GenAI for optimization of laminated composite structures

COMMERCIAL AIRCRAFT

Driss Chraibi

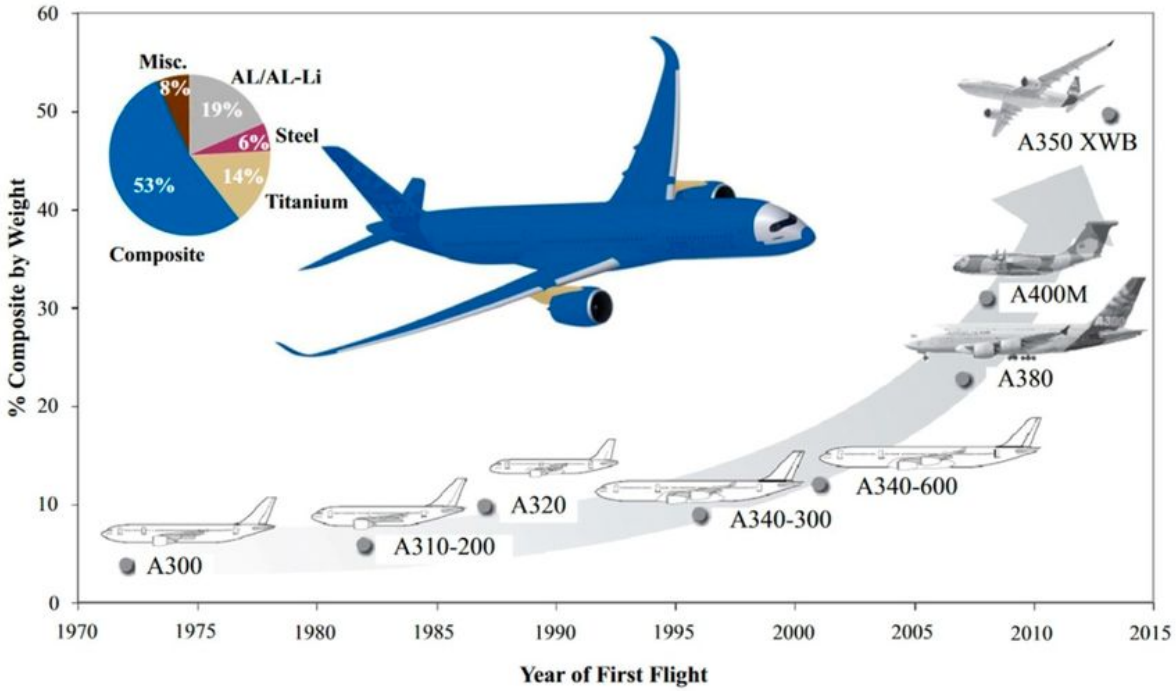
18/07/2025

AIRBUS

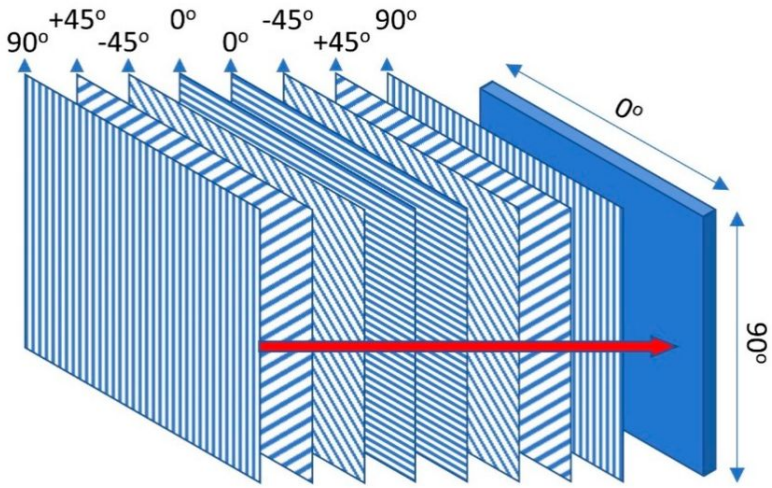
Optimal design of laminated composite structures

Industrial Context

The Rise of Composites



A zoom on the stacking



Constituents: Carbon Fibers carry **Tension**, Matrix ensures **Load Transfer**.

Result: A Unidirectional (UD) Ply.

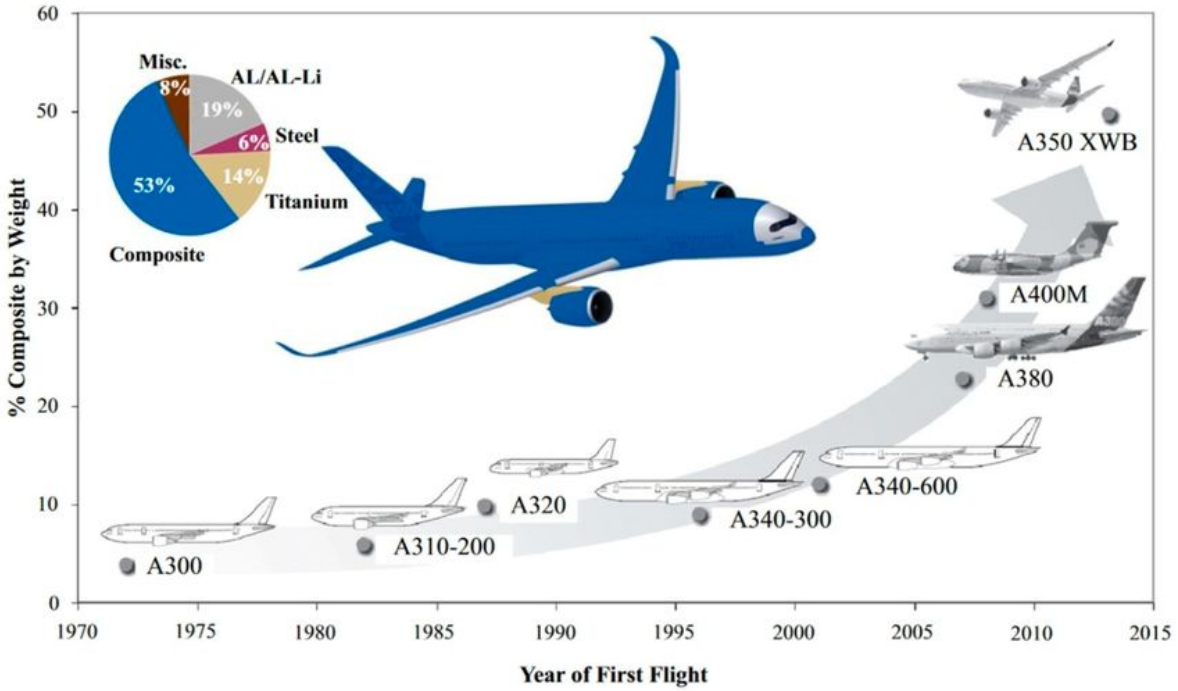
Anisotropy: High strength in Longitudinal **0°**, but weak in Transverse **90° & Shear**.

