signal Processing*, vol. 224, p. 112192, Feb. 2025, doi: 10.1016/j.ymssp.2024.112192.

[21] G. Wang and J. Xiang, 'Remain useful life prediction of rolling bearings based on exponential model optimized by gradient method', *Measurement*, vol. 176, p. 109161, May 2021, doi: 10.1016/j.measurement.2021.109161.

[22] Y. Qin, C. Yuen, Y. Shao, B. Qin, and X. Li, 'Slow-Varying Dynamics-Assisted Temporal Capsule Network for Machinery Remaining Useful Life Estimation', *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 592–606, Jan. 2023, doi: 10.1109/TCYB.2022.3164683.

[23] C. Yin, Y. Li, Y. Wang, and Y. Dong, 'Physics-guided degradation trajectory modeling for remaining useful life prediction of rolling bearings', *Mechanical Systems and Signal Processing*, vol. 224, p. 112192, Feb. 2025, doi: 10.1016/j.ymssp.2024.112192.

[24] M. Yan, X. Wang, B. Wang, M. Chang, and I. Muhammad, 'Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model', *ISA Transactions*, vol. 98, pp. 471–482, Mar. 2020, doi: 10.1016/j.isatra.2019.08.058.

[25] M. Pandiyan and T. N. Babu, 'Systematic Review on Fault Diagnosis on Rolling-Element Bearing', *J. Vib. Eng. Technol.*, vol. 12, no. 7, pp. 8249–8283, Oct. 2024, doi: 10.1007/s42417-024-01358-4.

[26] J. Zhou, Y. Qin, D. Chen, F. Liu, and Q. Qian, 'Remaining useful life prediction of bearings by a new reinforced memory GRU network', *Advanced Engineering Informatics*, vol. 53, p. 101682, Aug. 2022, doi: 10.1016/j.aei.2022.101682.

[27] L. Song, T. Lin, Y. Jin, S. Zhao, Y. Li, and H. Wang, 'Advancements in bearing remaining useful life

prediction methods: a comprehensive review', *Meas. Sci. Technol.*, vol. 35, no. 9, p. 092003, Sep. 2024, doi: 10.1088/1361-6501/ad5223.

Caractérisation de la complémentarité des détecteurs d'anomalies par l'analyse des contributions SHAP

Jordan Levy[1,2], Paul Saves[1], Moncef Garouani[1], Nicolas Verstaevel[1], Benoit Gaudou[1]

[1] IRIT, Université Toulouse Capitole
[2] TwinswHeel, Soben

jordan.levy@irit.fr, paul.saves@irit.fr, moncef.garouani@irit.fr, nicolas.verstaevel@irit.fr, benoit.gaudou@irit.fr

Résumé

La détection d'anomalies non supervisée est un problème difficile en raison de la diversité des distributions de données et de l'absence d'étiquettes. Les méthodes ensemblistes sont souvent adoptées pour pallier ces difficultés en combinant plusieurs détecteurs d'anomalies pour réduire les biais individuels et augmenter la robustesse. Cependant, construire un ensemble véritablement complémentaire reste difficile car de nombreux détecteurs reposent sur des critères de discrimination similaires et finissent par produire des scores d'anomalie redondants. Par conséquent, le potentiel de l'apprentissage ensembliste est souvent limité par la difficulté d'identifier des modèles qui capturent vraiment différents types d'irrégularités. Pour remédier à cela, nous proposons une méthodologie pour caractériser les détecteurs d'anomalies à travers leurs mécanismes de décision. En utilisant les explications additives de Shapley (SHAP), nous quantifions comment chaque modèle attribue de l'importance aux caractéristiques d'entrée, et nous utilisons ces profils d'attribution pour mesurer la similarité entre les détecteurs. Nous montrons que les détecteurs ayant des explications similaires ont tendance à produire des scores d'anomalie corrélés et à identifier des anomalies qui se chevauchent largement. Inversement, la divergence des explications indique de manière fiable un comportement de détection complémentaire. Nos résultats démontrent que les métriques basées sur les explications offrent un critère différent, souvent meilleur, des sorties brutes pour sélectionner des modèles dans un ensemble. Cependant, nous démontrons également que la diversité seule est insuffisante ; une performance individuelle élevée des détecteurs d'anomalies reste un prérequis pour des ensembles efficaces. En ciblant explicitement la diversité des explications tout en maintenant la qualité des modèles, nous sommes capables de construire des ensembles plus diversifiés, plus complémentaires et finalement plus efficaces pour la détection d'anomalies non supervisée.

Mots-clés

Détection d'anomalies non supervisée, Modèle ensembliste, Sélection de modèles, Explicabilité.

Abstract

Unsupervised anomaly detection is a challenging problem due to the diversity of data distributions and the lack of labels. Ensemble methods are often adopted to mitigate these challenges by combining multiple anomaly detectors, which can reduce individual biases and increase robustness. Yet building an ensemble that is genuinely complementary remains challenging, since many detectors rely on similar decision cues and end up producing redundant anomaly scores. As a result, the potential of ensemble learning is often limited by the difficulty of identifying models that truly capture different types of irregularities. To address this, we propose a methodology for characterizing anomaly detectors through their decision mechanisms. Using SHapley Additive exPlanations, we quantify how each model attributes importance to input features, and we use these attribution profiles to measure similarity between detectors. We show that detectors with similar explanations tend to produce correlated anomaly scores and identify largely overlapping anomalies. Conversely, explanation divergence reliably indicates complementary detection behavior. Our results demonstrate that explanation-driven metrics offer a different, usually better, criterion than raw outputs for selecting models in an ensemble. However, we also demonstrate that diversity alone is insufficient ; high individual model performance remains a prerequisite for effective ensembles. By explicitly targeting explanation diversity while maintaining model quality, we are able to construct ensembles that are more diverse, more complementary, and ultimately more effective for unsupervised anomaly detection.

Keywords

Unsupervised Anomaly Detection, Ensemble Learning, Model Selection, Explainable AI.

1 Introduction

La détection d'anomalies est un problème difficile, principalement en raison de la nature intrinsèque des anomalies. Les anomalies sont définies comme des déviations par rapport à ce qui est considéré comme un comportement normal [5]. Par conséquent, selon la façon dont cette normalité est définie, un algorithme ou un autre peut être plus adapté pour une

bonne détection. Par exemple, certaines méthodes peuvent définir la normalité en termes géométriques et utiliser des règles basées sur la distance pour repérer les valeurs aberrantes, tandis que d'autres utilisent des hypothèses probabilistes et signalent les instances rares ou à faible probabilité comme des anomalies [19].

Dans de nombreux contextes réels, l'obtention d'anomalies étiquetées est coûteuse ou irréalisable car les événements anormaux sont rares, coûteux à produire ou dangereux à provoquer (*e.g.*, dans l'industrie nucléaire la détection doit fonctionner sans attendre que des défauts se produisent [17]). Par conséquent, les approches semi-supervisées et non supervisées sont souvent préférées. Dans la détection d'anomalies non supervisée (Unsupervised Anomaly Detection, UAD), les spécialistes, ne pouvant pas définir exhaustivement le comportement "normal" à la main, s'appuient généralement sur des méthodes d'apprentissage automatique. Ainsi, au lieu de définir eux-mêmes la normalité, ils utilisent les hypothèses sous-jacentes des algorithmes (géométriques, probabilistes, etc.). Cependant, aucune hypothèse n'est garantie d'être appropriée pour toutes les applications [1], et comme indiqué dans le théorème du "*no free lunch*", aucun détecteur unique ne surpasse systématiquement les autres sur tous les jeux de données [22, 9]. Néanmoins, le choix de l'algorithme reste critique, car une sélection inappropriée peut conduire à de mauvaises performances [1]

Une stratégie courante pour atténuer ce problème consiste à combiner plusieurs détecteurs au sein d'une méthode ensembliste, en tirant parti des atouts de chaque détecteur et de la diversité de leurs hypothèses sur ce qui constitue un comportement normal. L'apprentissage ensembliste a été largement adopté pour cette raison et a démontré de solides performances empiriques dans divers scénarios de détection d'anomalies [1]. En agrégeant des détecteurs avec divers biais inductifs, les approches ensemblistes peuvent capturer une plus grande variété de types d'anomalies. Cette diversité se traduit souvent par une meilleure couverture de détection [21].

Un défi clé dans l'apprentissage ensembliste est la sélection de détecteurs appropriés. Bien qu'une approche ensembliste robuste nécessite théoriquement des détecteurs à la fois diversifiés et performants [21], l'identification de tels comportements complémentaires reste un problème ouvert [13].

Pour relever ce défi, nous proposons une nouvelle méthodologie qui caractérise le comportement des détecteurs d'anomalies en utilisant les explications additives de Shapley (SHAP) [12]. Contrairement aux approches reposant uniquement sur les sorties, nous nous concentrons sur la compréhension des mécanismes de décision internes des détecteurs d'anomalies. Notre étude révèle que les détecteurs ayant des schémas d'explication similaires ont tendance à produire des scores d'anomalie redondants, tandis que la divergence des explications est un indicateur fort de complémentarité. Par conséquent, nous démontrons que la sélection de détecteurs basée sur leur comportement d'explication conduit à une amélioration des performances du modèle ensembliste. Cela conduit à trois contributions majeures :

— Une analyse des algorithmes d'UAD basée sur leurs explications SHAP, qui démontre que leurs comportements sont corrélés à la similarité de leurs sorties.
— Une comparaison entre les explications SHAP et les similarités des sorties de modèles pour sélectionner des modèles de détection d'anomalies diversifiés.
— Une étude empirique quantifiant l'importance relative de la diversité par rapport à la performance individuelle, démontrant que la qualité du modèle reste un prérequis critique.

Cet article est structuré comme suit : la Section 2 présente les travaux connexes. La méthodologie de notre approche est décrite dans la Section 3. Les résultats expérimentaux sont présentés en Section 4. Enfin, nous concluons l'article dans la Section 5.

2 Travaux Connexes

2.1 Algorithmes de détection d'anomalies non supervisée

L'UAD est un problème largement étudié [19] avec un grand nombre d'algorithmes, chacun fondé sur des hypothèses différentes concernant la nature des anomalies. Des bibliothèques comme PyOD [23] ont été introduites pour standardiser l'utilisation de ces modèles. Parmi ceux-ci, les méthodes basées sur la distance (KNN, LOF, CBLOF) caractérisent les anomalies en fonction de leur éloignement des points voisins. De leur côté, les algorithmes basés sur la reconstruction (AutoEncoder, PCA) détectent les anomalies en apprenant à reconstruire des modèles de données normaux. D'autres approches, telles que HBOS, ECOD et COPOD, exploitent des hypothèses probabilistes, tandis que OCSVM et DeepSVDD utilisent la classification à une classe. Enfin, des algorithmes comme Isolation Forest et LODA reposent respectivement sur le partitionnement aléatoire et la projection spatiale.

2.2 Sélection de modèles pour la détection d'anomalies non supervisée

La sélection d'un modèle approprié pour une tâche d'UAD est communément appelée Sélection de Modèle de Valeurs Aberrantes Non Supervisée (Unsupervised Outlier Model Selection, UOMS). L'UOMS est un problème particulièrement difficile qui a gagné en importance à mesure que le nombre de modèles disponibles, comprenant diverses familles algorithmiques et configurations d'hyperparamètres, continue de s'étendre avec l'apparition de nouvelles méthodes dans la littérature [19].

Une stratégie courante en UOMS consiste à estimer la qualité du modèle directement à partir de données non étiquetées de manière non supervisée, pour sélectionner les meilleurs modèles. Ces approches se divisent en deux catégories principales. Les méthodes *autonomes* (stand-alone) [8, 16] calculent un score non supervisé pour chaque détecteur indépendamment, tandis que les méthodes *basées sur le consensus* (consensus-based) évaluent les détecteurs en mesurant l'accord au sein d'un groupe de modèles et en sélectionnant ceux qui se conforment le mieux au groupe [6, 11]. Les preuves empiriques indiquent que les critères d'évaluation

autonomes échouent souvent à fournir des résultats cohérents ou fiables en pratique. En revanche, les approches basées sur le consensus, malgré leur coût de calcul plus élevé, sont apparues comme une stratégie plus efficace et prometteuse pour l'UOMS [13].

Il existe peu de travaux dans la littérature scientifique sur l'UOMS pour l'apprentissage d'ensemble. Dans [21], les auteurs montrent que la diversité des hypothèses algorithmiques tend à donner de meilleurs résultats. Pour caractériser la diversité, les auteurs utilisent une corrélation de Pearson pondérée entre les scores d'anomalie des modèles. Cependant, ils soulignent également que la diversité est importante, mais que les algorithmes choisis doivent déjà avoir de bonnes performances sur le jeu de données pour obtenir un modèle d'ensemble performant. À partir de l'hypothèse précédente, les auteurs de [18] ont introduit SELECT, qui, au lieu d'essayer de sélectionner des modèles en fonction de leur diversité, sélectionne des modèles en fonction de leurs performances à partir d'une pseudo-étiquette de vérité terrain créée à partir des scores d'anomalie. Plus récemment, dans [3], les auteurs font de la détection d'anomalies dans les séries temporelles en utilisant un ensemble de plusieurs auto-encodeurs convolutionnels. Ils optimisent la diversité en intégrant une métrique basée sur la dissimilarité des sorties de reconstruction. Ils démontrent également que cette stratégie axée sur la diversité améliore les performances de détection.

Bien qu'il ait été démontré que les sorties des modèles comme les scores d'anomalies peuvent refléter la diversité, dans ce travail, nous étudions une méthode de consensus basée sur les valeurs SHAP, qui caractérisent le comportement des modèles de détection d'anomalies sur différents jeux de données.

2.3 Explicabilité

L'Intelligence Artificielle Explicable (Explainable Artificial Intelligence, XAI) cherche à rendre les modèles complexes transparents en produisant des représentations interprétables par l'homme sur la manière dont les entrées, la structure du modèle et l'incertitude produisent des sorties particulières [14]. Parmi les méthodes d'explication locales et agnostiques au modèle, les explications additives de Shapley (SHAP) sont largement utilisées car elles fournissent des attributions de caractéristiques axiomatiques au niveau de l'instance [12].

Au-delà de l'interprétabilité, les représentations SHAP ont été exploitées pour comparer des modèles et former des représentations informatives : des travaux récents utilisent des attributions SHAP agrégées pour identifier des familles de modèles ou pour servir de caractéristiques pour des tâches d'apprentissage en aval [20, 7]. À notre connaissance, l'exploitation de SHAP pour la comparaison et la sélection de modèles n'a pas été systématiquement étudiée dans le cadre de l'UAD, où l'absence d'étiquettes rend la sélection des modèles particulièrement difficile.

3 Méthodologie

3.1 Cadre du problème

Nous considérons le problème d'UAD, où l'objectif est d'identifier des échantillons anormaux au sein d'un jeu de données $\mathcal{D} = \{x_1, \ldots, x_n\}$ contenant n instances dans un espace de caractéristiques à d dimensions. Soit $\mathcal{M}$ un ensemble de m modèles de détection d'anomalies $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$. Pour un jeu de données donné, chaque modèle M_i produit un vecteur de score d'anomalie $s_i = M_i(\mathcal{D}) \in \mathbb{R}^n$ où $s_i^{(k)}$ est le score d'anomalie du $k^{\text{ième}}$ point d'entrée selon le $i^{\text{ième}}$ modèle. Chaque modèle produit également un vecteur de prédictions binaires $a_i \in \{0,1\}^n$, obtenu par un seuillage du vecteur de scores d'anomalie, indiquant directement la présence ou l'absence d'anomalies. Plus précisément pour chaque instance k, la prédiction est $a_i^{(k)} = 0$ si $s_i^{(k)} < \tau_i$, et 1 sinon. Le paramètre τ_i est un seuil de décision interne, propre à chaque modèle i. Notre objectif est d'analyser le comportement de ces modèles et d'identifier des groupes de modèles qui partagent des structures interprétatives similaires.

3.2 Similarité des modèles à partir des explications

Pour chaque modèle M_i, nous calculons sa matrice d'explication SHAP $Sh_i \in \mathbb{R}^{n \times d}$, où $Sh_i^{(k)} \in \mathbb{R}^d$ est le vecteur représentant la contribution de chaque caractéristique au score d'anomalie de l'instance x_k.
Nous définissons la similarité de comportement entre deux modèles M_i et M_j comme la corrélation de Pearson moyenne par instance entre leurs vecteurs SHAP :

$$\rho_{ij}^{\text{PS}} = \frac{1}{n} \sum_{k=1}^{n} \text{corr}\big(Sh_i^{(k)}, Sh_j^{(k)}\big).$$

Pour capturer la cohérence du classement entre les importances des caractéristiques plutôt que les magnitudes brutes, nous calculons également une similarité basée sur le Gain Cumulé Actualisé Normalisé (Normalized Discounted Cumulative Gain, NDCG) :

$$\rho_{ij}^{\text{NDCG}} = \frac{1}{2n} \sum_{k=1}^{n} \Big(\text{NDCG}\left(|Sh_i^{(k)}|, |Sh_j^{(k)}|\right) + \text{NDCG}\left(|Sh_j^{(k)}|, |Sh_i^{(k)}|\right) \Big),$$

où le NDCG évalue l'accord dans l'importance classée des caractéristiques entre deux détecteurs, et $|\cdot|$ correspond à la valeur absolue. Si les deux modèles attribuent une importance SHAP élevée aux mêmes caractéristiques, leur valeur NDCG sera proche de 1, indiquant un comportement explicatif similaire [2].

3.3 Lier les similarités des explications aux sorties des détecteurs

Pour comparer la similarité des explications des modèles avec la similarité des sorties, nous introduisons deux matrices supplémentaires : la matrice des corrélations de scores et la matrice des similarités de Jaccard.

Nous définissons la similarité entre deux modèles M_i et M_j sur la base de la corrélation de leurs scores d'anomalie. Plus précisément, nous calculons la corrélation de Pearson moyenne par instance entre leurs vecteurs de scores comme suit :

$$\rho_{ij}^{\text{Score}} = \frac{1}{n}\sum_{k=1}^{n} \text{corr}\big(s_i^{(k)}, s_j^{(k)}\big).$$

Il en résulte une matrice symétrique qui reflète la similarité par paires entre les modèles vis-à-vis des scores attribués.
Pour comparer directement les prédictions de deux modèles i et j, la similarité de Jaccard entre les deux vecteurs de prédictions a_i et a_j est calculée comme suit :

$$J_{ij} = \frac{|a_i \cap a_j|}{|a_i \cup a_j|},$$

avec $|\cdot|$ le cardinal de l'ensemble. Cette métrique mesure le chevauchement entre les modèles : une valeur de 1 implique des prédictions identiques, tandis que 0 implique des ensembles disjoints d'anomalies détectées.
Chaque matrice de similarité $P \in \{\rho^{PS}, \rho^{NDCG}, \rho^{Scores}, J\}$ peut également être transformée en une matrice de dissimilarité $D \in \{\delta^{PS}, \delta^{NDCG}, \delta^{Scores}, \delta^{J}\}$. Ces matrices sont calculées selon la relation $D = 1 - P$, où chaque élément représente la distance entre deux détecteurs.
Afin de quantifier la relation entre les différentes mesures, nous utilisons le test de Mantel [15] pour déterminer si deux matrices de dissimilarité données sont statistiquement corrélées. Le coefficient de Mantel (r_M) est calculé comme la corrélation de Pearson entre les éléments triangulaires supérieurs des matrices, la signification statistique étant établie par des tests de permutation.