To resist loads in all directions, we stack plies at specific angles:
Design Variable: *Stacking Sequence*
theta = [0, 45, 90, -45...]

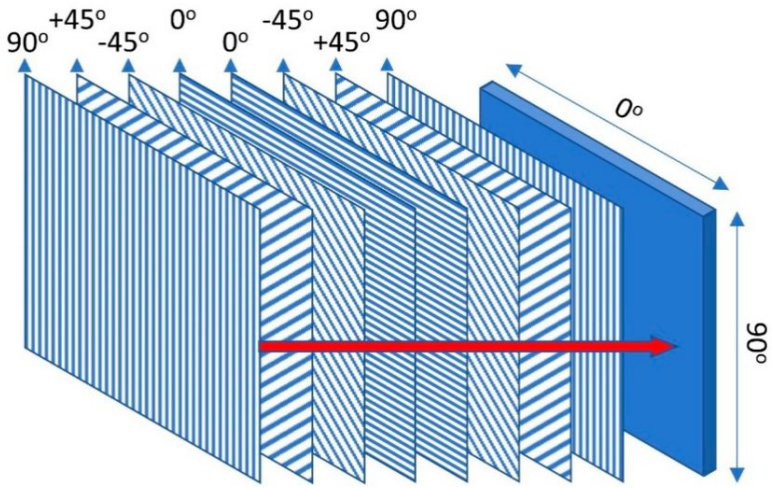
AIRBUS

Industrial Context

The Rise of Composites



A zoom on the stacking



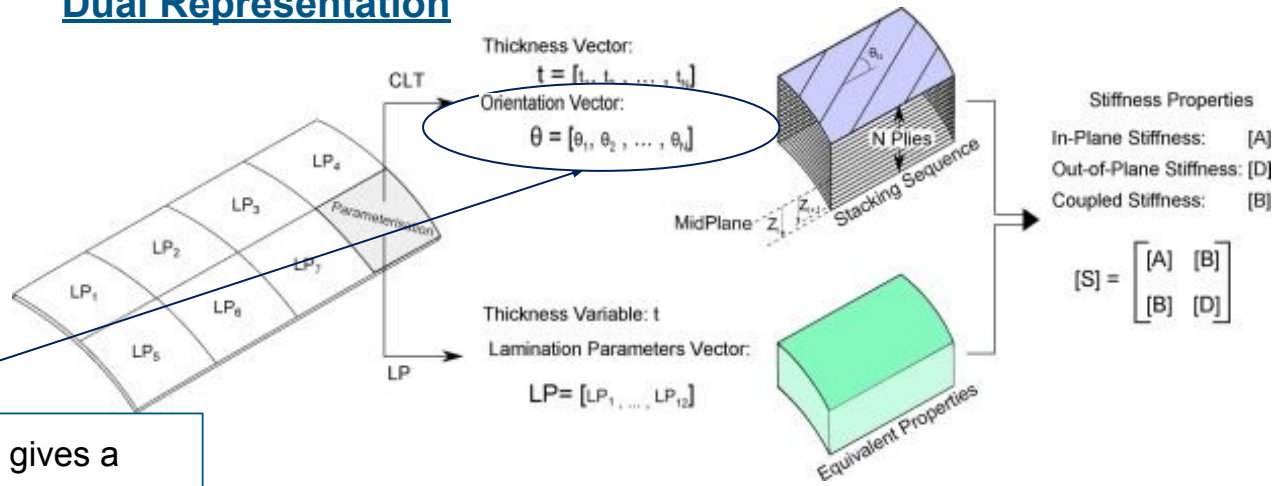
Constituents: Carbon Fibers carry **Tension**, Matrix ensures **Load Transfer**.

Result: A Unidirectional (UD) Ply.

Anisotropy: High strength in Longitudinal **0°**, but weak in Transverse **90° & Shear**.

To resist loads in all directions, we stack plies at specific angles:
Design Variable: *Stacking Sequence*
theta = [0, 45, 90, -45...]

Dual Representation

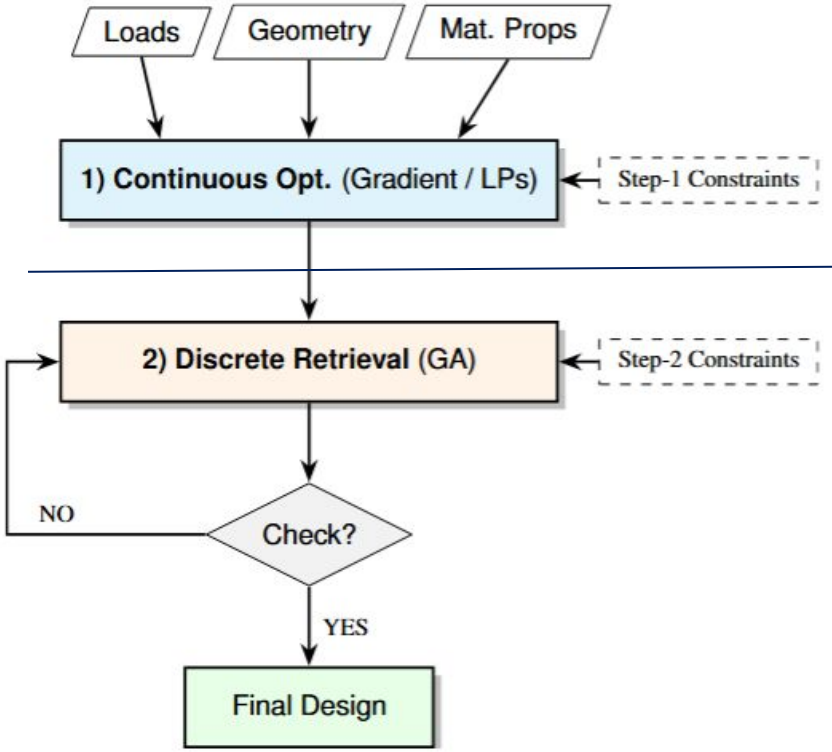


Problem: While one sequence gives a unique set of LPs, the reverse is not true. **Multiple sequences can result in the same LPs.**

AIRBUS

Problem Statement: The Inverse Design Challenge

The Bi-step Optimization process



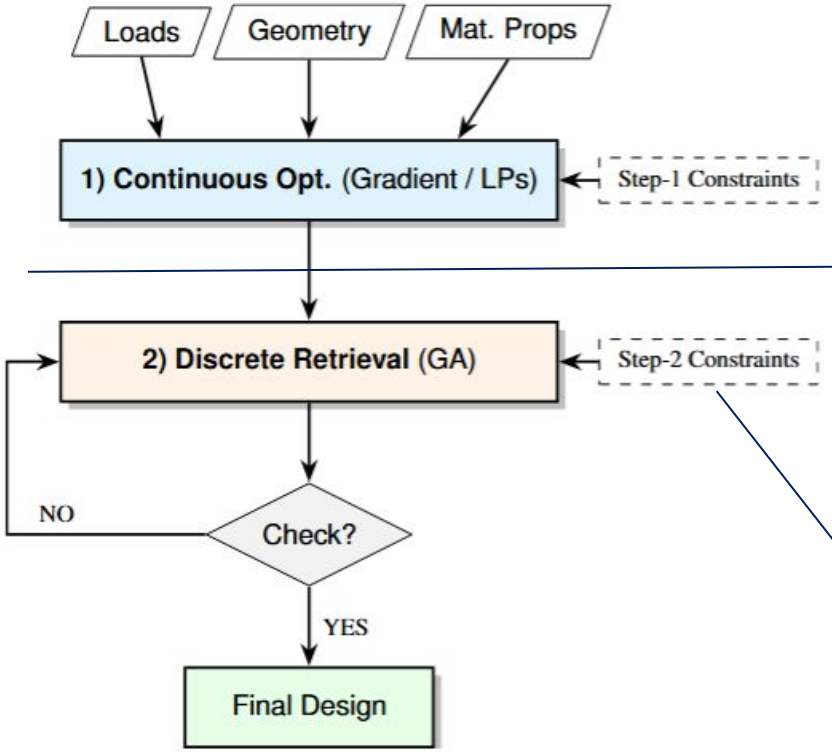
Step 1 : **First Level Problem (FLP - Continuous):** → Fast & Efficient (Convex problem)

Output (step 1) = Input (step2) = The set of optimal lamination parameters and the optimal number of plies for each stacking.

Step 2 : **Second Level Problem (SLP - Discrete):** A "Stacking Sequence Retrieval" step to find actual manufacturing sequences (angles) matching the FLP target. → Slow & Computationally Expensive

Problem Statement: The Inverse Design Challenge

The Bi-step Optimization process



Step 1 : **First Level Problem (FLP - Continuous):** → Fast & Efficient (Convex problem)

Output (step 1) = Input (step2) = The set of optimal lamination parameters and the optimal number of plies for each stacking.

Step 2 : **Second Level Problem (SLP - Discrete):** A "Stacking Sequence Retrieval" step to find actual manufacturing sequences (angles) matching the FLP target. → Slow & Computationally Expensive

Local (Single panel)

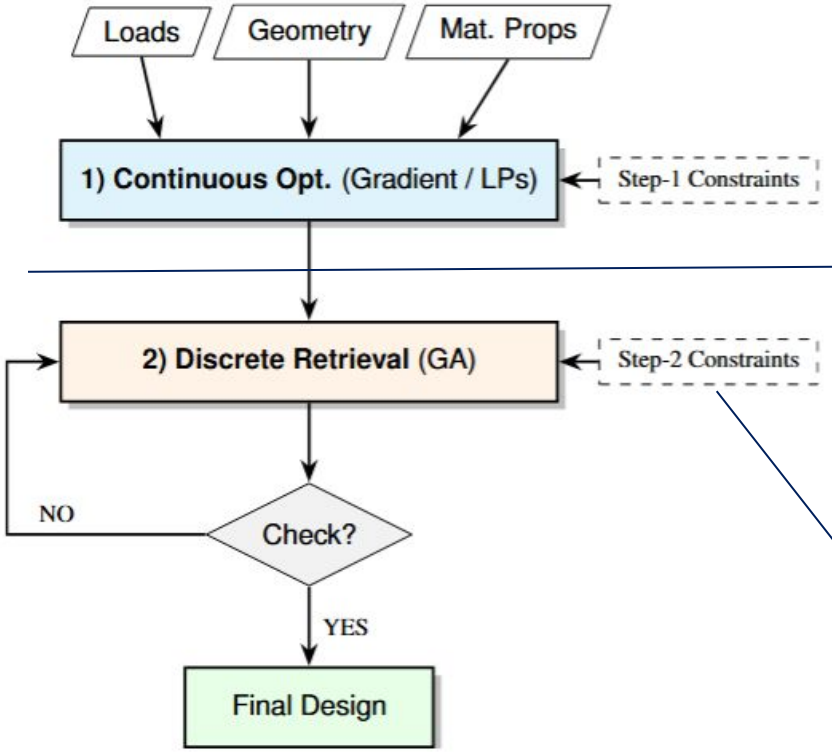
- ➔ Symmetry
- ➔ Contiguity
- ➔ Balance
- ➔ Disorientation
- ➔ Proportion

"Feasible Stacking"

AIRBUS

Problem Statement: The Inverse Design Challenge

The Bi-step Optimization process



Step 1 : **First Level Problem (FLP - Continuous):** → Fast & Efficient (Convex problem)

Output (step 1) = Input (step2) = The set of optimal lamination parameters and the optimal number of plies for each stacking.