4 Expérimentations

Nous avons mené des expériences pour répondre aux questions suivantes : (1) La similarité des explications implique-t-elle une similarité des prédictions ? (2) Les métriques basées sur SHAP surpassent-elles les sorties brutes pour quantifier la diversité ? et (3) Dans quelle mesure cette diversité impacte-t-elle la précision et la robustesse de l'ensemble résultant ?

4.1 Configuration expérimentale

Dans les expériences, 14 algorithmes UAD ont été utilisés : COF, KNN, LOF, IForest, PCA, CBLOF, LODA, HBOS, MCD, OCSVM, DAGMM, DeepSVDD, COPOD et ECOD tels qu'implémentés dans la bibliothèque PyOD [23]. Les hyperparamètres définis par les auteurs de chaque modèle ont été conservés et aucune indication du pourcentage d'anomalie n'a été fournie pour aucun jeu de données. Au niveau de l'explicabilité, nous utilisons l'implémentation de la méthode agnostique Kernel SHAP disponible dans la bibliothèque Python `shap`. Le choix de cette méthode était nécessaire pour assurer un cadre unifié d'explication à travers nos divers algorithmes. De plus, pour assurer une approximation stable, le jeu de données d'arrière-plan a été résumé en utilisant l'algorithme de partitionnement k-means avec $k = 50$ centroïdes.

Pour augmenter la robustesse des résultats, chaque jeu de données a été divisé à cinq reprises en un ensemble d'entraînement et un ensemble de test, l'ensemble d'entraînement représentant 80 % des données. La graine aléatoire pour chaque division a été fixée égale à l'indice d'itération pour assurer la reproductibilité. Le code source utilisé dans les expériences est disponible sur GitHub [1].

4.2 Jeux de données considérés

En raison du coût de calcul élevé associé à SHAP, la sélection des jeux de données a été restreinte aux 50 % plus petits jeux de données disponibles dans ADBench [9]. De plus, les jeux de données contenant plus de 20 caractéristiques ont été supprimés pour maintenir la faisabilité des calculs. Ce processus de filtrage initial a abouti à un ensemble de 16 jeux de données provenant de diverses applications : annthyroid (AN), breastw (BR), glass (GL), Hepatitis (HE), Lymphography (LY), mammography (MA), PageBlocks (PB), Pima (PI), Stamps (ST), thyroid (TH), vertebral (VE), vowels (VO), WBC (WB), Wilt (WL), wine (WN) et yeast (YE).

4.3 Corrélation entre les matrices de similarité

Nous avons calculé les quatre matrices de similarité ρ^{PS} (corrélation linéaire entre les SHAP), ρ^{NDCG} (similarité des classements d'importance des caractéristiques SHAP), ρ^{Scores} (corrélation linéaire entre les scores d'anomalie), J (indice de Jaccard entre les prédictions d'anomalie) pour chaque jeu de données. Les matrices moyennes sur tous les jeux de données sont présentées dans la Figure 1, où un partitionnement hiérarchique a été appliqué pour optimiser l'ordre. Visuellement, deux groupes de modèles émergent. Premièrement, COPOD et ECOD se regroupent systématiquement, ce qui est attendu car les deux algorithmes reposent sur des hypothèses de distribution de données similaires. Deuxièmement, un groupe plus large comprenant OCSVM, AutoEncoder, IForest, PCA, KNN, CBLOF et GMM présente une forte corrélation. Ces algorithmes partagent des caractéristiques sous-jacentes liées aux métriques de distance et aux stratégies d'encodage des données.
Le test de Mantel est utilisé pour évaluer si les matrices de similarité sont statistiquement corrélées. Le Tableau 1 rapporte les corrélations de Mantel moyennes sur les jeux de données. Nous observons une forte corrélation ($r_M = 0.83$) entre δ^{PS} et δ^{NDCG}, confirmant que la similarité basée sur SHAP est cohérente en termes de magnitude et de classement de l'importance des caractéristiques. De même, δ^{Scores} et δ^{J} montrent une forte corrélation ($r_M = 0.76$), ce qui est intuitif puisque les modèles avec des distributions de scores similaires ont tendance à produire des prédictions binaires similaires. Enfin, la corrélation entre δ^{PS} et δ^{J} ($r_M = 0.67$) suggère que les détecteurs partageant des schémas de raisonnement similaires (tels que capturés par SHAP) ont également tendance à donner des prédictions d'anomalie similaires. Cependant, la similarité entre les modèles dépend du jeu de données. Plus précisément, le Tableau 2 montre les corrélations

1. https://github.com/jordanlv/Analyzing-SHAP-UOMS

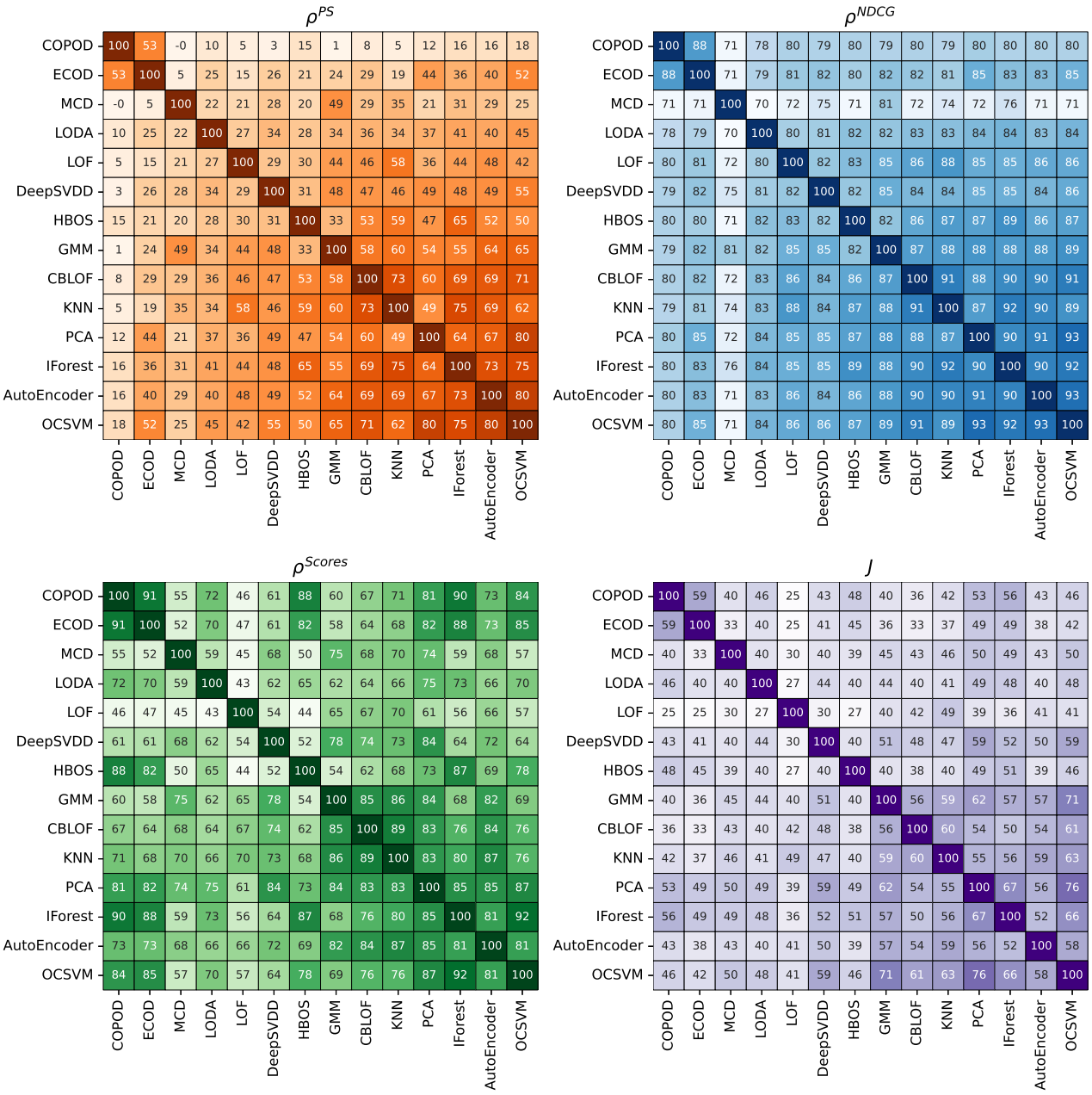


FIGURE 1 – Similarité moyenne entre les modèles sur tous les jeux de données. Disposition : Haut-Gauche : corrélations des valeurs SHAP ; Haut-Droite : NDCG des SHAP ; Bas-Gauche : corrélations des scores d'anomalie ; et Bas-Droite : similarités de Jaccard.

entre δ^{PS} et δ^{Scores}, et entre δ^{PS} et δ^{J} pour chaque jeu de données. Habituellement, les deux paires sont corrélées. Certains jeux de données montrent de fortes corrélations entre les matrices ($r_M > 0.5$) comme AN, PI, TH ou VO, tandis que d'autres ont une corrélation modérée ou faible entre les deux matrices. Enfin, certains jeux de données intéressants sont BR, LY ou YE où $r_M(\delta^{PS}, \delta^{Scores}) \ll r_M(\delta^{PS}, \delta^{J})$.

Dans l'ensemble, la similarité des modèles est corrélée avec leurs prédictions. Dans les sections suivantes, nous étudions la combinaison de modèles éloignés dans les matrices de similarité pour améliorer les résultats globaux d'un ensemble.

4.4 Agrégation des prédictions des modèles

L'objectif d'une méthode ensembliste est de combiner les prédictions de plusieurs détecteurs pour créer une méthode plus robuste. Un défi dans ce type de méthode est de savoir comment agréger chaque prédiction. Chaque détecteur d'anomalies produit un score d'anomalie et une prédiction indiquant si un point de données est normal ou anormal. Pour tirer véritablement parti de la spécialité de chaque modèle, nous utilisons les scores d'anomalie comme entrée

TABLE 1 – Corrélations de Mantel moyennes entre les matrices de distance, moyennées sur les jeux de données. Toutes les corrélations sont significatives ($p \leq 0.004$).

	δ^{PS}	δ^{NDCG}	δ^{Scores}	δ^{J}
δ^{PS}	1.00	0.83	0.55	0.67
δ^{NDCG}		1.00	0.54	0.57
δ^{Scores}			1.00	0.76
δ^{J}				1.00

TABLE 2 – Corrélations de Mantel ($\times 10^2$) entre les matrices de distance pour chaque jeu de données.

Jeu de données	$r_M(\delta^{PS}, \delta^{Scores})$	$r_M(\delta^{PS}, \delta^{J})$
AN	75	78
BR	22	51
GL	42	24
HE	26	51
LY	36	84
MA	56	51
PB	33	66
PI	55	52
ST	56	53
TH	72	76
VE	26	-4
VO	56	58
WB	18	15
WL	49	49
WN	26	30
YE	38	62

pour notre fonction d'agrégation. Les méthodes d'agrégation connues pour ces scores incluent l'utilisation du maximum, de la moyenne arithmétique, ou la moyenne des rangs (une méthode consistant à substituer les scores bruts par leur classement d'anormalité avant de les moyenner) [1]. Nous avons évalué ces fonctions sur tous les ensembles possibles de 3 modèles distincts parmi notre groupe de 14 modèles (résultant en 364 ensembles) sur chaque jeu de données. Compte tenu du déséquilibre de classe de chaque jeu de données, nous utilisons l'Aire Sous la Courbe Précision-Rappel (AUCPR) pour évaluer les performances. Comme le montre le Tableau 3, l'agrégation par rang donne des résultats supérieurs sur 11 des 16 jeux de données. Par conséquent, nous adoptons l'agrégation par rang pour le reste de cette étude.

4.5 Complémentarité

Pour sélectionner les modèles les plus dissimilaires, nous utilisons les matrices de dissimilarité D, ce qui donne des matrices sur la distance entre deux détecteurs. De nouveau, à partir du groupe de 14 modèles, nous avons construit des ensembles de taille $n = 3$. Le Tableau 4 présente les corrélations entre l'AUCPR des ensembles et les distances de diversité pour chaque matrice de dissimilarité. Une corrélation positive entre la diversité et la performance implique que l'augmentation de la diversité de l'ensemble conduit à des résultats de détection supérieurs.

TABLE 3 – AUCPR moyen ($\times 10^2$) entre tous les 364 ensembles avec les 3 principales stratégies d'agrégation. Les meilleurs résultats sont en gras.

Jeu de données	Rang	Max	Moyenne
AN	**26**	21	9
BR	**98**	59	49
GL	18	**22**	19
HE	**84**	64	47
LY	**100**	68	36
MA	**30**	8	4
PB	**60**	35	12
PI	**55**	45	42
ST	**62**	35	22
TH	**58**	28	5
VE	18	**28**	**28**
VO	**41**	11	8
WB	**97**	41	25
WL	6	**9**	7
WN	49	**53**	31
YE	37	**40**	39
moyenne	**52**	35	24

TABLE 4 – Corrélation ($\times 10^2$) entre la distance des modèles et l'AUCPR des ensembles. Les plus grands résultats sont en gras.

Jeu de données	δ^{PS}	δ^{NDCG}	δ^{Scores}	δ^{J}
AN	-20	**39**	-1	-32
BR	**-0**	-17	-90	-59
GL	**5**	-2	-33	-37
HE	**-28**	-32	-48	-31
LY	15	-0	-29	**16**
MA	**-9**	-68	-18	3
PB	-18	**21**	-17	-4
PI	**37**	14	30	24
ST	**40**	15	15	36
TH	-38	**19**	-12	-48
VE	-44	-30	-12	**-7**
VO	-66	-54	**-0**	-3
WB	**51**	36	-11	-28
WL	55	**62**	46	55
WN	43	**48**	7	1
YE	29	37	34	**45**
moyenne	3	**5**	-9	-4

Le tableau présente trois informations clés. Premièrement, la sélection de la diversité à partir des valeurs SHAP tend

à donner de meilleures performances que les sorties des modèles (scores et prédictions). Sur 11 des 16 jeux de données, l'utilisation de δ^{PS} et δ^{NDCG} comme diversité tend à donner de meilleurs résultats que l'utilisation de δ^{Scores} et δ^{J}. Deuxièmement, comme indiqué précédemment, la diversité contribue à l'apprentissage d'ensemble en élargissant la gamme des anomalies détectées. Cependant, une sélection de modèles efficace doit également tenir compte de la performance individuelle des modèles, un facteur non explicitement optimisé dans cette étude. Cette limitation explique probablement pourquoi les corrélations entre les métriques de distance et la performance restent modérées. De plus, dans certains jeux de données, ces corrélations sont systématiquement négatives, suggérant que la diversité n'est pas toujours bénéfique. Ce phénomène se produit notamment lorsqu'un seul modèle est plus performant que les autres. Dans de tels cas, la condition principale pour un ensemble réussi est l'inclusion de ce modèle spécifique. De plus, des corrélations négatives peuvent survenir dans des jeux de données très complexes où tous les modèles présentent de mauvaises performances. Dans ces scénarios, même une grande diversité ne peut pas compenser le manque de détection significative, entraînant un ensemble inefficace. Enfin, il est intéressant de noter que les SHAP et les scores ne montrent pas la même diversité car les corrélations varient entre les jeux de données. Par exemple, sur le jeu de données WB, les corrélations entre δ^{Scores} et δ^{J} sont négatives, tandis que δ^{PS} et δ^{NDCG} sont positives. Par conséquent, les deux métriques mettent en évidence des diversités différentes.

4.6 Diversité et performances individuelles

Bien que nous ayons montré que la diversité améliore la performance de l'ensemble en élargissant la couverture des anomalies, notre analyse précédente a négligé la précision individuelle des modèles. Ici, nous affinons nos résultats pour démontrer que malgré la valeur de la diversité, la qualité individuelle de chaque modèle reste un facteur critique. Pour les besoins de cette section, nous utilisons δ^{PS} pour quantifier la diversité.

La Figure 2 illustre l'importance de la performance individuelle de chaque modèle pour obtenir un ensemble efficace. Plus précisément, la Figure 2a présente les résultats pour le jeu de données LY. Une corrélation claire entre la diversité, la performance individuelle et la précision de l'ensemble est observable sur ce jeu de données. Cependant, cette corrélation n'est pas toujours bénéfique. Dans certains scénarios, imposer la diversité peut être préjudiciable à l'ensemble. Par exemple, la Figure 2b révèle que pour le jeu de données VO, la diversité offre un gain négligeable et peut même conduire à des résultats sous-optimaux.

Pour évaluer l'impact relatif de la qualité du modèle par rapport à la diversité, nous avons effectué une régression linéaire pour prédire le gain de performance de l'ensemble en utilisant la performance individuelle moyenne et les scores de diversité. Les poids résultants, présentés dans le Tableau 5, indiquent que bien que la performance individuelle soit généralement le facteur dominant, la diversité joue un rôle complémentaire crucial. Dans 12 des 16 jeux de données, le coefficient de diversité est positif, confirmant sa valeur en tant qu'amplificateur de performance. Notamment, pour les jeux de données WB et LY, le poids de la diversité rivalise ou dépasse même celui de la performance individuelle (ratios de 1,2 et 0,8 respectivement). Dans l'ensemble, avec un ratio moyen de 0,2, la diversité joue un rôle secondaire mais précieux dans la conception d'ensembles UAD.