Step 2 : **Second Level Problem (SLP - Discrete):** A "Stacking Sequence Retrieval" step to find actual manufacturing sequences (angles) matching the FLP target. → Slow & Computationally Expensive

Local (Single panel)

- ➔ Symmetry
- ➔ Contiguity
- ➔ Balance
- ➔ Disorientation
- ➔ Proportion

Global (Multi panel)

- ➔ Blending

→ Stiffness Properties

"Feasible Stacking"

AIRBUS

Optimal design of laminated composite structures

Single Panel Generation (Local)

SeqGPT

Using Transformers to find a solution for the inverse problem

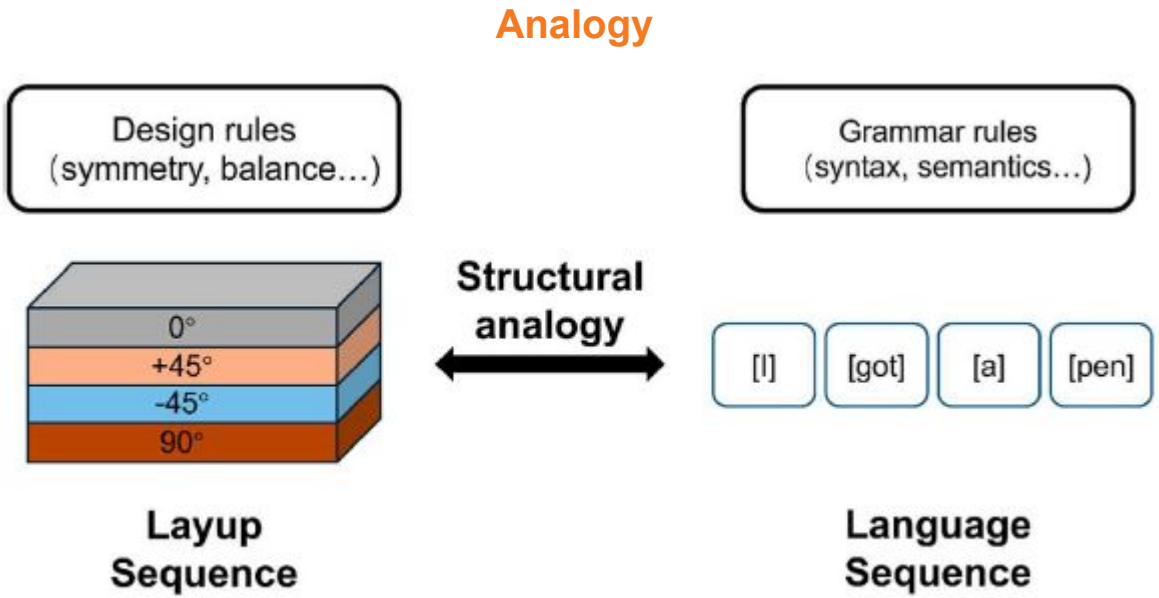


Fig. 1. Conceptual analogy between laminate layup design and language modelling.

Using Transformers to find a solution for the inverse problem

Goal: Use **Generative AI** to generate *feasible* stacking sequences that match target Lamination Parameters LP's.

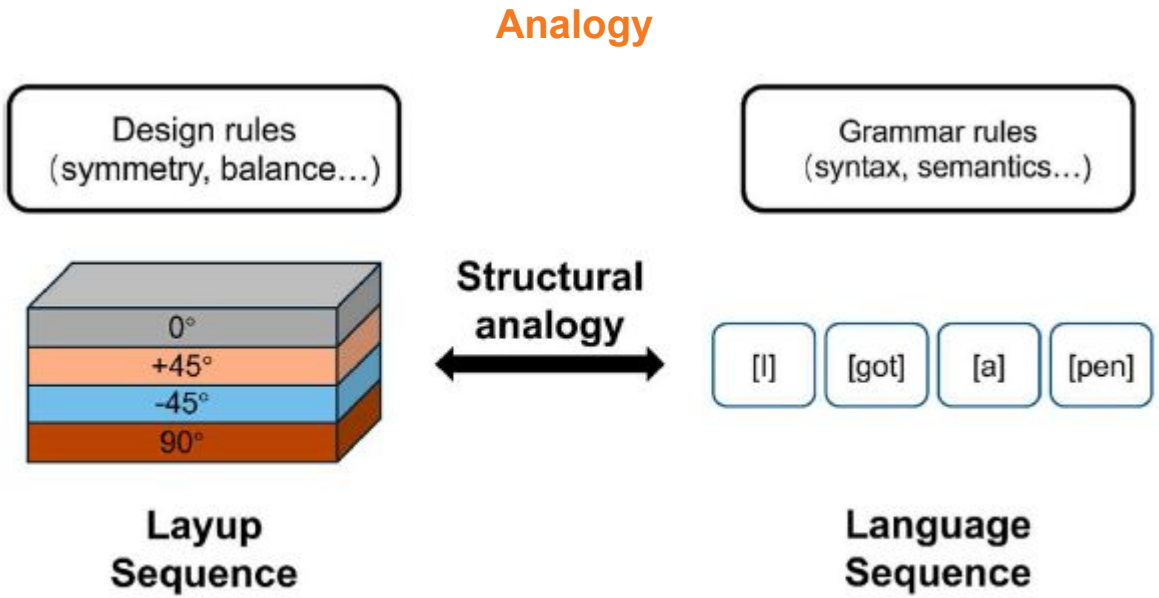
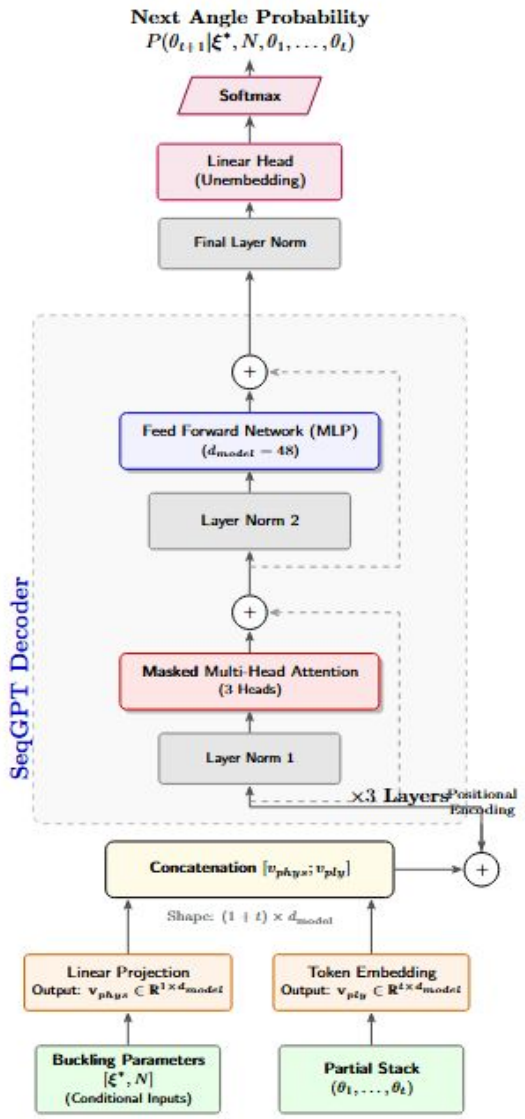


Fig. 1. Conceptual analogy between laminate layup design and language modelling.

Training: Learning Physics through Language Modeling

Next token prediction

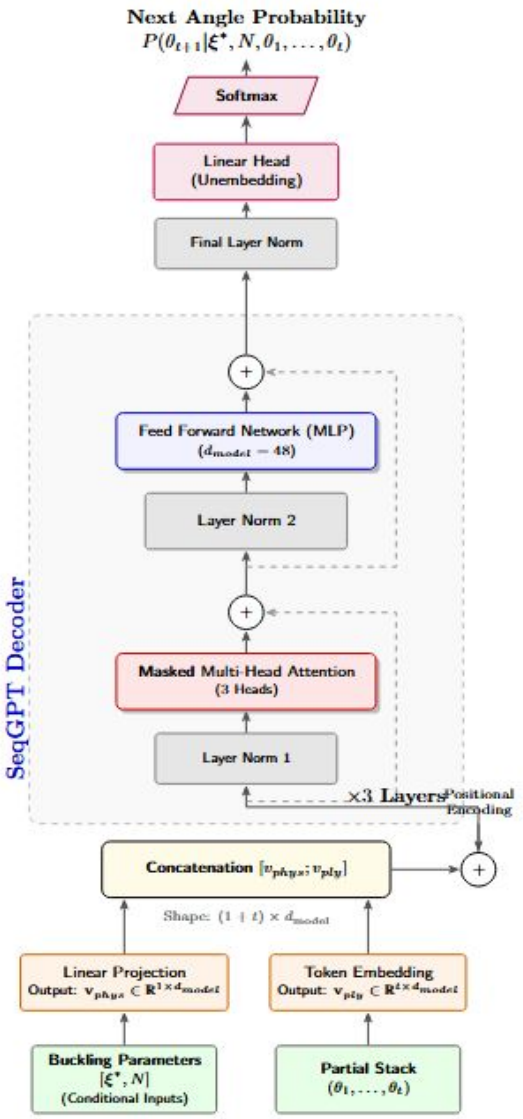
minGPT



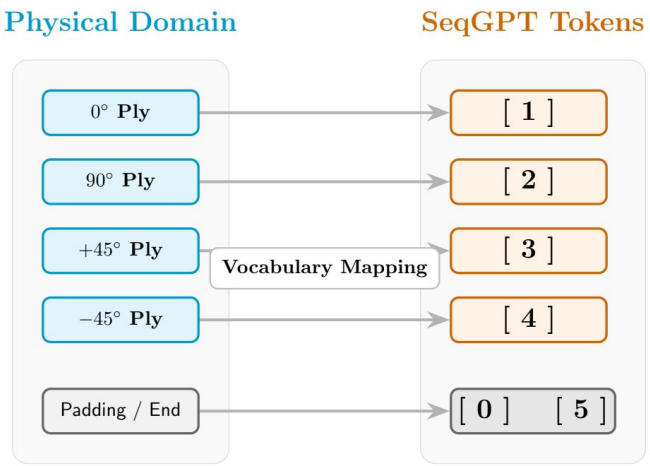
Training: Learning Physics through Language Modeling

Next token prediction

minGPT



Tokenizer

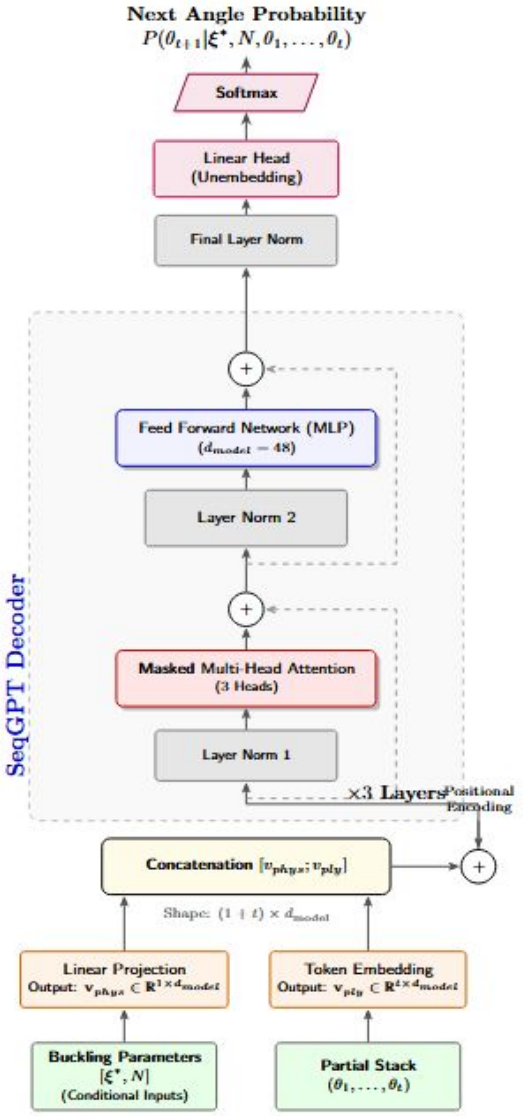


AIRBUS

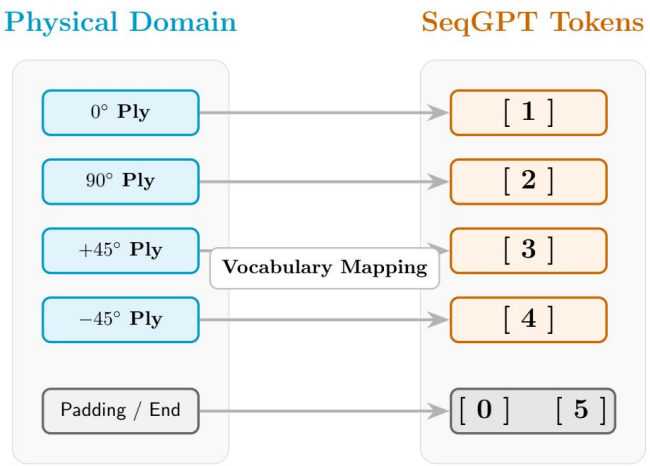
Training: Learning Physics through Language Modeling