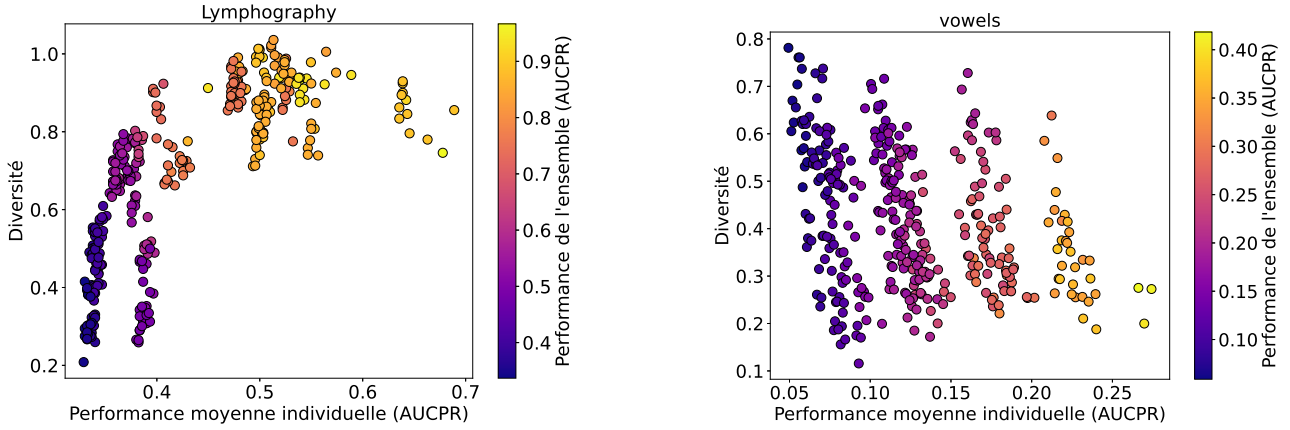
TABLE 5 – Poids de régression linéaire ($\times 10^2$) prédisant la performance de l'ensemble à partir de la performance individuelle moyenne et de la diversité, ainsi que leur ratio (vert : positif ; rouge : négatif).

Jeu de données	Perf. Indiv.	Diversité	Ratio
AN	2.7	0.5	0.2
BR	6.3	2.0	0.3
GL	1.1	0.1	0.1
HE	6.5	0.5	0.1
LY	10.7	8.4	0.8
MA	3.4	0.7	0.2
PB	5.0	2.2	0.4
PI	1.7	0.2	0.1
ST	5.6	0.3	0.0
TH	5.7	1.7	0.3
VE	0.1	-0.0	-0.1
VO	7.5	-0.6	-0.1
WB	2.4	2.8	1.2
WL	0.0	0.0	0.4
WN	8.0	-0.1	-0.0
YE	0.3	-0.0	-0.0
moyenne	4.2	1.2	0.2

5 Conclusion

Nous avons présenté une méthodologie pour sélectionner des modèles dans des ensembles UAD, basée sur la similarité de leurs explications. Nous avons démontré que la diversité s'avère bénéfique et permet d'améliorer les résultats. Nous avons analysé quatre métriques de diversité : deux fondées sur les explications SHAP et deux directement sur les sorties des modèles. Ces métriques ont conduit à des résultats différents lors de la sélection, indiquant qu'elles capturent des formes de diversité distinctes. De manière générale, les métriques basées sur SHAP ont affiché des résultats supérieurs à ceux basés sur les sorties. Enfin, nous avons établi que malgré les avantages de la diversité, la performance individuelle des modèles reste le facteur décisif : des modèles faibles mais diversifiés ne peuvent pas surpasser des modèles forts mais similaires. Cette recherche tend à montrer que l'explicabilité devrait être davantage prise en compte dans l'UOMS, car elle fournit de nouvelles informations sur le comportement des modèles.

Une limite de notre approche réside dans le coût de calcul de



(a) Jeu de données Lymphography (LY).

(b) Jeu de données Vowels (VO).

FIGURE 2 – Relation entre la diversité de l'ensemble (donnée par δ^{PS}) et la performance individuelle moyenne. Chaque point représente un ensemble de modèles. L'échelle de couleurs indique la performance globale de l'ensemble (AUCPR).

SHAP, qui peut devenir prohibitif pour les jeux de données comportant un grand nombre d'instances ou de caractéristiques. Cependant, le coût de calcul peut être atténué en utilisant des techniques d'approximation ou des modèles de substitution. De plus, notre stratégie est agnostique quant à la méthode d'explication, permettant l'utilisation de techniques d'interprétabilité moins coûteuses si nécessaire. Par exemple, dans [4], les auteurs ont récemment démontré comment l'agrégation de profils de dépendance partielle (PDP) sur un ensemble de modèles quasi-optimaux peut fournir des métriques fiables pour l'incertitude et la robustesse des explications.

Bien que nous ayons modélisé avec succès la diversité, l'optimisation des modèles individuels constitue une perspective d'évolution naturelle de ces travaux. L'ajustement fin des hyperparamètres permettrait de maximiser le potentiel de chaque détecteur, offrant ainsi un levier supplémentaire pour améliorer la performance globale de l'ensemble.

Concernant les travaux futurs, nous envisageons d'étudier la divergence entre les similarités basées sur SHAP et celles issues des sorties brutes afin d'affiner le processus de sélection de modèles. Par ailleurs, la performance individuelle étant critique, l'intégration de méthodes d'estimation de la qualité des modèles au sein de notre méthodologie constitue une priorité. Enfin, nous visons à étendre cette stratégie à l'UAD pour les séries temporelles, un domaine où la complexité accrue rend les approches ensemblistes particulièrement pertinentes [10].

Remerciements

La recherche présentée dans cet article a bénéficié d'un financement de l'Association Nationale de la Recherche et de la Technologie sous le numéro de subvention CIFRE 2023/1398 et de la société Soben. Les auteurs remercient également l'Agence Nationale de la Recherche pour le financement du projet MIMICO sous le numéro de subvention ANR-24-CE23-0380.

Références

[1] Charu C Aggarwal. *Outlier Analysis*. Springer, 2016.

[2] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

[3] David Campos, Tung Kieu, Chenjuan Guo, Feiteng Huang, Kai Zheng, Bin Yang, and Christian S Jensen. Unsupervised time series outlier detection with diversity-driven convolutional ensembles. *Proceedings of the VLDB Endowment*, 15(3) :611–623, 2021.

[4] Mustafa Cavus, Jan N van Rijn, and Przemysław Biecek. Beyond the single-best model : Rashomon partial dependence profile for trustworthy explanations in automl. In *International Conference on Discovery Science*, pages 445–459. Springer, 2025.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection : A survey. *ACM computing surveys (CSUR)*, 41(3) :1–58, 2009.

[6] Sunny Duan, Loic Matthey, Andre Saraiva, Nicholas Watters, Christopher P Burgess, Alexander Lerchner, and Irina Higgins. Unsupervised model selection for variational disentangled representation learning. *arXiv preprint arXiv :1905.12614*, 2019.

[7] Moncef Garouani, Ayah Barhrhouj, and Olivier Teste. Xstacking : An effective and inherently explainable framework for stacked ensemble learning. *Information Fusion*, 2025.

[8] Nicolas Goix. How to evaluate the quality of unsupervised anomaly detection algorithms ? *arXiv preprint arXiv :1607.01152*, 2016.

[9] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. ADBench : anomaly detection benchmark. *NeurIPS*, 35 :32142–32159, 2022.

[10] Jordan Levy, Clément Blanco-Volle, Nicolas Verstaevel, Benoit Gaudou, and Vincent Talon. Timeciel : Contextual interactive ensemble learning for time series classification. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 316–327. Springer, 2025.

[11] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. InfoGAN-CR and ModelCentrality : Self-supervised Model Training and Selection for Disentangling GANs. In *International conference on machine learning*, pages 6127–6139. PMLR, 2020.

[12] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *NeurIPS*, 30, 2017.

[13] Martin Q Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. The need for unsupervised outlier model selection : A review and evaluation of internal evaluation strategies. *ACM SIGKDD Explorations Newsletter*, 25(1) :19–35, 2023.

[14] Andreas Madsen, Himabindu Lakkaraju, Siva Reddy, and Sarath Chandar. Interpretability needs a new paradigm. *arXiv*, 2024.

[15] Nathan Mantel. The detection of disease clustering and a generalized regression approach. *Cancer research*, 27 :209–220, 1967.

[16] Henrique O Marques, Ricardo JGB Campello, Jörg Sander, and Arthur Zimek. Internal evaluation of unsupervised outlier detection. *TKDD*, 14(4) :1–42, 2020.

[17] Sang Won Oh, Hye Seon Jo, Ho Jun Lee, Man Gyun Na, SW Oh, HS Jo, HJ Lee, and MG Na. Anomalies detection by unsupervised learning using explainable artificial intelligence in nuclear power plants. In *Transactions of the Korean Nuclear Society Spring Meeting Jeju, Korea*, 2022.

[18] Shebuti Rayana and Leman Akoglu. Less is more : Building selective anomaly ensembles. *TKDD*, 10(4) :1–33, 2016.

[19] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5) :756–795, 2021.

[20] Paul Saves, Pramudita Satria Palar, Muhammad Daffa Robani, Nicolas Verstaevel, Moncef Garouani, Julien Aligon, Benoit Gaudou, Koji Shimoyama, and Joseph Morlier. Surrogate modeling and explainable artificial intelligence for complex systems : A workflow for automated simulation exploration. *arXiv preprint*, 2025.

[21] Erich Schubert, Remigius Wojdanowski, Arthur Zimek, and Hans-Peter Kriegel. On evaluation of outlier rankings and outlier scores. In *International conference on data mining*, pages 1047–1058. SIAM, 2012.

[22] David H Wolpert and William G Macready. No free lunch theorems for optimization. *Transactions on evolutionary computation*, 2002.

[23] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod : A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96) :1–7, 2019.

A Surrogate Policy Model for Auditing Black-Box Recommendation Systems: Application to Change Detection

Marouane Bazzaoui[1], Matthieu Jonckheere[1], Erwan Le Merrer[2], Gilles Trédan[1]

[1] LAAS, CNRS, Toulouse, France
[2] Inria, Rennes, France

marouane.bazzaoui@laas.fr

Abstract

Recommender systems increasingly shape information exposure. As a result, auditing them has become a growing necessity. A key challenge is to understand what can be inferred about a recommender's behaviour from black-box observations alone, i.e., without access to its internals. In this paper, we propose a method to audit recommender systems using a surrogate policy model. This surrogate policy estimator provides a local approximation of the recommender system's behaviour with a characterized approximation error. We establish the consistency and asymptotic normality of this estimator, enabling hypothesis testing. We then propose a change detection task for assessing whether or not the recommender has updated its behaviour.

Keywords

Auditing, recommender systems, surrogate models, inverse reinforcement learning.

1 Introduction

Recommendation systems play a crucial filter role in modern information flows. This role can have a dramatic impact on society [11, 14, 15, 5, 1]. Emerging regulatory frameworks, such as the EU Digital Services Act, increasingly require that such systems be subject to external scrutiny.
Auditing a recommendation system, however, is considerably more challenging than auditing a standard classifier. The space of possible items' states is huge [6]. The content evolves through complex and unknown dynamics, and the items' features are difficult to assess. Moreover, the memory effect from personalization induces a massive trajectory space impossible to explore exhaustively. Finally, external parties typically do not have access to the model's internals.
To address these challenges and provide formal guarantees, we adapt an inverse reinforcement learning (IRL) approach to the recommender auditing problem. In classifier auditing and explainability, the notion of a surrogate model is central. We here explore the construction of a surrogate recommender. We follow a two-step approach : 1) construct a local approximation (a surrogate) of the target model, and 2) conduct tests on the surrogate to obtain an audit decision. See Figure 1 for an overview of the process.

In order to illustrate a potential application of our surrogate construction approach, we consider the *change detection* problem. Change detection is arguably the simplest audit task : observing a black-box model at two distinct time intervals and deciding whether the underlying recommendation mechanism has changed. This task is particularly important for recommender systems, whose decisions meaningfully shape exposure. Detecting changes is especially crucial for sensitive topics, where a shift can signal a change in the system's neutrality or bias across competing viewpoints.
To sum up, this paper provides the following contributions :

- We consider the recommendation system a black box and we formalize a surrogate model. We show that it converges consistently to an approximation of the recommender's behaviour under a few assumptions. We show that the surrogate estimator is asymptotically normally distributed.
- We demonstrate the value of the surrogate estimator and its asymptotic normality via its application to the change detection problem. We establish a statistical test for detecting changes in the black-box model.
- We conduct experiments in a controlled environment to evaluate the effectiveness of our approach in the change detection task.

2 Previous Work

Auditing recommender systems has been an active research area for over a decade. Earlier works frame auditing from a security perspective : a (passive) recommender may be subjected to an external attack, which the audit aims to detect. Several studies have been conducted on detecting manipulations and shilling attacks in the recommendation systems context [8, 7]. While these works introduce methods to detect changes to the recommender's behaviour, they are focused only on item-level anomalies.
Explaining the decisions of a recommender from a black box perspective through local surrogate models is an approach explored in recent works such as LIME-RS [13], LIRE [4], the LIME-RS adherence and constancy study [2]. In [19] a model-agnostic framework was introduced to produce faithful post-hoc explanations. Some works ex-

plain why a particular item was recommended, *e.g.*, through counterfactual explanation methods [3]. Others focus on the recommender's broader behaviour, *e.g.*, through counterfactual audit methods [9]. Overall, this line of work is primarily focused on explainability. Compared to these works, our adapted IRL-based approach comes with formal guarantees in a favorable setting.

3 Problem Formulation

3.1 Black-Box Interaction Model

We model the target recommender as a black box that recommends items to a user from a corpus of n items indexed by $i \in \{1, 2, \dots n\}$. We consider discrete recommendation steps $t \in \{1, 2, \dots T\}$.

Each item is modeled as a discrete-time Markov chain whose state evolves at each step (whether it got recommended or not). This state represents the item features used for recommendation, for instance : number of views, age, or the intrinsic toxicity of the item. At each step, we denote the state of item $i \in \{1, 2, \dots, n\}$ by $S_i \in \mathcal{S}$ and the corpus' state vector by $\mathcal{C} = (S_1, S_2, \dots, S_n)$; the recommender then selects an action vector $A \in \{0, 1\}^n$, where $A_i = 1$ indicates that item i is recommended and $A_i = 0$ otherwise. We assume that exactly m items are recommended at each step. For readability, we omit the time index t throughout, all quantities S_i, $\mathcal{C}$, and A should be understood as referring to a generic time step.

In the reinforcement learning literature, this setting is commonly known as a restless multi-armed bandit, in which the recommender follows a policy π that, at each step, selects which m items to recommend.

In addition, we assume that the observable information of the black-box model is restricted to (i) a representation of the items' states in the form of a features matrix $\Phi(\mathcal{C}) \in \mathbb{R}^{n \times d}$ whose i-th row is the feature vector $\phi_i(S_i) \in \mathbb{R}^d$ of item i; and (ii) the agent's action $A \in \{0, 1\}^n$.

The model is observed at a time interval, during which we observe N distinct trajectories, of length T. We denote the trajectories by

$$\mathcal{T}_k := (\Phi_{k,t}, A_{k,t})_{t=1}^{T}, \quad k \in \{1, \dots N\}.$$

We therefore define the trajectories' set

$$\mathbb{T} = \{\mathcal{T}_k\}_{k=1}^{N}.$$

We write

$$\mathcal{T}_k \sim P_T$$

for some law P_T on $\left(\mathbb{R}^{n \times d} \times \{0, 1\}^n\right)^T$.

3.2 Surrogate Policy Model

In a recommender system, the decision to recommend an item depends not only on the item itself, but also on the other items in the corpus. Externally auditing such a system becomes challenging when considering all factors at play. To overcome this, we introduce a parametric surrogate model that bases its decisions on a scoring function, assigning higher recommendation probabilities to items with higher scores. Our approach consists in transferring the essential decision logic of the recommender to a simple, interpretable surrogate model with well-characterized asymptotic behaviour that enables formal hypothesis testing.

We introduce $I : \mathcal{S} \to \mathbb{R}$ an index function defined as a function that scores an item depending entirely on its state $S \in \mathcal{S}$. Let X_I denote an index vector, and π_I denote a soft-index-policy, such that

$$X_I(\mathcal{C}) := (0, I(S_2) - I(S_1), \dots I(S_n) - I(S_1))$$

$$\text{and} \quad \pi_I = \mathrm{st}_m(X_I)$$

where $\mathcal{C} = (S_1, \dots S_n)$, and $\mathrm{st}_m : \mathbb{R}^n \to (0, 1)^n$ denotes the soft-top-m function, a smooth approximation of the standard top-m operator. It is derived from the optimal plan of an entropic transport problem [18, 10].

By restricting the index function I to a parametric family of functions, we can define a parametric surrogate model based on the soft index policy π_I. By defining the surrogate based on a smooth approximation rather than the top-m operator itself, the model's output becomes differentiable. This enables gradient-based optimization and asymptotic analysis. Because the surrogate policy is built on the soft-top-m function [18], it is non-deterministic, and it recommends m items in expectation.

In this work, inspired by LIME [16], we consider linear models for their efficiency as well as their intrinsic interpretability. We restrict the index I to be linear in $\phi_S \in \mathbb{R}^d$, where ϕ_S denotes the features of an item whose state is $S \in \mathcal{S}$, namely

$$I_\theta(S) = \theta^\top \phi_S.$$

Hence, rather than considering an arbitrary index I, we restrict ourselves to the linear family induced by this parametrization.

Let $X_\theta = X_{\hat{I}_\theta}$, the linear surrogate model is then defined by

$$\hat{\pi}_\theta = \mathrm{st}_m(X_\theta).$$

The purpose of the surrogate model is to approximate the behaviour of the recommender's policy. A good surrogate should therefore reproduce the decision patterns of the black-box model with a controlled approximation error, while remaining simple enough to be interpretable.

4 Surrogate Policy Estimator

In order to approximate the recommender's policy, we use the maximum-likelihood approach. We first establish that the surrogate model class can asymptotically reproduce any index-policy. We then introduce an empirical estimator used to fit the surrogate from data, and we establish its consistency and asymptotic normality.

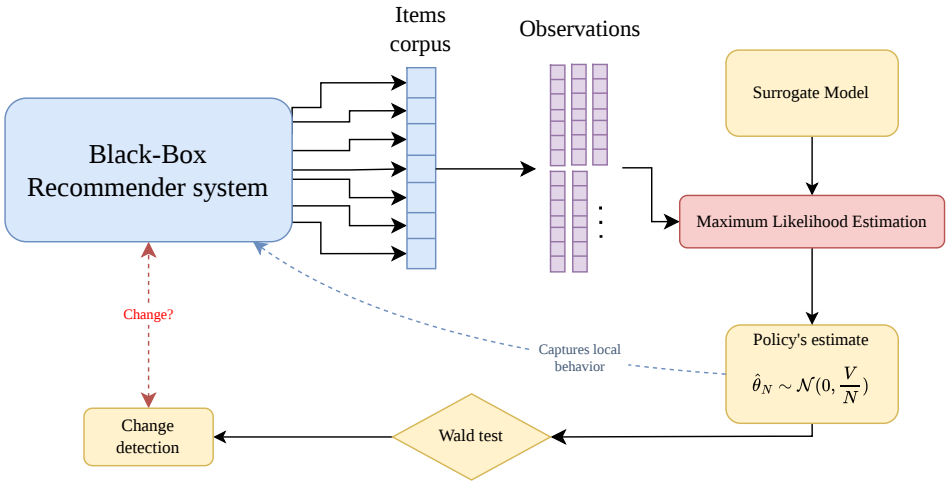


FIGURE 1 – Overview of the surrogate model auditing framework with illustrative downstream application.

4.1 Surrogate Model Asymptotic Compatibility

Under the assumption that the recommendation system is optimized with respect to some unknown objective, its policy π is deterministic, since any unconstrained MDP admits a deterministic optimal policy. We also assume that the recommender policy π admits an index representation, *i.e.*,

$$\exists I \in \mathcal{F}(\mathcal{S}, \mathbb{R}), \;\; \forall \mathcal{C} \in \mathcal{S}^n \;\; ; \;\; \pi(\mathcal{C}) = \underset{i}{\text{top-m}} \; I(S_i)$$

where $I : \mathcal{S} \to \mathbb{R}$ denotes an index, and $\mathcal{F}(\mathcal{S}, \mathbb{R})$ denotes the set of functions from $\mathcal{S}$ to $\mathbb{R}$.
We refer to any policy π that admits an index representation as an index-policy. We define the set of all index-policies Π as follows

$$\Pi = \left\{ \pi \;\middle|\; \begin{array}{l} \pi \in \mathcal{F}(\mathcal{S}^n, \{0,1\}^n), \\ \exists I \in \mathcal{F}(\mathcal{S}, \mathbb{R}) \; ; \;\; \pi(\mathcal{C}) = \text{top-m}_i \; I(S_i) \end{array} \right\}.$$

where $\mathcal{F}(\mathcal{S}^n, \{0,1\}^n)$ denotes the set of functions from $\mathcal{S}^n$ to $\{0,1\}^n$, and $\mathcal{F}(\mathcal{S}, \mathbb{R})$ denotes the set of functions from $\mathcal{S}$ to $\mathbb{R}$.

Theorem 1. *Let π denote a recommender index-policy and I its index, π_I denote a soft-index-policy with the same index I as the recommender, $\mathcal{C}$ denote the corpus' state vector, and $X_I(\mathcal{C}) = (0, I(S_2) - I(S_1), \dots I(S_n) - I(S_1))$ denote the index vector. Then*

$$\|\pi(\mathcal{C}) - \pi_I(\mathcal{C})\|_2 \leq \frac{\varepsilon(\ln n + \ln 2)}{X_I(S_{\sigma_{m+1}}) - X_I(S_{\sigma_m})}$$

where $\epsilon > 0$ is a parameter of the soft-top-m function [18], and σ the sorting permutation of the vector X_I such that $X_I(S_{\sigma_m})$ and $X_I(S_{\sigma_{m+1}})$ are the m-th and $(m+1)$-th highest values in the vector X_I.

The following theorem establishes that the surrogate model, despite using a smooth relaxation, can approximate any index-policy arbitrarily well as the index values grow large.

Theorem 2. *Let Π be the set of all index policies. Then for $\pi \in \Pi$ an arbitrary index-policy, there exist a sequence of indexes $(I_k) \subset \mathcal{F}(\mathcal{S}, \mathbb{R})$ such that*

$$\forall \mathcal{C} \in \mathcal{S}^n \;\; \pi_{I_k}(\mathcal{C}) \underset{k}{\to} \pi(\mathcal{C})$$

$$\textit{and} \quad \forall \mathcal{C} \in \mathcal{S}^n \;\; \|I_k(\mathcal{C})\| \underset{k}{\to} +\infty$$

where $\pi_{I_k} = \text{st}_m(X_{I_k})$ a soft-index-policy, and $I_k(\mathcal{C})$ denotes a vector such that $I_k(\mathcal{C}) = (I_k(S_1), \dots I_k(S_n))$.

Theorem 1 establishes the approximation error of the surrogate policy model.
Theorem 2 implies that for any given recommender index policy, there exists a sequence of surrogate policies (I_k) realising an asymptotic approximation.

4.2 Maximum Likelihood Estimator

In this section, we define the surrogate policy estimator. We proceed as follows : first, we formalize the population's negative log-Likelihood risk, then introduce an L_2 penalty, derive its empirical counterpart, and finally define the resulting surrogate empirical estimator.
In order to have a well-defined population risk, we assume that the second moments of the feature matrices Φ_t in trajectories $\mathcal{T}$ are finite, *i.e.*, $\mathbb{E}\|\Phi_t\|_2^2 < \infty$ for all $t \in \{1, \dots T\}$. This condition prevents feature values from taking excessively large values too often, and is trivially satisfied when features are bounded.
Let $\hat{\pi}_\theta$ denote the policy defined by the surrogate model parameterized by θ. We introduce the population risk L as follows

$$L(\theta) := \mathbb{E}_{\mathcal{T}\sim P_T}\left[\sum_{(\Phi,A)\in\mathcal{T}} -\log P_\theta(A \mid \Phi)\right]$$

where $P_\theta(A \mid \Phi)$ denotes the conditional probability of the action A being played by $\hat{\pi}_\theta$ knowing Φ.
As established in Theorem 2, the minimizer of L lies on the infinite boundaries of the parameter space $\theta \in \mathbb{R}^d$, which motivates penalizing the risk. Let L^λ denote the penalized population criterion, defined by

$$L^\lambda(\theta) := L(\theta) + \lambda\|\theta\|_2^2$$

where $\lambda > 0$ denotes the regularization parameter.
The regularized population criterion L^λ is twice differentiable and strongly convex in $\theta \in \mathbb{R}^d$ and therefore admits a unique minimizer $\theta^* \in \mathbb{R}^d$,

$$\theta^* := \underset{\theta\in\mathbb{R}^d}{\arg\min}\ L^\lambda(\theta).$$

Let $\hat{L}_N^\lambda$ denote the empirical counterpart of the penalized population criterion $L^\lambda(\theta)$ on a sample of trajectories $\mathbb{T} = (\mathcal{T}_i)_{i=1}^N$. We assume the independence of these trajectories

$$\mathcal{T}_1, \mathcal{T}_2, \dots \mathcal{T}_N \overset{\text{iid}}{\sim} P_T.$$

We denote the empirical target as $\hat{\theta}_N$, then

$$\hat{\theta}_N := \underset{\theta\in\mathbb{R}^d}{\arg\min}\ \hat{L}_N^\lambda(\theta).$$

In this work, we focus primarily on the penalized estimator since the unpenalized criterion may fail to admit a finite minimizer. Therefore, from this point onward, we implicitly consider the penalized version of quantities (criterion, target) unless stated otherwise.

4.3 Asymptotic Normality

We now explicitly characterize the convergence of our surrogate estimator. First, we establish in Theorem 3 the consistency of the estimator. We then derive the asymptotic distribution using a standard Taylor expansion argument. We include key steps of the derivation that introduce the Hessian H and the covariance Σ of the score vector. Defining these two terms is essential because they appear in the asymptotic normal distribution result.

Theorem 3. *Let θ^* denote the population target parameter of the surrogate policy estimator, and let $\hat{\theta}_N$ denote the empirical counterpart from a sample of N trajectories. Then*

$$\hat{\theta}_N \xrightarrow[N\to+\infty]{p} \theta^*.$$

*Implying that the surrogate policy estimator is **consistent**.*

We write the empirical estimator as

$$\hat{\theta}_N = \underset{\theta\in\mathbb{R}^d}{\arg\min}\ \hat{L}_N^\lambda(\theta), \quad \hat{L}_N^\lambda(\theta) = \frac{1}{N}\sum_{k=1}^N l^\lambda(\mathcal{T}_k, \theta)$$

where $l^\lambda(\mathcal{T}_k, \theta)$ denotes the instantaneous loss at the trajectory $\mathcal{T}_k$, and defined by

$$l^\lambda(\mathcal{T}_k, \theta) = \left[\sum_{(\Phi,A)\in\mathcal{T}} -\log P_\theta(A \mid \Phi)\right] + \lambda\|\theta\|_2^2.$$

Let $\Psi_N : \mathbb{R}^d \to \mathbb{R}^d$ denote the score function defined by

$$\Psi_N(\theta) = \nabla_\theta \hat{L}_N^\lambda(\theta).$$

Since $\hat{L}_N^\lambda$ is twice differentiable, the score function Ψ_N and its Jacobian $\nabla\Psi_N$ are well defined on $\mathbb{R}^d$.
Using the fact that $\Psi_N(\hat{\theta}_N) = 0$ together with a Taylor expansion of Ψ_N around θ^*, we get

$$\sqrt{N}\left(\hat{\theta}_N - \theta^*\right) = -\left(\nabla\Psi_N(\tilde{\theta}_N)\right)^{-1}\sqrt{N}\Psi_N(\theta^*) \quad (1)$$

where $\tilde{\theta}_N \in \mathbb{R}^d$ lies between $\hat{\theta}_N$ and θ^*, *i.e.*, $\tilde{\theta}_N$ is a convex combination of $\hat{\theta}_N$ and θ^*.
Because the Hessian of $\hat{L}_N^\lambda$ is positive definite, and therefore invertible, the matrix $\left(\nabla\Psi_N(\tilde{\theta}_N)\right)^{-1}$ is well defined.
Let $H \in \mathbb{R}^{d\times d}$ denote the Hessian of L^λ at θ^*, then by LLN we get

$$\left(\nabla\Psi_N(\tilde{\theta}_N)\right)^{-1} \xrightarrow[N\to+\infty]{p} H^{-1}. \quad (2)$$

Computational complexity. The Hessian H can be computed in $O(nd^2)$ time and $O(nd)$ memory, *i.e.*, linearly in the catalog size n.
Let $\Sigma \in \mathbb{R}^{d\times d}$ be the covariance matrix of $\nabla l^\lambda(\mathcal{T}, \theta^*)$, then by CLT we get

$$\sqrt{N}\Psi_N(\theta^*) \xrightarrow[N\to+\infty]{d} \mathcal{N}(0, \Sigma). \quad (3)$$

Theorem 4. *Let θ^* denote the population target parameter of the surrogate policy estimator, and let $\hat{\theta}_N$ denote the empirical counterpart from a sample of N trajectories. Using (1), (2), and (3) together with Slutsky's theorem, we get*

$$\sqrt{N}\left(\hat{\theta}_N - \theta^*\right) \xrightarrow[N\to+\infty]{d} \mathcal{N}(0, H^{-1}\Sigma H^{-1})$$

where $\Sigma = \mathrm{Var}(\nabla l^\lambda(\mathcal{T}, \theta^*))$ and $H = \nabla^2 L^\lambda(\theta^*)$.

5 Change Detection Problem

Although surrogate policy estimation is our central contribution, its value is best demonstrated through a concrete downstream task. We use change detection as such a demonstration.
The model is observed at two non-overlapping episodes indexed by $e \in \{1, 2\}$. During both episodes, we observe N distinct trajectories, each of length T, each observed in a different corpus of items. In this setting, we test whether the policy of the agent changed between episode 1 and episode 2 or remained unchanged.

Let π_1 and π_2 be the agent's policies at episode 1 and episode 2, respectively. The change detection problem consists in deciding whether π_1 is different from π_2 based only on observable information : $\mathbb{T}_1$ and $\mathbb{T}_2$. Formalized as a hypothesis test, it can be written as

$$H_0' : \pi_1 = \pi_2, \quad H_1' : \pi_1 \neq \pi_2.$$

Our approach to this problem is to use the surrogate policy model, to estimate the agent's policies in the two episodes, then compare the two resulting policy estimates. Let θ_e^* be the population target parameter vector of the surrogate policy estimator for episode e. We define the following hypothesis test

$$H_0 : \theta_1^* = \theta_2^*, \quad H_1 : \theta_1^* \neq \theta_2^*.$$

Theorem 5. *Let θ_e^* be the population target parameter vector of the surrogate policy estimator for episode e. Let $\pi_e = \text{top-m}(X_e)$ be the true agent's policy at episode e, and X_e the underlying index vector. Then*

$$\pi_1 = \pi_2 \implies \theta_1^* = \theta_2^*.$$

By Theorem 5, we obtain $H_0' \implies H_0$. Therefore, if we refute H_0 we can refute the null hypothesis H_0', and consequently detect a change in the agent's policy.

5.1 Wald Test

Let $\hat{\Delta} := \hat{\theta}_2 - \hat{\theta}_1$ denote the difference in the estimators between episode 1 and episode 2. Then $\hat{\Delta}$ is asymptotically normal, and under H_0,

$$\hat{\Delta} \sim \mathcal{N}\left(0, 2\,\frac{V}{N}\right)$$

where N denotes the number of trajectories per episode. and $V = H^{-1}\Sigma H^{-1}$ denotes the asymptotic variance common to both episodes under H_0.
Let $\hat{V}_e$ denote a consistent estimator of the asymptotic variance at episode e. Let W denote the Wald test statistic defined by

$$W := \hat{\Delta}^\top \left(\frac{\hat{V}_1 + \hat{V}_2}{N}\right)^{-1} \hat{\Delta}.$$

Under H_0,

$$W \xrightarrow{d} \chi_d^2$$

where d denotes the dimension of the parameters' space. Therefore, for a given significance level $\alpha \in (0, 1)$, we reject the null-hypotheses H_0, and consequently $H_0' : \pi_1 = \pi_2$, when

$$W > q_{1-\alpha}$$

where $q_{1-\alpha}$ denotes the $(1-\alpha)$-quantile of the χ_d^2 distribution.

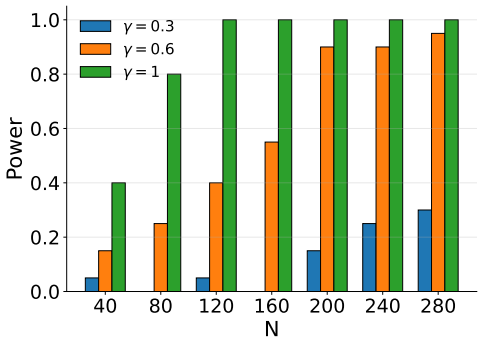


FIGURE 2 – Empirical power $(1-\beta)$ as a function of the sample size N for three separation values γ.

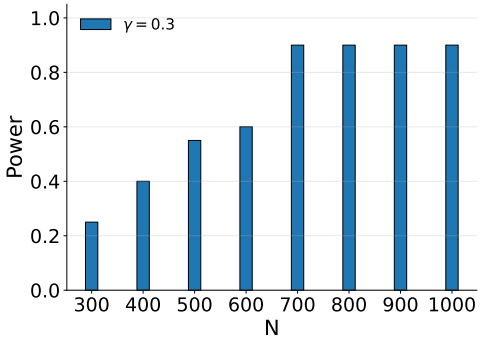


FIGURE 3 – Empirical power $(1-\beta)$ as a function of the sample size N for a small separation value $\gamma = 0.3$.

6 Experimental Study

While previous results characterize the asymptotic correctness of our approach, to be useful in practice an estimator must provide accurate results in the finite-sample regime. To that end, we evaluate change detection on synthetic data using a controlled simulated environment, which allows us to test our method's performance across a range of settings.

Experimental setup. We simulate the behaviour of a recommender by a linear index-policy. We observe its decisions across different sets of items, each of size $n = 10$. Its interactions with each set of items yield a trajectory over $T = 5$ consecutive recommendation steps. Each item is characterized by three features : popularity, toxicity, and profitability. The popularity value evolves in time while toxicity and profitability stay unchanged, but all three are randomly initialized. We conduct the change detection test between two generated trajectory sets, one by a *toxic* policy, and the other by a *neutral* policy ; both use the same weights for popularity and profitability, $\theta_{pop} = 10$ and $\theta_{prf} = 5$. The toxic policy additionally assigns a positive weight $\theta_{tox} = \gamma$ to toxicity, whereas the neutral policy assigns a null weight to toxicity $\theta_{tox} = 0$. Both simulated policies prefer recommending popular and profitable items, but the *toxic* recommender has systematic preferences for toxic items, while the *neutral* recommender is indifferent

to toxicity.
In this setup, the change detection hypothesis test, introduced in Section 5, is formulated as

$$H_0 : \theta^*_{tox} = \theta^*_{neut} \ , \quad H_1 : \theta^*_{tox} \neq \theta^*_{neut}$$

where θ^*_{tox} and θ^*_{neut} denote the surrogate model's estimations of the weights vector of the *toxic* recommender and the *neutral* recommender, respectively. We set the confidence level in our experiments at $(1 - \alpha) = 95\%$.
The primary question in this section is the trade-off between (i) cost, *i.e.*, the number of observed trajectories required to detect a change ; (ii) separation, *i.e.*, the distance between the two tested policies ; (iii) reliability, *i.e.*, the ability to detect a change when one occurs. We quantify the trade-off between these three axes by the following metrics :

- **Cost :** N, the number of trajectories generated under each policy.
- **Separation :**γ, the difference in the toxicity weights between the two policies.
- **Reliability :** $1 - \beta$ (power), the probability of detecting a change when one occurs.

The estimates are obtained via a gradient descent algorithm by minimizing the empirical criterion.
We vary the cost and separation parameters in a grid where N takes the values $\{40, 80, 120, 160, 200, 240, 280\}$, and γ takes the values $\{0.3, 0.6, 1.0\}$. We repeat the test 20 times for each pair (N, γ), and we measure the empirical power $1 - \beta$. See Figure 2.
In order to investigate the trade-off between the cost and the reliability in the task of detecting a small change $\gamma = 0.3$, we evaluate the power $1 - \beta$ at a bigger scale of $N \in \{300, 400, 500, 600, 700, 800, 900, 1000\}$. See Figure 3.

7 Discussion

Limitations of the modeling. While our black-box modeling choices simplify reality, we argue that they remain reasonably aligned with it. First, assuming that the recommender's policy is index-based is not overly restrictive, since in this type of sequential allocation problem the optimal policy admits an index-based representation under mild conditions [17]. Second, a linear surrogate may not adequately represent abrupt policy shifts or strongly non-linear dynamics ; extending the framework to non-linear surrogate models lies outside the scope of this paper. Finally, the i.i.d. trajectory assumption can be justified in practice by collecting each trajectory from a different item set, or by separating trajectories sufficiently to eliminate temporal dependence.

Feature observability. The items' states in the present work are represented by features that we assume capture all the information relevant to the recommender. However, it is usually impossible for an external party to observe that representation in its entirety. In practice, auditors can observe a subset of the features. In that case, a change in the recommender's preferences with respect to the observed features can be detected, while a change orthogonal to the observed information is invisible to the test. Moreover, unobserved features that correlate with observed ones can cause omitted-variable bias in our estimates. The estimated coefficients thus represent the recommender's behavior projected onto the observed features space, which is sufficient for auditing a change, though not for causal interpretation of the features' effects.

Beyond change detection. In the present work, we construct a surrogate model that locally approximates a recommender's policy and that has statistical characteristics that facilitate rigorous analysis. While we focus on the change detection task in this work, the proposed framework addresses a broader range of questions that we leave for future work : What is the system optimizing ? Are there viewpoint biases ? Can the system be manipulated ?

8 Conclusion

We proposed a framework for auditing black-box recommender systems using a surrogate policy model. We derived a simple surrogate policy estimator that provides a local approximation of the recommender system's behaviour with a characterized approximation error. We established consistency and asymptotic normality of the surrogate policy estimator, enabling hypothesis testing and further analysis of the local approximation. We demonstrated the auditing capability of the framework through a change detection task, both theoretically and experimentally in a controlled simulation environment.
This work opens several directions for future research. Applying the change detection method to real-world recommender systems is an important next step. Moreover, the proposed surrogate policy model can serve as a foundation for a broader range of auditing tasks, such as bias identification and fairness assessment.