Next token prediction

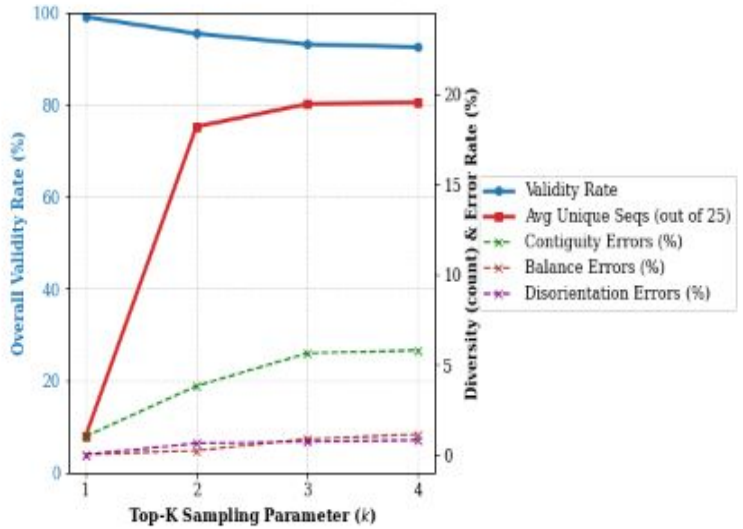
minGPT



Tokenizer



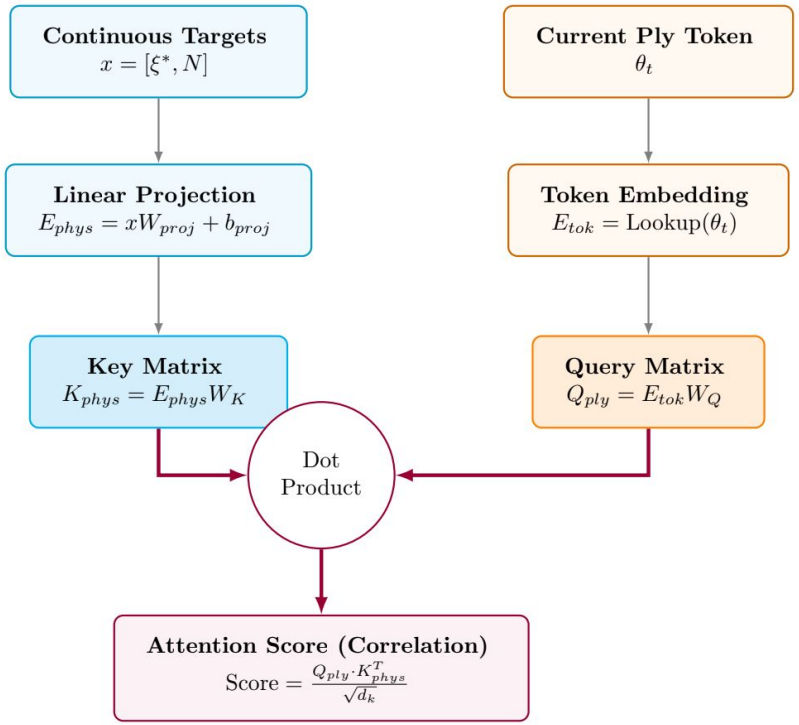
Validity rate



AIRBUS

Interpretability using attention score

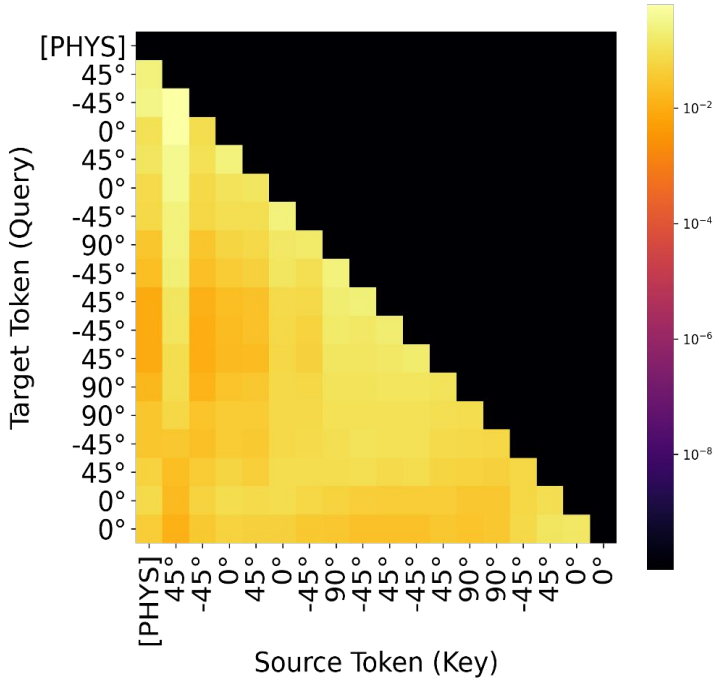
Correlation between plies generation and physical parameters is only due to Attention



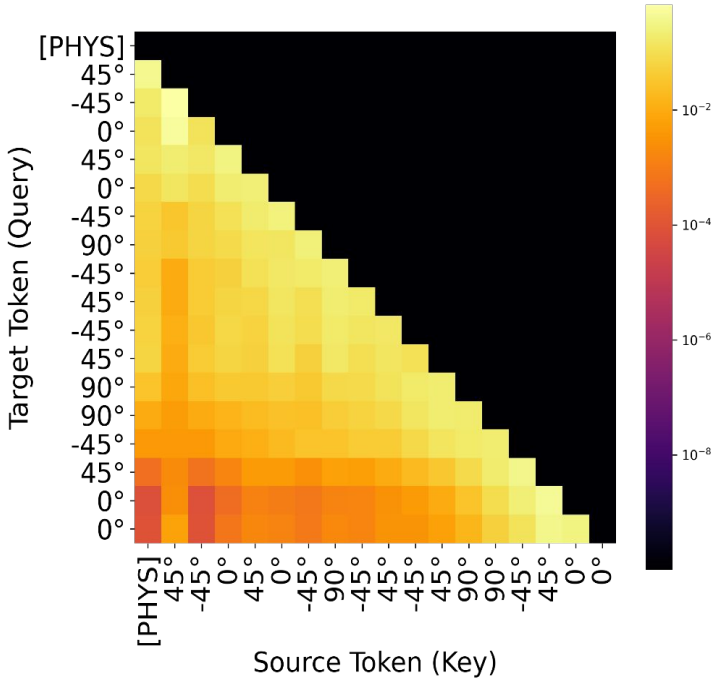
Why are attention mechanisms so powerful ? :

- Flexibility over time
- Flexibility over space
- Parallelization

Global



Local (Markovian constraint)