References

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv :1907.13286*, 2019.

[2] Vito Walter Anelli, Alejandro Bellogín, Tommaso Di Noia, Francesco Maria Donini, Vincenzo Paparella, and Claudio Pomo. Adherence and constancy in lime-rs explanations for recommendation. *arXiv preprint arXiv :2109.00818*, 2021.

[3] Oren Barkan, Veronika Bogina, Liya Gurevitch, Yuval Asher, and Noam Koenigstein. A counterfactual framework for learning and evaluating explanations for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3723–3733, 2024.

[4] Léo Brunot, Nicolas Canovas, Alexandre Chanson, Nicolas Labroche, and Willème Verdeaux. Preference-based and local post-hoc explanations for recommender systems. *Information Systems*, 108 :102021, 2022.

[5] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and

decreases utility. In *Proceedings of the 12th ACM conference on recommender systems*, pages 224–232, 2018.

[6] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[7] Min Gao, Renli Tian, Junhao Wen, Qingyu Xiong, Bin Ling, and Linda Yang. Item anomaly detection based on dynamic partition for time series in recommender systems. *PloS one*, 10(8) :e0135155, 2015.

[8] Min Gao, Quan Yuan, Bin Ling, and Qingyu Xiong. Detection of abnormal item based on time intervals for recommender systems. *The Scientific World Journal*, 2014(1) :845897, 2014.

[9] Homa Hosseinmardi, Amir Ghasemian, Miguel Rivera-Lanas, Manoel Horta Ribeiro, Robert West, and Duncan J Watts. Causally estimating the effect of youtube's recommender system using counterfactual bots. *Proceedings of the national academy of sciences*, 121(8) :e2313377121, 2024.

[10] Gauri Jain, Pradeep Varakantham, Haifeng Xu, Aparna Taneja, Prashant Doshi, and Milind Tambe. Irl for restless multi-armed bandits with applications in maternal and child health. In *Pacific Rim International Conference on Artificial Intelligence*, pages 165–178. Springer, 2024.

[11] Erwan Le Merrer, Gilles Trédan, and Ali Yesilkanat. Modeling rabbit-holes on youtube. *Social network analysis and mining*, 13(1) :100, 2023.

[12] Whitney K Newey and Daniel McFadden. Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4 :2111–2245, 1994.

[13] Caio Nóbrega and Leandro Marinho. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing*, pages 1671–1678, 2019.

[14] Derek O'Callaghan, Derek Greene, Maura Conway, Joe Carthy, and Pádraig Cunningham. Down the (white) rabbit hole : The extreme right and online recommender systems. *Social Science Computer Review*, 33(4) :459–478, 2015.

[15] Manoel Horta Ribeiro, Raphael Ottoni, Robert West, Virgílio AF Almeida, and Wagner Meira Jr. Auditing radicalization pathways on youtube. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 131–141, 2020.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[17] Peter Whittle. Restless bandits : Activity allocation in a changing world. *Journal of applied probability*, 25(A) :287–298, 1988.

[18] Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. *Advances in neural information processing systems*, 33 :20520–20531, 2020.

[19] Zhichao Xu, Hansi Zeng, Juntao Tan, Zuohui Fu, Yongfeng Zhang, and Qingyao Ai. A reusable model-agnostic framework for faithfully explainable recommendation and system scrutability. *ACM Transactions on Information Systems*, 42(1) :1–29, 2023.

A Proofs

Proof of Theorem 1. Let π denote a recommender index-policy and I its index, π_I denote a soft-index-policy with the same index I as the recommender, $\mathcal{C}$ denote the vector of the items' state, and $X_I(\mathcal{C}) = (0, I(S_2) - I(S_1), \dots I(S_n) - I(S_1))$ denote the index vector.
We show that

$$\|\pi(\mathcal{C}) - \pi_I(\mathcal{C})\|_2 \leq \frac{\varepsilon(\ln n + \ln 2)}{X_I(S_{\sigma_{m+1}}) - X_I(S_{\sigma_m})}$$

where $\epsilon > 0$ is a parameter of the soft-top-m function [18], and σ the sorting permutation of the vector X_I.
For simplicity we denote $X = X_I(\mathcal{C})$
The concerned theorem, is a direct consequence of a result in [18] (Theorem 2). It states the following

$$\|\Gamma_{soft}(X) - \Gamma(X)\|_F \leq \frac{\varepsilon(\ln n + \ln 2)}{n\left(X_I(S_{\sigma_{m+1}}) - X_I(S_{\sigma_m})\right)} \tag{1}$$

where Γ_{soft} and Γ are defined by (we suppress the X notation for simplicity)

$$\Gamma_{soft} = \arg\min_{\Gamma_{soft}} \langle \Gamma_{soft}, C\rangle + \epsilon H(\Gamma_{soft}),$$
$$\text{s.t } \Gamma_{soft} 1_2 = u,\ \Gamma_{soft} 1_n = v$$

and

$$\Gamma = \arg\min_{\Gamma} \langle \Gamma, C\rangle,$$
$$\text{s.t } \Gamma 1_2 = u,\ \Gamma 1_n = v$$

where $\epsilon > 0$, $u = (\frac{1}{n}, \dots \frac{1}{n}) \in \mathbb{R}^n$, $v = (\frac{n-k}{n}, \frac{k}{n})$, and $C \in \mathbb{R}^{n\times 2}$ such that $C_{i1} = x_i^2$ and $C_{i2} = (x_i - 1)^2$.
By [18] we have

$$\pi(\mathcal{C}) = \text{top-m}(X) = n\Gamma(X)\ [0,1]^\top$$

and

$$\pi_I(\mathcal{C}) = \text{st}_m(X) = n\Gamma_{soft}(X)\ [0,1]^\top$$

Then

$$\|\pi_I(\mathcal{C}) - \pi(\mathcal{C})\|_2 \leq n\,\|\Gamma_{soft}(X) - \Gamma(X)\|_F \tag{2}$$

By (1) and (2) we conclude

$$\|\pi_I(\mathcal{C}) - \pi(\mathcal{C})\|_2 \leq \frac{\varepsilon(\ln n + \ln 2)}{X_I(S_{\sigma_{m+1}}) - X_I(S_{\sigma_m})}$$

□

Proof of Theorem 2. We show that for an arbitrary index-policy $\pi \in \Pi$ the following

$$\exists (I_k) \subset \mathcal{F}(\mathcal{S}, \mathbb{R})\ \ \forall \mathcal{C} \in \mathcal{S}^n\ \ \pi_{I_k}(\mathcal{C}) \underset{k}{\to} \pi(\mathcal{C})$$

where $\pi_{I_k} = \text{st}_m(X_{I_k})$, and $I_k(\mathcal{C})$ denotes a vector such that $I_k(\mathcal{C}) = (I_k(S_1), \dots I_k(S_n))$.
For simplification let $X_{I_k} = X_k$.
the policy π admits an index-based structure, let I be its index. Take a sequence of indexes $I_k = kI$ defined as the recommender's index times k. Since X_k is defined by

$$X_k = (0, I_k(S_2) - I_k(S_1), \dots I_k(S_n) - I_k(S_1))$$

then $X_k = kX$ where

$$X = (0, I(S_2) - I(S_1), \dots I(S_n) - I(S_1))$$

The recommender's policy $\pi = \text{top-m}\, X$ depends only on the ranking of X, then it can also be written as $\pi = \text{top-m}\, kX$.
Then we can consider π and π_{I_k} to have the same index I_k.
Hence, by Theorem 1, we have

$$\|\pi_{I_k}(\mathcal{C}) - \pi(\mathcal{C})\|_2 \leq \frac{\varepsilon(\ln n + \ln 2)}{kX(S_{\sigma_{m+1}}) - kX(S_{\sigma_m})}$$

where σ denotes a sorting permutation of X.
We have

$$\frac{\varepsilon(\ln n + \ln 2)}{k\left(X(S_{\sigma_{m+1}}) - X(S_{\sigma_m})\right)} \underset{k}{\to} 0$$

Then

$$\pi_{I_k}(\mathcal{C}) \underset{k}{\to} \pi(\mathcal{C})$$

Now we show that if π_{I_k} converges to π, then

$$\forall \mathcal{C} \in \mathcal{S}^n\ \ \|I_k\| \to \infty.$$

We assume that $I_k(\mathcal{C})$ does not diverge to infinity and we show a contradiction.
Since $I_k(\mathcal{C})$ does not diverge to infinity, it is bounded and By Bolzano–Weierstrass, it has a further sub-sequence

$$I_{k_j}(\mathcal{C}) \to J \in \mathbb{R}^n.$$

Because st_m is continuous : $\pi_{I_{k_j}}(\mathcal{C}) \to \pi_J(\mathcal{C})$.
Hence $\pi(\mathcal{C}) = \pi_J(\mathcal{C})$, a contradiction because $\pi(\mathcal{C})$ is a deterministic policy and $\pi_J(\mathcal{C})$ is a non-deterministic policy knowing J finite.

□

Proof of Theorem 3. Let θ^* denote the population target parameter of the surrogate policy estimator, and let $\hat{\theta}_N$ denote the empirical counterpart based on a sample of N trajectories. We show that the estimator is consistent, that is,

$$\hat{\theta}_N \underset{N\to\infty}{\xrightarrow{p}} \theta^*.$$

We begin by formulating the gradient and the Hessian of one component of the instantaneous loss (at one state-action pair instead of the whole trajectory). For simplicity, we denote this component by l, without explicitly writing the state or time indices. We also denote by π the value of the

policy at a given state, viewed as a vector, and by π_i its coordinates. Let X denote the index vector, and let x_i be its i-th coordinate. We have

$$\pi = \mathrm{st}_m(X).$$

Let $\bar{R}$ and R denote a partition of $\{1, 2, \ldots, n\}$, where R is the set of recommended indices and $\bar{R}$ is the set of non-recommended ones. Then

$$l = -\sum_{i \in R} \log(\pi_i) - \sum_{i \in \bar{R}} \log(1 - \pi_i).$$

The goal is to find ∇l and $\nabla^2 l$. Then, by linearity of all the criteria, we can derive any other gradient or Hessian. Since the soft-index policies are defined as the minimizers of entropic optimal transport problems [18], we know that they are twice differentiable. This implies that all the criteria based on such policies are also twice differentiable.
The KKT formula for the entropic OT problem in [18] (See Proof of Theorem 1)

$$\Gamma_{ij} = \exp\left(\frac{f_i + g_j - C_{ij}}{\epsilon}\right), \qquad \forall i, j,$$

where f and g are the Lagrange multipliers.
We consider a small perturbation :

$$\Gamma_{ij} \to \Gamma_{ij} + \delta\Gamma_{ij}, \qquad x_i \to x_i + \delta x_i,$$

$$f_i \to f_i + \delta f_i, \qquad g_j \to g_j + \delta g_j.$$

We then obtain

$$\delta\Gamma_{ij} = \Gamma_{ij}\,\frac{\delta f_i + \delta g_j - \delta C_{ij}}{\epsilon}, \qquad \forall i, j.$$

Since Γ has constant marginals, we have

$$\sum_i \delta\Gamma_{ij} = 0, \qquad \sum_j \delta\Gamma_{ij} = 0.$$

After some algebra, we obtain

$$\begin{cases} u_i\,\delta f_i + \sum_j \Gamma_{ij}\,\delta g_j = a_i, \\ \sum_i \Gamma_{ij}\,\delta f_i + v_j\,\delta g_j = b_j, \end{cases}$$

where u and v denote the marginals of Γ. The terms a_i and b_j are given by

$$a_i = 2\Gamma_{i1} x_i\,\delta x_i + 2\Gamma_{i2}(x_i - 1)\,\delta x_i,$$

$$b_1 = 2\sum_i \Gamma_{i1} x_i\,\delta x_i,$$

$$b_2 = 2\sum_i \Gamma_{i2}(x_i - 1)\,\delta x_i.$$

Knowing that $\pi_i = n\Gamma_{i2}$, and after some algebra, we get

$$\delta\pi_i = \frac{2}{\epsilon}\pi_i(1 - \pi_i)\left(\delta x_i - \frac{\sum_r \pi_r(1 - \pi_r)\delta x_r}{W}\right),$$

where

$$W = \sum_r \pi_r(1 - \pi_r).$$

Therefore,

$$\frac{\partial \pi_i}{\partial x_j} = \begin{cases} -\dfrac{2}{\epsilon W}(1 - \pi_i)\pi_j(1 - \pi_j), & \text{if } i \neq j, \\ \dfrac{2}{\epsilon}(1 - \pi_i)\left(1 - \dfrac{\pi_i(1 - \pi_i)}{W}\right), & \text{if } i = j. \end{cases}$$

Then, after some algebra, we get the gradient of l with respect to X :

$$\nabla_{X,i} l = \begin{cases} \dfrac{2}{\epsilon}(\pi_i - 1), & \text{if } i \in R, \\ \dfrac{2}{\epsilon}\pi_i, & \text{if } i \in \bar{R}. \end{cases}$$

Let $M = I_n - M'$, where I_n is the identity matrix and M' is the matrix whose entries are all zero except on the first column, which is filled with ones. Then, by definition,

$$X = M\,\Phi\,\theta,$$

where Φ denotes the feature matrix and θ denotes the parameter of the index.
Hence, the gradient of l with respect to the parameter θ is

$$\begin{aligned} \nabla l &= \Phi^\top M^\top \nabla_X l \\ &= \Phi^\top \nabla_X l. \end{aligned}$$

Using the expressions of $\nabla_X l$ and $\frac{\partial \pi_i}{\partial x_j}$, we obtain the Hessian $H^{(X)}$ of l with respect to X :

$$H^{(X)}_{ij} = \begin{cases} \dfrac{4}{\epsilon^2 W}\pi_i(1 - \pi_i)\big(W - \pi_i(1 - \pi_i)\big), & \text{if } i = j, \\ -\dfrac{4}{\epsilon^2 W}\pi_i(1 - \pi_i)\pi_j(1 - \pi_j), & \text{if } i \neq j. \end{cases}$$

We also obtain the Hessian $H^{(\theta)}$ of l with respect to θ :

$$H^{(\theta)} = \Phi^\top M^\top H^{(X)} M \Phi.$$

Notice that

$$H^{(X)}_{ii} = \sum_{j \neq i} \left|H^{(X)}_{ij}\right| \qquad \text{for every } i \in \{1, \ldots, n\}.$$

Hence, $H^{(X)}$ is positive semi-definite, and consequently $H^{(\theta)}$ is also positive semi-definite. By linearity of the expectation and of the summation operator, we conclude that the unpenalized risks $L(\theta)$ and $\hat{L}_N(\theta)$ are convex. By adding the ℓ_2-penalization term, the population criterion $L^\lambda(\theta)$ and its empirical counterpart $\hat{L}^\lambda_N(\theta)$ become 2λ-strongly convex (*i.e.*, the Hessian matrices of both terms are $\succcurlyeq 2\lambda I_d$ where I_d denotes the identity matrix). In particular, L^λ admits a unique minimizer θ^*.
Since the trajectories $T_1, \ldots, T_N$ are i.i.d., the law of large numbers gives, for any fixed $\theta \in \mathbb{R}^d$,

$$\hat{L}^\lambda_N(\theta) \xrightarrow[N \to \infty]{p} L^\lambda(\theta).$$

Since $\hat{L}_N^\lambda$ is convex, L^λ has the unique minimizer θ^*, and $\hat{L}_N^\lambda(\theta)$ converges pointwise to $L^\lambda(\theta)$ for every $\theta \in \mathbb{R}^d$, consistency follows from Newey and McFadden [12] (Theorem 2.7) :

$$\hat{\theta}_N \xrightarrow[N\to\infty]{p} \theta^*.$$

Hence, the surrogate policy estimator is consistent.

□

Proof of Theorem 4. Let θ^* denote the population target parameter of the surrogate policy estimator, and let $\hat{\theta}_N$ denote the empirical counterpart from a sample of N trajectories. We show that

$$\sqrt{N}\big(\hat{\theta}_N - \theta^*\big) \xrightarrow{d} \mathcal{N}\big(0,\, H^{-1}\Sigma H^{-1}\big)\,,$$

where $\Sigma = \mathrm{Var}\big(\nabla l^\lambda(T,\theta^*)\big)$ and $H = \nabla^2 L^\lambda(\theta^*)$.
We write

$$\hat{L}_N^\lambda(\theta) = \frac{1}{N}\sum_{k=1}^{N} l^\lambda(T_k,\theta), \qquad \Psi_N(\theta) = \nabla_\theta \hat{L}_N^\lambda(\theta).$$

Then

$$\Psi_N(\theta) = \frac{1}{N}\sum_{k=1}^{N} \nabla_\theta l^\lambda(T_k,\theta).$$

By Theorem 3, we have

$$\hat{\theta}_N \xrightarrow{p} \theta^*.$$

Since $\hat{L}_N^\lambda$ is differentiable and $\hat{\theta}_N$ is its unique minimizer, we have

$$\Psi_N(\hat{\theta}_N) = \nabla \hat{L}_N^\lambda(\hat{\theta}_N) = 0.$$

And since θ^* is the unique minimizer of the population criterion L^λ, we also have

$$\nabla_\theta L^\lambda(\theta^*) = 0$$
$$\mathbb{E}\big[\nabla_\theta l^\lambda(T,\theta^*)\big] = 0.$$

We now apply a Taylor expansion of Ψ_N around θ^*. Since Ψ_N is continuously differentiable, there exists $\tilde{\theta}_N$ between $\hat{\theta}_N$ and θ^* such that

$$0 = \Psi_N(\hat{\theta}_N) = \Psi_N(\theta^*) + \nabla\Psi_N(\tilde{\theta}_N)\big(\hat{\theta}_N - \theta^*\big).$$

Rearranging gives

$$\sqrt{N}\big(\hat{\theta}_N - \theta^*\big) = -\big(\nabla\Psi_N(\tilde{\theta}_N)\big)^{-1}\sqrt{N}\,\Psi_N(\theta^*). \quad (1)$$

We know that $\nabla^2 L_N^\lambda(\tilde{\theta}_N) = \nabla\Psi_N(\tilde{\theta}_N)$ is positive definite, then its inverse in (1) is well defined.
We now study the the right-hand side of (1).
First, since $\hat{\theta}_N \xrightarrow{p} \theta^*$, and $\tilde{\theta}_N$ a convex combination of $\hat{\theta}_N$ and θ^*, it also converges in probability to θ^*. Also,

$$\nabla\Psi_N(\tilde{\theta}_N) = \nabla^2 \hat{L}_N^\lambda(\tilde{\theta}_N) = \frac{1}{N}\sum_{k=1}^{N} \nabla^2 l^\lambda(T_k,\tilde{\theta}_N).$$

By the law of large numbers we obtain

$$\nabla\Psi_N(\tilde{\theta}_N) \xrightarrow{p} \mathbb{E}\big[\nabla^2 l^\lambda(T,\theta^*)\big] = \nabla^2 L^\lambda(\theta^*) = H.$$

Therefore

$$\big(\nabla\Psi_N(\tilde{\theta}_N)\big)^{-1} \xrightarrow{p} H^{-1}. \quad (2)$$

Second, we have

$$\sqrt{N}\,\Psi_N(\theta^*) = \frac{1}{\sqrt{N}}\sum_{k=1}^{N} \nabla l^\lambda(T_k,\theta^*).$$

The trajectories $T_1,\dots,T_N$ are i.i.d., and

$$\mathbb{E}\big[\nabla l^\lambda(T,\theta^*)\big] = 0.$$

Hence, by the multivariate central limit theorem,

$$\sqrt{N}\,\Psi_N(\theta^*) \xrightarrow{d} \mathcal{N}(0,\Sigma), \quad (3)$$

where

$$\Sigma = \mathrm{Var}\big(\nabla l^\lambda(T,\theta^*)\big).$$

Combining (1), (2), and (3) with Slutsky's theorem yields

$$\sqrt{N}\big(\hat{\theta}_N - \theta^*\big) \xrightarrow{d} -H^{-1}Z, \qquad Z \sim \mathcal{N}(0,\Sigma).$$

Since H is a Hessian matrix, it is symmetric, and since $-Z$ has the same distribution as Z, we get

$$-H^{-1}Z \sim \mathcal{N}\big(0,\, H^{-1}\Sigma H^{-1}\big)\,.$$

Therefore

$$\sqrt{N}\big(\hat{\theta}_N - \theta^*\big) \xrightarrow{d} \mathcal{N}\big(0,\, H^{-1}\Sigma H^{-1}\big)\,.$$

□

Proof of Theorem 5. Let θ_e^* be the population target parameter vector of the surrogate policy estimator for episode e. Let $\pi_e = \text{top-m}(X_e)$ be the true agent's policy at episode e, and X_e the underlying index vector. We show

$$\pi_1 = \pi_2 \implies \theta_1^* = \theta_2^*.$$

In order to show that implication, we have to prove that a well-defined mapping $F(\pi) = \theta^*$ exists.
Under the assumption that all trajectories are i.i.d, the initial distribution of the features Φ_1 is the same between episode 1 and episode 2.
Since the transition probabilities in the system depend only on the current state and action, the joint distribution of the features in a trajectory depend only on the policy that have been played. Consequently, the joint distribution of the actions in a trajectory also depends only on the policy that has been played.
Then the distribution of the trajectories P_T depends only on the policy that has been played. And we know that the population target parameter vector is derived from the distribution the trajectories. Hence a mapping $\pi \to P_T \to \theta*$.
We conclude

$$\pi_1 = \pi_2 \implies \theta_1^* = \theta_2^*.$$

□

Un modèle logique combinant la théorie de Dempster-Shafer et la théorie des possibilités pour la détection des erreurs de fixation

Gabrielle Porcher[1,2], Frédéric Boulanger[1], Nicolas Sabouret[2]

[1] Université Paris-Saclay, CNRS, ENS, CentraleSupélec, LMF, Gif-sur-Yvette, France
[2] Université Paris-Saclay, CNRS, LISN, Gif-sur-Yvette, France

Résumé

Les erreurs de fixation sont des erreurs d'origine humaine qui surviennent lorsqu'un individu se concentre de manière excessive sur une seule idée, une solution, une information ou une perspective, au point d'ignorer d'autres possibilités ou alternatives. Cet article présente un modèle fondé sur la logique formelle pour détecter des erreurs de fixation d'un opérateur humain dans une situation critique. Les informations communiquées à l'opérateur ainsi que ses actions sont représentées par des prédicats. Nous utilisons la théorie des fonctions de croyance pour calculer les états possibles du monde, puis nous utilisons le résultat dans le cadre possibiliste pour déterminer si les actions de l'opérateur sont cohérentes ou non pour alerter en temps réel l'opérateur sur un risque de fixation. Nous illustrons ce fonctionnement sur deux cas d'étude médicaux.

Mots-clés

Logique, Incertitude, Biais cognitifs, Théorie des possibilités, Théorie des fonctions de croyances

Abstract

Fixation errors are human-induced errors caused by an excessive focus on a single idea, solution, piece of information, or perspective, to the point of ignoring other possibilities. This paper presents a logic-based model for detecting fixation errors made by a human operator in a critical situation. The operator's actions and the information they receive are represented using predicates. We employ belief functions theory to compute the possible states of the world, and then utilize the possibilistic framework to determine whether the operator's actions are consistent with these states. This enables real-time alerts to be issued to the operator about a risk of fixation. We illustrate this approach with two medical case studies.

Keywords

Logics-Based Modeling, Uncertainty, Cognitive Biases, Possibility Theory, Theory of Belief Functions

1 Contexte et problématique

Le travail présenté dans cet article est réalisé dans le cadre du projet ANR IDEFIX (artificial Intelligence to DisEngage from FIXation) qui vise à décrire, étudier et prévenir les erreurs de fixations à l'aide d'un modèle d'intelligence artificielle. Dans des domaines comme la médecine ou l'aviation, dans lesquels la décision de l'opérateur met en jeu des vies humaines, il existe de nombreux outils ou méthodes pour aider l'opérateur dans sa décision. Les aides cognitives [6, 22] sont très utilisées : elles indiquent des conduites à suivre ou des éléments à vérifier mais elles n'aident pas toujours à identifier les bons diagnostics [3].

Les systèmes d'aide à la décision peuvent aussi être utilisés [17] mais ils présentent des risques de biais d'automatisation [13]. Dans le projet ANR IDEFIX, nous nous intéressons à une autre approche qui consiste à utiliser l'intelligence artificielle pour alerter les opérateurs lorsqu'ils se trouvent dans des situations où ils pourraient être victimes de biais de fixation. Le biais de fixation [9] est un cas particulier de biais cognitif [12] qui survient lorsqu'un individu se concentre de manière excessive sur une seule idée, une solution, une information ou une perspective, au point d'ignorer d'autres possibilités. Il est particulièrement critique en médecine ou en aviation et se retrouve sous différentes formes (tunnelisation, biais de confirmation, biais d'ancrage) [9]. Notre objectif n'est donc pas de dire à l'opérateur ce qu'il doit faire, comme dans un système d'aide à la décision, mais de l'alerter d'un risque potentiel.

L'un des enjeux dans le déploiement d'un tel outil est celui de l'explicabilité : un modèle « boîte noire » peut difficilement indiquer à l'opérateur les raisons de l'alerte. C'est pourquoi nous faisons le choix de nous appuyer sur un modèle en logique formelle qui manipule une représentation des informations et des actions de l'opérateur. Concrètement, dans le cadre du projet IDEFIX, les opérateurs sont confrontés à des scénarios (conçus par des experts métiers et des psychologues) favorisant l'apparition de biais de fixation. Le modèle informatique doit alors suivre en temps réel les évènements du scénario et alerter l'opérateur en cas de risque de fixation.

Nous espérons, à terme, montrer que l'IA permet de sortir plus rapidement de la fixation. Dans cet article, nous présentons une première version du modèle et du mécanisme d'alerte.

La construction d'un tel modèle soulève plusieurs problématiques :

1°) Pour créer des alertes pertinentes, il est nécessaire d'évaluer la possibilité d'une maladie (en médecine) ou d'une panne (en aviation) au cours du temps à partir des informations disponibles. Pour cela, il faut déterminer à quel point chaque information est associée à une maladie ou une panne, et comment ces associations se combinent quand plusieurs informations concernent une même maladie ou panne. Nous devons donc construire un **modèle logique capable d'effectuer des raisonnements de ce type**.

2°) Les informations transmises à l'opérateur sont parfois incomplètes ou incertaines. De même, les règles de raisonnement qui relient ces observations aux maladies ou pannes peuvent être ambiguës, ainsi que les règles qui décrivent les actions à effectuer pour les traiter. La raison en est qu'il n'est pas possible de capturer l'ensemble du réel dans des règles exactes et qu'il est donc nécessaire de manipuler des règles floues, avec de l'incertitude. Par exemple, une personne qui a la grippe a généralement de la fièvre, mais pas toujours. Un traitement est recommandé contre la COVID, sauf si le patient a des contre-indications, *et cetera*. Nous aurons donc besoin d'un **modèle logique capable de manipuler des incertitudes aussi bien au niveau des prédicats que des règles d'inférence**.

3°) Les experts humains utilisent des règles propres à leur métier pour réagir à des informations qu'ils reçoivent. Ces règles sont liées à des concepts situés à différents niveaux d'abstraction que les experts manipulent dans leur raisonnement. Par exemple lorsqu'un patient tousse, le médecin pense à une infection pulmonaire (concept abstrait), sans immédiatement décider entre une grippe ou une COVID (concepts plus concrets). Il faut donc que notre modèle métier supporte une modélisation hiérarchique des pannes et des maladies pour pouvoir effectuer des raisonnements sur des concepts de différents niveaux et les relier aux différentes actions. Mais surtout, ces actions peuvent nous renseigner sur le processus de décomposition des possibilités effectué par l'opérateur dans son diagnostic. Par exemple, demander un test bactérien permet d'isoler un sous-ensemble de maladies infectieuses sans pour autant savoir de quelle maladie précise il s'agit. Nous aurons donc besoin d'un **modèle métier qui permet de catégoriser des concepts de différents niveaux d'abstraction**.

Pour répondre à ces trois problématiques (et donc détecter un potentiel biais de fixation), nous proposons dans cet article un modèle logique capable de gérer l'incertitude et de suivre une exploration hiérarchique des pannes ou des maladies à l'aide d'un mécanisme d'inférence. Dans la prochaine section, nous présentons les travaux sur lesquels nous nous sommes appuyés pour construire ce modèle. Nous présentons dans la section 3 un premier cas d'étude qui servira d'exemple pour illustrer notre approche. Nous présentons le modèle lui-même dans la section 4, ainsi que le résultat obtenu sur notre cas d'étude. Dans la section 5 nous appliquons notre approche à un cas d'étude du projet IDEFIX. Enfin, nous discutons des limites et perspectives de ce modèle dans la section 7.

2 État de l'Art

Un de nos objectifs (problématique n° 1) est d'évaluer la possibilité des maladies ou pannes selon les évènements qui se présentent dans le scénario. Ces évènements peuvent être interprétés comme des symptômes déclenchés par les maladies et pannes elles-mêmes, ce qui se traduit par des règles logiques de la forme :

$$maladie/panne \rightarrow symptome$$

Dans ce cadre, retrouver l'origine du symptôme consiste à utiliser *l'abduction logique*. Des approches comme la logique abductive [14, 5] permettent de lister les causes possibles à partir des effets.
Dans nos travaux, nous avons besoin de combiner ces modèles abductifs avec un modèle de l'incertitude et avec un modèle de catégories hiérarchiques. Nous présentons dans les sections suivantes différents travaux allant dans ce sens.

2.1 L'incertitude

Pour modéliser le raisonnement de l'opérateur, nous avons besoin d'introduire de l'incertitude à la fois au niveau des prédicats et au niveau des règles. Il existe de nombreux travaux utilisant la logique pour le diagnostic médical, en particulier les systèmes experts tels que MYCIN [20] reposant sur des règles logiques et des facteurs de certitude, et les approches bayésiennes [15, 11]. Cependant, une difficulté est que nous n'avons pas connaissance des probabilités exactes associées à chaque fait et règle dans le contexte, ce qui rend difficile l'utilisation d'approches probabilistes, y compris des approches bayésiennes de l'abduction [16].

Logiques possibilistes

Un cadre logique permettant l'utilisation de poids non-probabilistes est le cadre de la logique possibiliste [8]. Elle vise à représenter et manipuler des connaissances incertaines qualitatives ou ordinales, sans recourir à une interprétation fréquentiste ou strictement probabiliste. Chaque formule logique est associée à un degré de nécessité (à quel point elle est nécessairement vraie) ou de possibilité (à quel point elle est possible d'après les informations disponibles), compris entre 0 et 1. Ces degrés ne sont pas interprétés comme des probabilités, mais comme des niveaux de compatibilité avec l'état du monde considéré. La combinaison des informations repose sur des opérateurs min et max, traduisant respectivement la conjonction prudente des contraintes, et l'agrégation de sources compatibles. Ce cadre a été utilisé avec succès pour représenter des raisonnements humains [18].
Cependant, dans nos travaux, nous faisons face à une difficulté supplémentaire : dans les scénarios sur lesquels nous travaillons, les observations sont nombreuses et peuvent donner des informations contradictoires. Dans le cadre possibiliste, la révision et la fusion d'informations se font avec les opérateurs min et max, ce qui avec nos données ne permet pas de conserver la finesse nécessaire au diagnostic. Si un symptôme faiblement associé à un diagnostic donné apparaît, les symptômes suivants, même très significatifs de

ce même diagnostic, seront absorbés sans pouvoir faire augmenter le degré de possibilité du diagnostic car l'algorithme prendra le minimum des valeurs de possibilité.

Exemple

On cherche à maximiser la mesure de possibilité de l'hypothèse H, qui peut être soit $Pneumonie$, soit $Covid$. On observe deux symptômes : des nausées (représentées par le prédicat $nausee$) puis un résultat de scanner caractéristique d'une pneumonie (que nous noterons $scan\_p$). La nausée est très peu associée aux deux hypothèses (0.1) tandis que le résultat au scanner est très associé à la pneumonie (0.9) et très peu à la COVID (0.1). La mesure de possibilité conjointe est donnée par :

$$\pi(nausee, scan\_p|H) = \min(\pi(nausee|H), \pi(scan\_p|H))$$

Si $H = Pneumonie$:
$\pi(nausee, scan\_p|Pneumonie) = \min(0.1, 0.9) = 0.1$

Si $H = Covid$:
$\pi(nausee, scan\_p|Covid) = \min(0.1, 0.1) = 0.1$

Dans cet exemple, on dit alors que les deux hypothèses $Pneumonie$ et $Covid$ expliquent les symptômes avec la même mesure de possibilité (0.1). Pourtant, on voudrait pouvoir associer l'hypothèse $Pneumonie$ à une valeur plus élevée, étant donné qu'elle est bien plus fortement associée au résultat de scanner obtenu.

Théorie des fonctions de croyance

Une autre approche proposée dans la littérature est la théorie des fonctions de croyance, introduite par Dempster [7] et formalisée par Shafer [19], qui introduit la notion de « probabilités incertaines » : la probabilité associée à chaque fait ou règle peut être comprise dans un intervalle (au lieu d'une valeur exacte). Elle permet ainsi de calculer la probabilité associée à chaque panne ou maladie à partir d'informations et de règles imprécises ou incertaines. En particulier, dans un cadre abductif, elle permet de raisonner sur des situations dans lesquelles un même symptôme peut être obtenu de manière incertaine à partir de pannes ou maladies différentes. On définit pour cela une *fonction de masse* qui attribue une valeur à chaque sous-ensemble de l'espace des hypothèses. La masse d'un ensemble représente la proportion de faits qui supportent l'hypothèse que l'état du monde soit dans cet ensemble et non dans un des sous-ensembles possibles. Elle représente donc à la fois le degré de certitude que l'état soit dans un ensemble, et l'absence d'information permettant de préciser l'état exact du monde. La règle de combinaison de Dempster permet alors d'agréger deux fonctions de masse indépendantes m_1 et m_2 définies sur le même ensemble d'hypothèses :

$$m = m_1 \oplus m_2$$

Nous illustrerons ce calcul dans la section 4. Nous montrerons qu'il n'y a alors pas d'effet d'absorption lié aux opérateurs min et max.

Remarque

La théorie des fonctions de croyances suppose un cadre de discernement (c'est-à-dire l'ensemble des abductibles exprimés) qui est fermé. En d'autres termes, l'ensemble des pannes ou maladies possibles doit être exhaustif. C'est le choix que nous avons fait dans notre modèle même si cette hypothèse pose plusieurs problèmes que nous discuterons dans la section 7.

2.2 La catégorisation

Le deuxième problème auquel nous faisons face est que le raisonnement des experts ne se situe pas directement au niveau des maladies mais utilise des catégories de plus haut niveau. Ce problème a été bien étudié par la communauté de l'ingénierie des connaissances, par exemple avec les logiques de description [1]. Ces modèles permettent de raisonner avec une représentation hiérarchique de la connaissance. Par exemple, on peut décrire qu'une atteinte pulmonaire provoque de la toux et en déduire que la grippe, qui est une sous-classe des atteintes pulmonaires, provoque ce symptôme. Dans notre cas, les actions menées par les opérateurs lors de leur exploration des possibilités peuvent être associées à des catégories de maladies ou pannes de plus haut niveau ; les représenter permet donc de suivre leur raisonnement. Par exemple, une radiographie thoracique peut être réalisée lorsqu'une atteinte pulmonaire est soupçonnée, mais cela ne permet pas de savoir si l'opérateur envisage une atteinte en particulier, comme une pneumonie.
Dans cet article, nous allons utiliser ce type de représentation et combiner cette catégorisation avec la théorie des fonctions de croyances pour mesurer la nécessité de la réalisation ou non des actions associées aux catégories.

3 Cas d'étude

Le modèle présenté dans la section suivante a été testé sur un scénario tiré d'un exemple réel dans lequel un médecin donne un traitement inadapté en raison d'un biais de fixation de type « tunnelisation » (c'est-à-dire une attention focalisée sur un symptôme ou une maladie particulière) décrit dans [2]. Nous utiliserons ce scénario pour illustrer les différents composants de notre modèle. L'implémentation de ce modèle est disponible sur GitHub [1].

Le scénario se déroule ainsi : un patient se présente avec un ensemble de symptômes (toux, fièvre, difficultés respiratoires). Le médecin en charge fait alors une série d'actions épistémiques (examens, tests, scanner) et reçoit des résultats. Il explore le diagnostic le plus probable, une pneumonie bactérienne typique, et donne le traitement associé. N'observant pas d'amélioration de l'état du patient, ce qui suggère un mauvais diagnostic, le médecin se tourne ensuite vers un diagnostic très peu probable, la COVID, vraisemblablement en raison du contexte de pandémie au moment des faits, et cela malgré des informations orientant vers une pneumonie atypique. Il donne donc le mauvais traitement alors qu'il disposait des bonnes informa-

1. https://github.com/GabriellePorcher/RJCIA_2026

tions (test COVID négatif, scanner suggérant fortement une pneumonie).