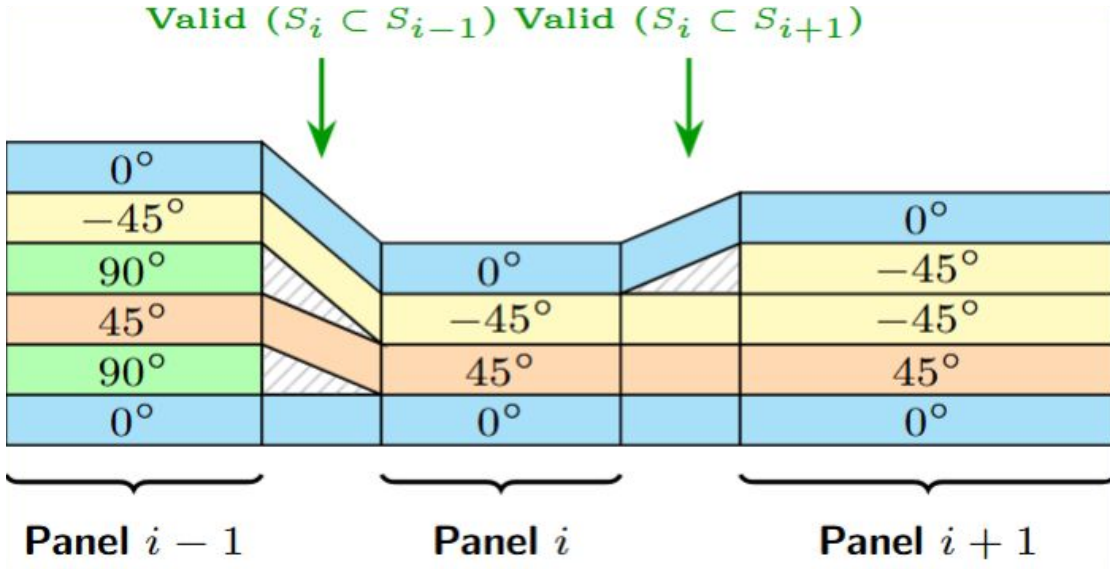


AIRBUS

Optimal design of laminated composite structures

The Blending Constraint

Compatibility



The Blending Constraint :

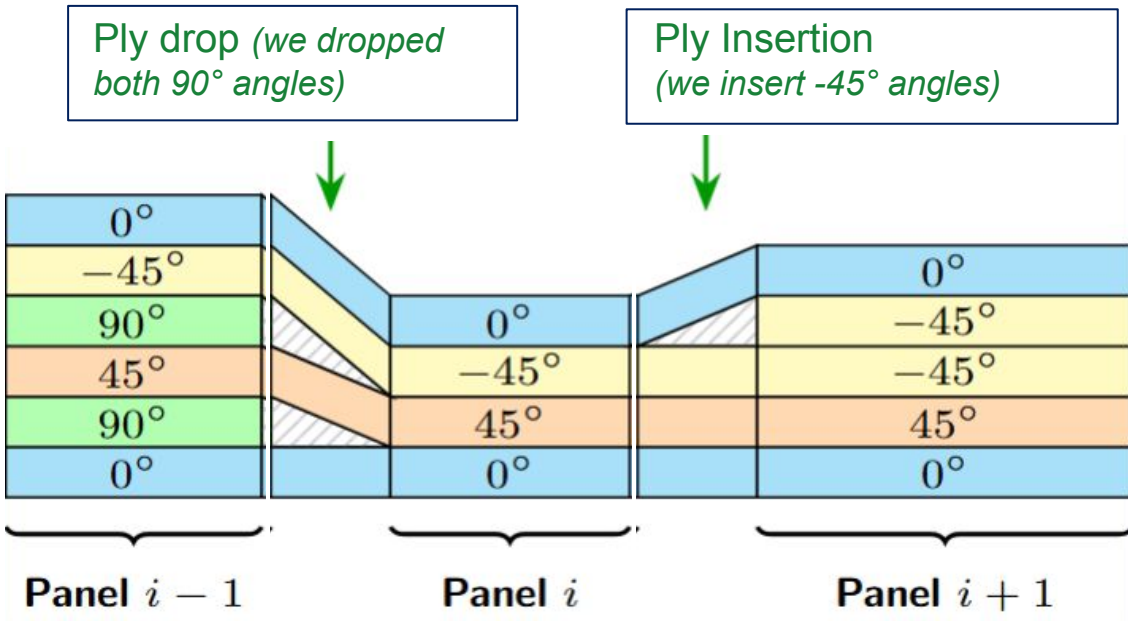
Each ply present in a thinner panel must continue into adjacent thicker panels to ensure a continuous load path across the structure.

AIRBUS

The SPD Approach

Idea

- Generate a Starting Sequence (S)
- Generate Adjacent Panel by Ply drops/Ply insertions.

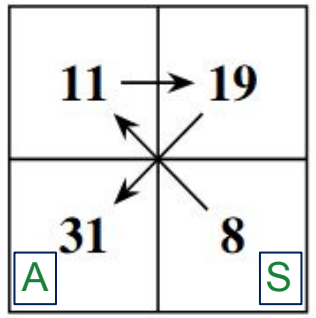


Easy ? But this is in 1D…

Need

In 2D, it is more difficult because the order in which we generate the sequences is important.

➔ We need to define a path *(Propagation Direction)* that guarantees blending by construction.



(a)