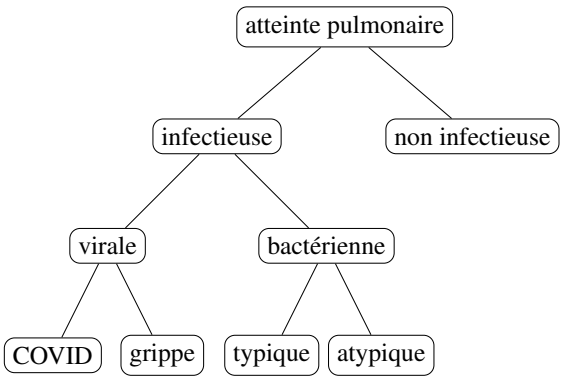


FIGURE 1 – Hiérarchie des atteintes pulmonaires

La hiérarchie des maladies que nous considérons est représentée sur la figure 1. Comme discuté dans la section précédente, il s'agit d'un monde fermé (alors que cet arbre est forcément très incomplet). Cette hiérarchie nous permet d'associer des actions à différentes catégories de maladies.

4 Modèle proposé

4.1 Principe général

Nous proposons de définir la fixation comme un écart entre la possibilité d'une maladie et à quel point l'opérateur semble l'envisager. Ainsi, un opérateur qui semble se concentrer sur une possibilité considérée comme peu probable par le modèle sera alerté d'un potentiel biais de fixation. L'algorithme évalue d'abord les pannes ou maladies possibles à partir des observations pour déterminer celles qui sont les plus probables. Il évalue ensuite les potentielles erreurs de l'opérateur en observant ses actions.
Notre proposition se divise donc en deux grandes parties :

— La détermination de la probabilité de la maladie ou panne, à partir des observations ;
— La détermination de la nécessité d'exécuter une action d'après des règles métier.

Selon le scénario, l'opérateur peut avoir à diagnostiquer une panne, une anomalie... Dans cette partie, de part le choix de notre cas d'étude, nous parlerons de suivi des maladies pour décrire les pistes réelles pouvant être explorées par les opérateurs.

4.2 Associations observations → maladies

Cette première partie de notre modèle s'appuie sur la théorie des fonctions de croyances de Dempster-Shafer [19] et la définition de fonctions de masses.

Considérons :

- $\boldsymbol{\Omega} = \{d_1, \ldots, d_n\}$ le cadre de discernement, correspondant à l'ensemble fini des maladies possibles ;
- $\boldsymbol{O} = \{o_1, \ldots, o_k\}$ l'ensemble des observations possibles (symptômes, signes, résultats d'examens) ;
- $\boldsymbol{2^{\Omega}}$ l'ensemble des sous-ensembles de Ω.

Pour chaque observation $o \in O$, nous devons définir une fonction de masse sur l'ensemble des diagnostics possibles :

$$m_o : 2^{\Omega} \to [0, 1]$$

telle que :

$$\sum_{S \subseteq \Omega} m_o(S) = 1 \quad \text{et} \quad m_o(\varnothing) = 0.$$

Interprétation :

- $\boldsymbol{m_o(\{d_i\})}$ représente le poids directement attribué à la maladie d_i par l'observation o ;
- $\boldsymbol{m_o(S)}$ avec $|S| > 1$, représente le poids associé aux éléments de S, sans discrimination (l'observation soutient un ensemble de maladies sans distinction) ;
- $\boldsymbol{m_o(\Omega)}$ modélise l'incertitude globale associée à l'observation.

Illustration sur le cas d'étude

Dans notre exemple, nous considérons 4 maladies (pneumonie typique, pneumonie atypique, grippe et COVID) :

$$\Omega = \{pneumo\_t, pneumo\_a, grippe, covid\}$$

et un ensemble d'observations correspondant aux symptômes, aux résultats d'examens cliniques ou à l'état général du patient :

$$O = \{toux, fievre, sat, rales, scanb, ac\}$$

où *toux* et *fievre* correspondent aux symptômes décrits par le patient à son arrivée, *sat* représente une saturation en oxygène normale mesurée par l'oxymètre, *rales* représente la présence de râles crépitants à l'auscultation pulmonaire, *scanb* représente la présence d'une structure en arbre en bourgeon lors du scanner thoracique et *ac* représente une amélioration générale de l'état du patient.
Il y a bien sûr d'autres observations (symptômes, examens ou description de l'état du patient) dans le scénario complet mais nous nous limiterons à cette liste pour illustrer notre modèle dans cet article.

Les valeurs attribuées aux fonctions de masse de chaque observation pour les maladies sont définies arbitrairement (elles nous permettent simplement ici d'illustrer les propriétés du modèle, pas de valider un résultat médical).
Voici un exemple de fonction de masse : l'observation d'une structure en arbre en bourgeons au scanner thoracique est un signe radiologique indiquant plutôt une pneumonie. Nous pouvons le représenter par :

$$m_{scanb} = \begin{cases} 0.8 & \to pneumo, \\ 0.05 & \to grippe, \\ 0.05 & \to covid, \\ 0.1 & \to \Omega. \end{cases}$$

avec $pneumo = \{pneumo\_t, pneumo\_a\}$ le sous-ensemble regroupant les deux pneumonies. L'observation $scanb$ supporte donc fortement l'hypothèse « pneumonie », sans permettre de discriminer les deux types de pneumonie. Remarquons que tous les sous-ensembles de Ω ne sont pas associés à une valeur. La masse des ensembles non-mentionnés est nulle. La masse d'Ω représente l'incertitude globale sur le diagnostic.

4.3 Calcul du score à partir des observations

Considérons $A \subseteq \Omega$ un ensemble de maladies et deux observations o_1 et o_2 dont on souhaite calculer la combinaison des fonctions de masse $(m_{o_1} \oplus m_{o_2})(A)$. Dans la théorie de Dempster-Shafer, le calcul de $\oplus$ se fait de la manière suivante :

$$(m_{o_1} \oplus m_{o_2})(A) = \frac{1}{1-K} \sum_{B \cap C = A} m_{o_1}(B)\, m_{o_2}(C)$$

où :

$$K = \sum_{B \cap C = \varnothing} m_{o_1}(B)\, m_{o_2}(C)$$

représente le degré de conflit entre les deux sources d'information, mesurant la contradiction entre ces sources.
Cet opérateur combine donc les fonctions de masse de tous les sous-ensembles qui soutiennent l'hypothèse A pour les deux observations.
Lorsque plusieurs observations sont faites dans un scénario, nous notons $m(A)$ le score final associé au diagnostic A après combinaison des fonctions de masses de toutes les observations (A peut aussi bien être une maladie dans Ω, ou une catégorie de maladies dans 2^Ω) :

$$m(A) = (m_{o_1} \oplus \cdots \oplus m_{o_k})(A)$$

Illustration sur le cas d'étude

Pour illustrer cette combinaison des fonctions de masse, considérons deux observations : une saturation en oxygène normale (sat), plutôt indicatrice de grippe ou de COVID, et la présence de râles crépitants à l'auscultation pulmonaire ($rales$), plutôt indicateurs d'une pneumonie, sans que cela permette de distinguer le type de bactérie responsable. Les valeurs des fonctions de masse pour ces deux observations sont décrites dans le tableau suivant :

A	$m_{sat}(A)$	$m_{rales}(A)$
$\{pneumo\}$	0.2	0.60
$\{grippe\}$	0.30	0.10
$\{covid\}$	0.40	0.20
Ω	0.10	0.10

Nous calculons alors la valeur du conflit K (ce qui est assez simple dans notre cas puisque tous les éléments sauf Ω sont disjoints) :

$$\begin{aligned} K &= 0.2 \times (0.1 + 0.2) \\ &\quad + 0.3 \times (0.6 + 0.2) \\ &\quad + 0.4 \times (0.6 + 0.1) \\ &= 0.58. \end{aligned}$$

Les masses combinées sont alors déterminées comme suit (avec $1 - K = 0.42$) :

$$\begin{aligned} m(\{pneumo\}) = &\\ &\tfrac{1}{0.42}(m_{sat}(\{pneumo\}) \cdot m_{rales}(\{pneumo\}) \\ &+ m_{sat}(\{pneumo\}) \cdot m_{rales}(\{\Omega\}) \\ &+ m_{sat}(\{\Omega\}) \cdot m_{rales}(\{pneumo\})) \\ &\approx 0.476 \end{aligned}$$

$$\begin{aligned} m(\{grippe\}) = &\\ &\tfrac{1}{0.42}(m_{sat}(\{grippe\}) \cdot m_{rales}(\{grippe\}) \\ &+ m_{sat}(\{grippe\}) \cdot m_{rales}(\{\Omega\}) \\ &+ m_{sat}(\{\Omega\}) \cdot m_{rales}(\{grippe\})) \\ &\approx 0.167 \end{aligned}$$

$$\begin{aligned} m(\{covid\}) = &\\ &\tfrac{1}{0.42}(m_{sat}(\{covid\}) \cdot m_{rales}(\{covid\}) \\ &+ m_{sat}(\{covid\}) \cdot m_{rales}(\{\Omega\}) \\ &+ m_{sat}(\{\Omega\}) \cdot m_{rales}(\{covid\})) \\ &\approx 0.334 \end{aligned}$$

$$\begin{aligned} m(\Omega) = &\\ &\tfrac{1}{0.42}(m_{sat}(\Omega) \cdot m_{rales}(\Omega)) \\ &\approx 0.024 \end{aligned}$$

Remarquons que la combinaison des masses par $\oplus$ permet bien d'obtenir des valeurs reflétant les informations données par les deux symptômes, où les poids peuvent être renforcés et affaiblis : ce type de résultat permet donc de déterminer à quel point les maladies sont soutenues par toutes les observations, et donc la pertinence qu'aurait le médecin à les explorer (actions épistémiques) ou à les traiter (actions curatives).

4.4 Modélisation des règles liées aux actions

La deuxième partie de notre modèle consiste à déterminer la nécessité d'une action. Cela permet d'alerter l'opérateur (*i.e.* le médecin dans notre exemple) d'un possible biais de fixation s'il ne la fait pas. Pour cela, nous allons nous appuyer sur la représentation hiérarchique des maladies.

Hiérarchie des maladies

Nous notons $\mathcal{H}$ la hiérarchie structurant les concepts de différents niveaux que manipulent les opérateurs dans un contexte donné (et dont un exemple est donné sur la figure 1). Ces concepts sont soit des éléments du cadre de discernement Ω, soit des concepts plus abstraits, par exemple $infection\_bacterienne$ qui regroupe les deux cas de pneumonie présents dans Ω.
Notons que la hiérarchie $\mathcal{H}$ peut aussi inclure des nœuds ne correspondant à aucune maladie spécifique de notre étude, tels que la catégorie « non infectieuse » sur la figure 1. Bien qu'aucun élément de Ω n'appartienne à cette catégorie, sa

représentation reste nécessaire dans notre modèle : elle permet d'associer des actions spécifiques d'un médecin en présence de ce type de pathologie. Les éléments de Ω sont nécessairement des feuilles de $\mathcal{H}$ car ce sont les diagnostics les plus concrets auxquels nous nous intéressons.

Le savoir-faire de l'opérateur détermine les actions à entreprendre en fonction de la catégorie de maladie suspectée. Nous représentons ces connaissances de l'expert sous forme de règles indiquant la nécessité $\alpha \in [0, 1]$ de réaliser une action a en présence d'un diagnostic $C \in \mathcal{H}$:

$$C \overset{\alpha}{\Rightarrow} a$$

Le degré de nécessité α représente la nécessité de l'action lorsque toutes les prémisses sont pleinement satisfaites.
Par exemple, en présence certaine d'une atteinte pulmonaire infectieuse d'origine bactérienne, il est nécessaire à 90% de prescrire des antibiotiques :

$$bacterienne \overset{0.9}{\Longrightarrow} antibio$$

Pour déterminer la nécessité d'une action, il nous faut combiner la nécessité d'appliquer une règle et la certitude que nous avons sur un diagnostic. La théorie de Dempster-Shafer qui a été pertinente pour fusionner les informations issues des observations (comme nous l'avons montré dans la partie précédente), ne s'applique pas ici. En effet, il ne s'agit pas de combiner différentes sources pour en déduire des faits, mais de mettre en regard la confiance dans un diagnostic et la nécessité de faire une action. Nous nous plaçons donc dans un cadre possibiliste.
Dans ce contexte, nous interprétons les fonctions de masse précédemment obtenues comme des degrés de nécessité au sens possibiliste, c'est-à-dire la nécessité que le patient soit atteint d'une maladie donnée.
Ainsi, pour tout concept $C \in \mathcal{H}$, nous définissons son degré de nécessité $N(C)$ comme suit :

— Si $C \in \Omega$, le degré de nécessité de la maladie est la valeur donnée par la combinaison des fonctions de masse pour toutes les observations :

$$N(C \in \Omega) = m(C)$$

— Sinon, le degré de nécessité d'un concept de plus haut niveau est défini comme la somme de sa masse et des degrés de nécessité de ses fils :

$$N(C \notin \Omega) = m(C) + \sum_{D \, is\text{-}a \, C} N(D)$$

avec $is\text{-}a$ la relation entre deux sommets de $\mathcal{H}$: $D \, is\text{-}a \, C$ si D est fils direct de C dans l'arbre $\mathcal{H}$.

Notons que la nécessité d'un concept qui ne généralise aucun élément de Ω (comme « non infectieuse » dans notre exemple) sera toujours nulle. Cela correspond au fait que, dans le scénario considéré, cette catégorie de maladies est extrêmement peu probable et aurait dû être écartée par l'opérateur.

Nécessité d'une action

Pour une règle donnée, en suivant le cadre de la logique possibiliste présenté à la section 2, nous pouvons calculer le degré de nécessité associé à l'action a comme la combinaison, avec l'opérateur $\min$, du degré de nécessité de C et de celui de la règle :

$$N(a) = \min(\alpha, \, N(C))$$

Dans le cas général, lorsque plusieurs règles métier peuvent conduire à la même action, nous combinons ces règles avec l'opérateur $\max$:

$$N(a) = \max_{C \overset{\alpha}{\Rightarrow} a}(\min(\alpha, \, N(C)))$$

Ainsi, le degré de nécessité d'une action correspond au maximum, pour toutes les règles qui s'appliquent, du minimum entre :

— le degré de nécessité de la prémisse,
— le degré de nécessité de la règle logique associée.

L'utilisation du $\max$ pour combiner les nécessités induites par plusieurs règles, plutôt que d'utiliser Dempster-Shafer, se justifie par le fait que l'on combine des nécessités et pas des masses : à partir du moment où une action est nécessaire, cette nécessité ne peut pas être amoindrie par le fait qu'une autre règle la rend aussi nécessaire, mais à un degré moindre.

Illustration sur le cas d'étude

Dans notre scénario, supposons que nous avons la règle suivante :

$$covid \overset{0.9}{\Longrightarrow} traitement\_covid$$

Si $m(\{covid\}) = 0.1$ alors le degré de nécessité associé à l'action $traitement\_covid$ est $\min(0.1, 0.9) = 0.1$.

Seuils d'alerte

Pour déterminer quand alerter l'opérateur qui n'effectue pas une action nécessaire, nous définissons trois valeurs :

— un seuil d'alerte σ_{min} qui représente le degré de nécessité en-dessous duquel une action ne devrait jamais être faite ;
— un seuil d'alerte σ_{max} qui représente le degré de nécessité à partir duquel une action devrait absolument être faite ;
— une durée δ qui représente le délai maximum avant une alerte pour les actions non effectuées.

Notre modèle déclenche une alerte dans deux cas :

— l'opérateur fait l'action a alors que $N(a) < \sigma_{min}$,
— il existe une action a telle que $N(a) > \sigma_{max}$ depuis au moins δ, et qui n'a pas été effectuée.

La durée δ permet de prendre en compte le temps nécessaire à l'opérateur pour effectuer l'action, et d'éviter les alarmes intempestives.
Dans ces deux situations, nous considérons qu'il y a un risque d'erreur de fixation de la part de l'opérateur.

4.5 Résultats sur le cas d'étude

Nous avons implémenté le scénario de notre cas d'étude (voir section 3) avec des fonctions de masses définies comme illustré dans la sous-section 4.3. Nous avons choisi $\sigma_{min} = 0.2$ comme seuil d'alerte pour les actions effectuées et $\sigma_{max} = 1$. Nous ne déclenchons donc pas d'alerte pour les actions non-effectuées car nous ne considérons dans ce scénario que les cas de fixation entraînant une action non justifiée, en raison de l'absence d'information sur les autres actions possibles [2].
La figure 2 illustre l'évolution des masses des quatre maladies, de l'ensemble Ω (ignorance totale) ainsi que les valeurs du conflit K. Nous voyons que la COVID est plus probable au début du scénario et diminue assez vite (après les premiers examens) alors que la masse associée à la pneumonie atypique augmente rapidement et domine les masses des autres maladies. La valeur de K qui reste élevée montre un conflit significatif pendant la majeure partie du scénario, jusqu'à ce que la bactérie responsable de la pneumonie atypique soit découverte par un test.

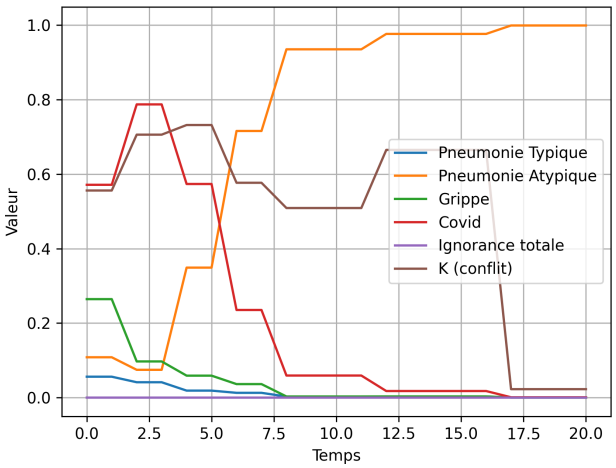


FIGURE 2 – Masses et conflit cas n° 1

Au cours du scénario, le médecin effectue plusieurs actions. Notre modèle donne une alerte pour quatre d'entre elles :

- La première à $t = 9$ lorsqu'il donne un traitement permettant de soigner les pneumonies typiques. À cette date, la masse associée à ce type de pneumonie est particulièrement faible.
  Dans la réalité, il n'est pas anormal d'explorer cette piste en premier. En l'absence d'information complémentaire, ce traitement va permettre au médecin soit de guérir le patient, soit d'écarter la maladie la plus probable en l'absence d'amélioration.
- La seconde alerte à $t = 11$ concerne un test COVID : au lieu de persister dans la catégorie des infections bactériennes, qui est l'explication la plus plausible, le médecin décide de tester la COVID qui est peu probable au vu des informations.
  Dans la réalité, le contexte général de la pandémie en 2020 explique le choix de cette action épistémique, peu invasive.
- Malgré un test COVID négatif, le médecin décide de donner un traitement anti COVID ($t = 13$) et de réaliser un second test COVID ($t = 14$).
  Il s'agit là manifestement d'erreurs de fixation. Notre modèle déclenche des alertes car la valeur attribuée à COVID continue de décliner.

Ce scénario simple montre deux choses : premièrement, que toutes les situations d'alerte ne correspondent pas nécessairement à des erreurs de fixation ($t = 9$ et $t = 11$) : le médecin peut avoir de bonnes raisons de faire une action qui ne semble pas nécessaire. Deuxièmement, malgré la simplicité des règles métier utilisées, notre modèle est capable de repérer une situation de fixation. Nous pouvons penser que si le médecin avait été alerté, il n'aurait peut-être pas fait l'erreur de prescrire un traitement anti COVID. C'est justement ce que nous voulons étudier dans le projet IDEFIX.

5 Un scénario du projet IDEFIX

Dans le cadre du projet ANR IDEFIX, des scénarios inspirés de situations réellement vécues à l'hôpital ont été construits dans un but de formation des élèves médecins. Ils sont conçus spécialement pour conduire à une fixation. Dans un de ces scénarios, le patient (qui est un mannequin de simulation médicale dans nos expériences) présente des troubles neurologiques et des symptômes infectieux, ce qui conduit à un premier diagnostic de méningite (cohérent avec ces symptômes). Des informations ultérieures viennent contredire ce diagnostic initial (le patient souffre en réalité de deux pathologies : une grippe, expliquant les symptômes infectieux, et un AVC, expliquant les troubles neurologiques). Une trentaine d'élèves de l'école de médecine de Lyon ont été confrontés à ce scénario et enregistrés (vidéo, audio et données du simulateur médical utilisé). Les informations nécessaires à l'application de notre modèle ont été extraites d'un de ces enregistrements afin de nous permettre d'évaluer son potentiel sur un cas réel.

Modélisation des hypothèses multiples

Ce scénario présente une problématique particulière : le patient présente conjointement deux pathologies. Le cadre de discernement (l'ensemble des hypothèses envisageables dans la théorie des fonctions de croyances) est ici constitué de la méningite, la grippe, l'AVC, et l'occurrence simultanée d'une grippe et d'un AVC : les hypothèses du cadre de discernement devant être mutuellement exclusives, nous gérons la présence de plusieurs pathologies simultanées en les ajoutant explicitement au cadre de discernement. Ainsi, $\{grippe\}$ correspond à un patient atteint de grippe seulement, tandis que $\{grippe\_avc\}$ correspond à un patient atteint de grippe et faisant un AVC.
Pour gérer l'attribution des masses dans ce cadre, lorsqu'un symptôme est lié à deux maladies du cadre de discernement qui peuvent être simultanées, on attribue la masse au sous-ensemble comprenant tous les éléments du cadre où la maladie figure. Par exemple, si le test de la grippe revient positif, on attribue la masse au sous-ensemble $\{grippe, grippe\_avc\}$ indiquant qu'une de ces deux hypo-

thèses est vraie (grippe seule ou grippe avec AVC) sans pouvoir déterminer laquelle d'après le résultat de test.
De la même manière, dans la modélisation de la nécessité des actions, si la grippe entraîne une action, on écrira que $grippe \vee grippe\_avc$ implique l'action : par exemple, $grippe \vee grippe\_avc \overset{\alpha}{\Longrightarrow} tamiflu$ signifie que si le minimum de α et de la masse associée à la grippe seule ou à la grippe avec AVC dépasse un seuil donné, le médecin devrait donner au patient le médicament antiviral Tamiflu, indiqué contre la grippe.

Entrées du modèle : observations du médecin

Nous distinguons les observations de l'opérateur, qui nous renseignent sur l'état du monde, et ses actions, qui nous renseignent sur les hypothèses qu'il envisage. Toutes les observations connues du participant sont utilisées en entrée du modèle de diagnostic (Dempster-Shafer), qu'elles soient présentes dans le briefing pré-intervention ou qu'elles surviennent lors du déroulement du scénario. Sont donc pris en compte des facteurs de risques (âge, fumeur, hypertension), des constantes vitales (fréquence cardiaque, pression artérielle, saturation en oxygène), des symptômes apparaissant spontanément (état des pupilles, niveau de conscience), et des résultats de tests ou d'actions réalisées (résultat de ponction lombaire, réactions à un traitement).

Entrées du modèle : actions du médecin

Les actions de l'élève médecin comprennent des actions qu'il effectue directement et des actions qu'il délègue à l'infirmier anesthésiste. Dans notre modèle, l'action « examen des pupilles » représente donc aussi bien l'examen des pupilles par le médecin que le fait qu'il demande à l'infirmier de les examiner. Nous distinguons également les actions de prise d'information, dites épistémiques (comme l'examen des pupilles), des actions de traitement, dites pragmatiques (par exemple administrer un antibiotique). Parmi ces dernières, certaines actions ne servent qu'à stabiliser l'état du patient et ne sont pas spécifiques à une pathologie particulière. Nous avons décidé de ne pas les représenter dans le modèle, car elles ne donnent pas lieu à des biais de fixation. Nous n'avons pas non plus représenté les actions de verbalisation des hypothèses (« je crois qu'il a la grippe ») mais nous verrons plus tard qu'elles pourraient être utiles pour valider notre modèle.
Comme dans le premier cas d'étude, les pathologies considérées sont regroupées en catégories dans la hiérarchie représentée sur la figure 3. Ces catégories sont utilisées comme prémisses dans les règles de nécessité des actions.

Résultats

L'évolution des masses des différentes pathologies au cours du scénario est illustrée sur la figure 4. À $t = 0$, les masses sont assez précises étant donné que les nombreuses informations données dans le briefing ont déjà été enregistrées. Si pendant la première moitié du scénario, l'hypothèse de la méningite est favorisée, elle diminue nettement à $t = 16$ au profit de l'hypothèse de la grippe et de l'AVC conjoints, lorsqu'on apprend que le patient est positif à la grippe. Cette

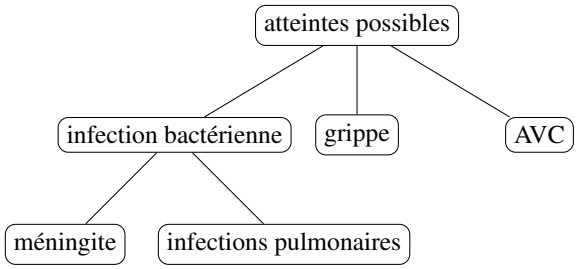


FIGURE 3 – Hiérarchie de pathologies du scénario

tendance se poursuit à $t = 28$ lorsque le patient présente une mydriase, un état de dilatation de la pupille caractéristique de l'AVC. Nous pouvons aussi remarquer que la masse de l'AVC seul reste à zéro car, dès le début de la simulation, il y a des indices clairs de maladie infectieuse, qui rend impossible ce singleton. De même, la masse de la grippe seule reste assez basse en raison des symptômes qui favorisent les sur-ensembles incluant aussi des affections neurologiques.

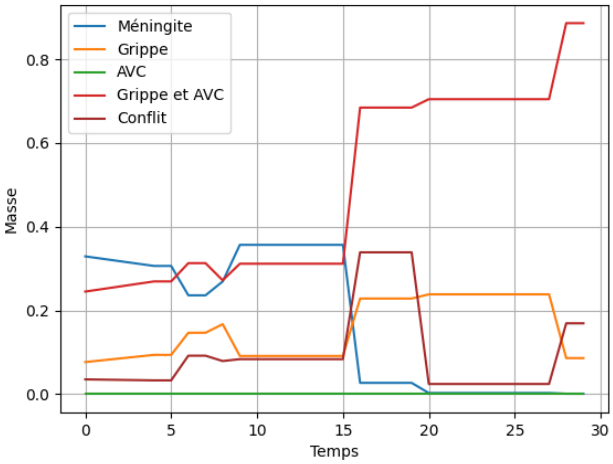


FIGURE 4 – Masses et conflit cas n° 2

Dans ce scénario, nous obtenons six alertes (avec les seuils $\sigma_{max} = 0.7$ et $\sigma_{min} = 0.3$) en réaction aux actions de l'élève médecin. À partir de $t = 16$, alors que la masse de l'ensemble avec grippe et AVC passe au dessus de σ_{max}, nous recevons une première alerte indiquant que l'action « réaliser un scanner avec injection » (scanner permettant de détecter l'AVC) doit être réalisé, étant donné que le degré de nécessité associé à cette règle est également supérieur au seuil fixé. Il s'agit de la seule action spécifique à l'AVC n'ayant pas encore été réalisée par l'opérateur dans ce scénario. Dans la réalité, il sera intéressant de notifier ce type d'action, qui pourrait permettre à l'opérateur d'envisager à nouveau la piste de l'AVC.
Les cinq alertes déclenchées ensuite correspondent à des actions spécifiques soit à la méningite, soit à des infections respiratoires, qui n'auraient pas dû être réalisées d'après le modèle, telle qu'un scanner thoracique. En réalité, bien que

les maladies détectables par ce test soient associées à des masses très faibles, réaliser ce type de test peut toujours être intéressant et surtout peu coûteux pour le patient s'il est stable.
En conclusion, le modèle propose des alertes cohérentes sur ce scénario en jugeant correctement les hypothèses les plus probables, et serait capable de notifier un utilisateur à la fois pour suggérer des actions et mettre en en avant des actions réalisées mais non-nécessaires, pouvant être causées par un biais de fixation. Il sera intéressant d'appliquer ce modèle avec les mêmes paramètres à d'autres participants à l'étude sur ce même scénario pour observer si les alertes restent pertinentes. Les prochaines étapes du projet IDEFIX vont consister à voir si les indications de notre modèle permettent effectivement au médecin, sur ces mêmes scénarios, de sortir plus tôt de la fixation.

6 Discussion

L'approche permettant l'évaluation des probabilités par la théorie des fonctions de croyances semble montrer de bons résultats sur le cas d'étude et le scénario du projet que nous avons utilisés, mais elle présente aussi plusieurs limites. En particulier, lorsque la valeur de K est trop élevée, Zadeh montre dans [23] qu'il faut conserver une certaine prudence dans l'interprétation des résultats. En effet, lorsque les sources sont contradictoires, la normalisation amplifie les masses des sous-ensembles non vides. Dans notre premier exemple, en sortie du modèle, une seule maladie domine très largement les autres, alors que dans la réalité plusieurs pistes sont probables en particulier en début de scénario alors que peu d'informations sont encore disponibles.
De plus, dans notre modèle, nous avons travaillé avec un cadre de discernement fermé (considérant que toutes les hypothèses sont représentées dans Ω) et avec des hypothèses mutuellement exclusives, comme le propose Shafer [19]. Pourtant, travailler avec un cadre ouvert et pouvoir représenter le fait que plusieurs hypothèses peuvent être vraies en même temps serait plus proche de la réalité à laquelle sont confrontés les opérateurs.
Enfin, l'attribution des masses est aussi une problématique importante pour l'utilisation de notre modèle. Elle nécessite une expertise métier dans un contexte spécifique à chaque situation. Dans notre exemple, le choix des masses permet de produire trois bonnes alertes mais il cause aussi une alerte qu'il aurait été préférable d'éviter, alors que l'opérateur explore la possibilité d'une pneumonie typique.
La section suivante propose des pistes de résolution pour les problématiques identifiées ici.

7 Conclusion et perspectives

Ce travail propose un modèle formel pour la détection en temps réel du biais de fixation chez les agents humains dans des contextes critiques, en combinant deux approches complémentaires : une modélisation de la réalité via la théorie des fonctions de croyances (Dempster-Shafer) pour estimer dynamiquement la nécessité des maladies ou pannes, et une modélisation de la pertinence des actions possibles via la logique possibiliste pour déterminer si le comportement de l'opérateur est conforme aux règles métier.
Parmi les problématiques citées dans la section 6, nous avions noté que Dempster-Shafer suppose que tous les éléments de Ω sont mutuellement exclusifs, ce qui n'est pas toujours le cas dans les applications que nous envisageons. Des modifications du cadre ont été proposées pour prendre en compte la conjonctions de plusieurs hypothèses, par exemple dans [4], Laurence Cholvy considère que leur conjonction est une hypothèse que l'on ajoute explicitement au cadre. Cela permet de mieux représenter la réalité des problématiques auxquelles les opérateurs sont confrontés. Selon le nombre d'hypothèses considérées, le nombre d'états peut exploser.
De plus, la théorie des fonctions de croyances suppose un cadre de discernement fermé, comme nous l'avons évoqué dans la section 2 et la section 6. Nous faisons l'hypothèse que toutes les maladies possibles sont décrites dans Ω puis rassemblées dans $\mathcal{H}$. En pratique, ce n'est pas réaliste et nous voudrions conserver la possibilité d'autres maladies non représentées. Pour cela, il est possible de s'appuyer sur le modèle de croyances transférables [21] qui définit un cadre similaire à la théorie des fonctions de croyances tout en adoptant une hypothèse de monde ouvert, sans normalisation, où la masse sur l'ensemble vide représente soit un conflit réel, soit la possibilité que la vérité ne se situe pas dans le cadre considéré. Ce modèle pourrait théoriquement permettre de mieux représenter l'état du monde, en particulier hors de scénarios contrôlés, mais c'est une piste qui reste à explorer.
Enfin, nous avons souligné la problématique de l'attribution des masses dans la théorie des fonctions de croyances. Il faudra être en mesure de calibrer les valeurs du modèle informatique pour chaque scénario (à l'aide d'une méthode de calcul de point fixe ou d'algorithmes évolutionnaires d'exploration de l'espace des paramètres comme CMA-ES [10] qui est souvent utilisé en simulation) afin d'optimiser les différentes masses pour obtenir des résultats correspondant à ceux observés dans nos scénarios réels avec des médecins humains dans de bonnes conditions, c'est-à-dire sans biais.

Malgré ces limites bien identifiées, notre travail ouvre des perspectives intéressantes en termes de modélisation de l'erreur humaine. Nous voudrions compléter notre modèle en y intégrant une représentation des désirs et intentions des opérateurs, qui permettraient d'expliquer des actions qui ne sont pas nécessairement des erreurs (par exemple lorsque le médecin donne priorité à la survie du patient plutôt qu'à son diagnostic) et de prendre en compte les conditions et effets des actions comme formalisé dans les logiques d'action de type BDI. Nous voudrions aussi modéliser différents niveaux d'urgence (au lieu de seuils fixes pour l'ensemble des maladies) pour qu'une piste relativement peu probable mais grave, donc représentant un risque important, puisse être explorée par l'opérateur. Il pourrait au contraire être intéressant de ne produire certaines alertes qu'en situation d'urgence et ainsi de ne pas saturer l'opérateur, ce qui ris-

querait de le désensibiliser et de diminuer sa réceptivité. De plus, nous avons pu remarquer que certaines alertes produites dans nos expérimentations sont un peu trop rigides ; une action épistémique peu coûteuse pour tester une maladie ou une panne, même dont la masse est faible, n'est pas nécessairement une erreur.
Enfin, dans le cadre du projet IDEFIX, ce modèle sera utilisé sur différents scénarios, médicaux et aéronautiques, auprès de professionnels. Quatre scénarios en médecine ont été implémentés et quatre autres en aviation sont en cours d'implémentation. Nous modéliserons les activités des 30 sujets dans chaque domaine (aviation et médecine) à l'aide de notre modèle pour étudier l'apparition d'alertes et déterminer comment calibrer notre modèle en vue d'expérimentations futures : les verbalisations des médecins lors des scénarios seront alors des outils précieux pour évaluer la pertinence des prédictions de notre modèle.
En appliquant notre approche à ces différents scénarios, nous pourrons vérifier si le cadre de modélisation que nous avons développé est suffisamment riche pour donner des résultats cohérents à la fois dans des contextes différents et dans plusieurs domaines métiers.

À terme, nous espérons construire un modèle qui permette non seulement d'alerter l'opérateur sur la possibilité d'un biais de fixation, mais aussi de produire des explications sur ces alertes, ce qui peut les rendre plus acceptables par l'humain. Notre objectif, dans une démarche d'interaction humain-IA, est de proposer un modèle d'IA explicable pour aider les médecins et les pilotes à mieux éviter ces erreurs qui peuvent avoir des conséquences dramatiques.

Références

[1] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook : Theory, Implementation and Applications*. Cambridge University Press, 2 edition, 2007.

[2] A Bertaux, B Alameda, J Tataw, and A Kenfak. Effet tunnel en contexte d'épidémie. *Revue Médicale Suisse*, 16(718) :2392–2396, 2020.

[3] Alex Chaparro, Joseph Keebler, Elizabeth Lazzara, and Anastasia Diamond. Checklists : A review of their origins, benefits, and current uses as a cognitive aid in medicine. *Ergonomics in Design : The Quarterly of Human Factors Applications*, 27, 01 2019.

[4] Laurence Cholvy. Non-exclusive hypotheses in dempster–shafer theory. *International Journal of Approximate Reasoning*, 53(4) :493–501, 2012.

[5] Alessandro Cimatti and Marco Schaerf. Abductive reasoning in description logics. *Journal of Automated Reasoning*, 22(1) :1–39, 1999.

[6] Asaf Degani and Earl L. Wiener. Cockpit checklists : Concepts, design, and use. *Human Factors*, 35(2) :345–359, 1993.

[7] Arthur P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematical Statistics*, 38(2) :325–339, 1967.

[8] Didier Dubois and Henri Prade. *Possibility Theory : An Approach to Computerized Processing of Uncertainty*. Plenum Press, 1988.

[9] Evie Fioratou, Rhona Flin, and Ronnie Glavin. No simple fix for fixation errors : cognitive processes and their clinical applications. *Anaesthesia*, 65(1) :61–69, 2010.

[10] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, 9(2) :159–195, June 2001.

[11] David Heckerman, Eric Horvitz, and Bharat Nathwani. The pathfinder system. *Proceedings / the ... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care*, 11 1989.

[12] Daniel Kahneman and Amos Tversky. Judgment under uncertainty : Heuristics and biases. *Science*, 185(4157) :1124–1131, 1974.

[13] David Lyell and Enrico Coiera. Automation bias and verification complexity : A systematic review. *Journal of the American Medical Informatics Association*, 24 :ocw105, 08 2016.

[14] John McCarthy. Circumscription – a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1–2) :27–39, 1980.

[15] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[16] David Poole. Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1) :81–129, 1993.

[17] Daniel J. Power. *Decision Support Systems : A Historical Overview*, pages 121–140. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[18] Eric Raufaste, Rui da Silva Neves, and Claudette Mariné. Testing the descriptive validity of possibility theory in human judgments of uncertainty. *Artificial Intelligence*, 148(1) :197–218, 2003. Fuzzy Set and Possibility Theory-Based Methods in Artificial Intelligence.

[19] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[20] Edward Shortliffe. Computer-based medical consultations : Mycin. *Artificial Intelligence - AI*, 388, 10 1976.

[21] Philippe Smets and Robert Kennes. The transferable belief model. *Artificial Intelligence*, 66(2) :191–234, 1994.

[22] World Health Organization. *WHO Surgical Safety Checklist and Implementation Manual*. World Health Organization, Geneva, 2009. First Edition.

[23] Lotfi A. Zadeh. A simple view of the dempster-shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(2) :85, Jun. 1986.