N : number of plies

Valid Path

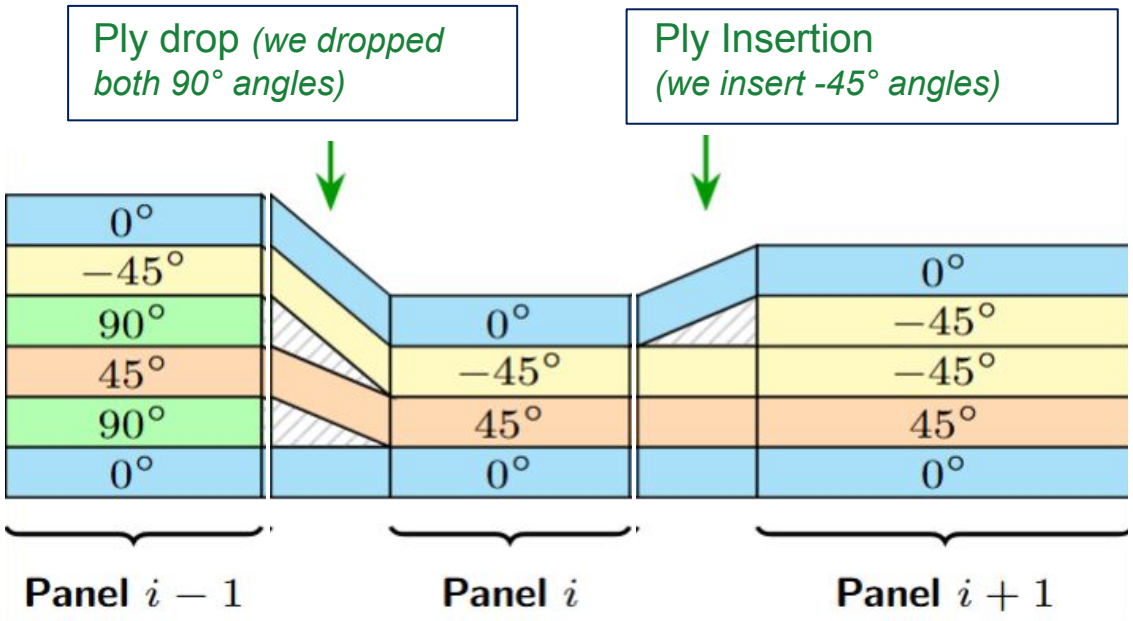
Generating the Thicker (Thinner) Pannel than generating Thinner (Thicker) ones after by Increasing (Decreasing) order.

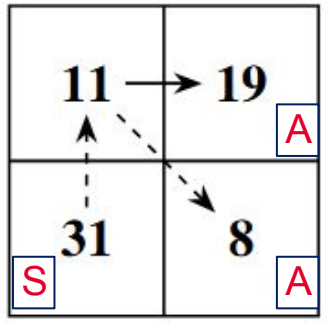
AIRBUS

The SPD Approach

Idea:

- Generate a Starting Sequence (S)
- Generate Adjacent Panel by Ply drops/Ply insertions.





(b)

Valid Path

- N=19 and N=8 are compatible because they are generated from the same parent N=11.
- N=31 and N=8 are compatible because N=8 is a subsequence of N=11 which is a subsequence of N=31.

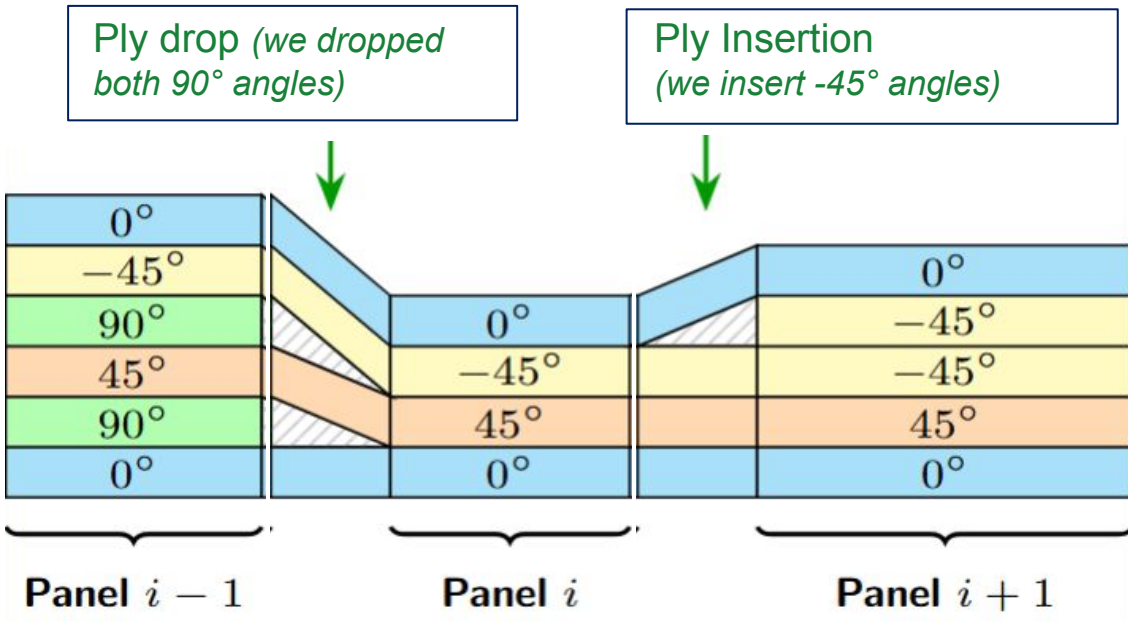
Easy ? But this is in 1D…

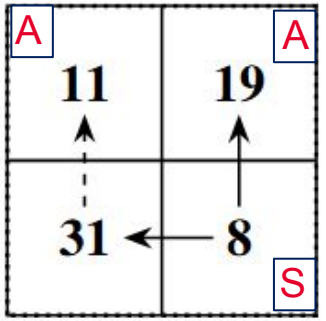
AIRBUS

The SPD Approach

Idea:

- Generate a Starting Sequence (S)
- Generate Adjacent Panel by Ply drops/Ply insertions.





(c)

Invalid Path.
The Generation Path here is not a valid propagation direction. We don't have any guarantee that the sequence N=11 will be compatible with the sequence N=19.

Easy ? But this is in 1D…

AIRBUS

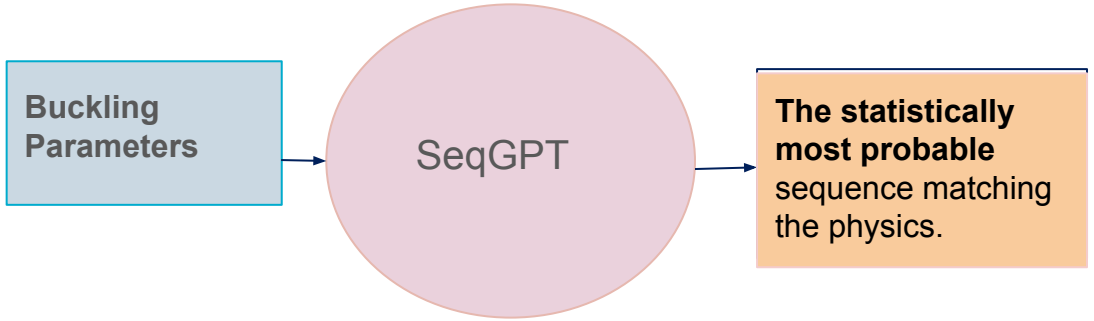
Optimal design of laminated composite structures

Neural Beam Search

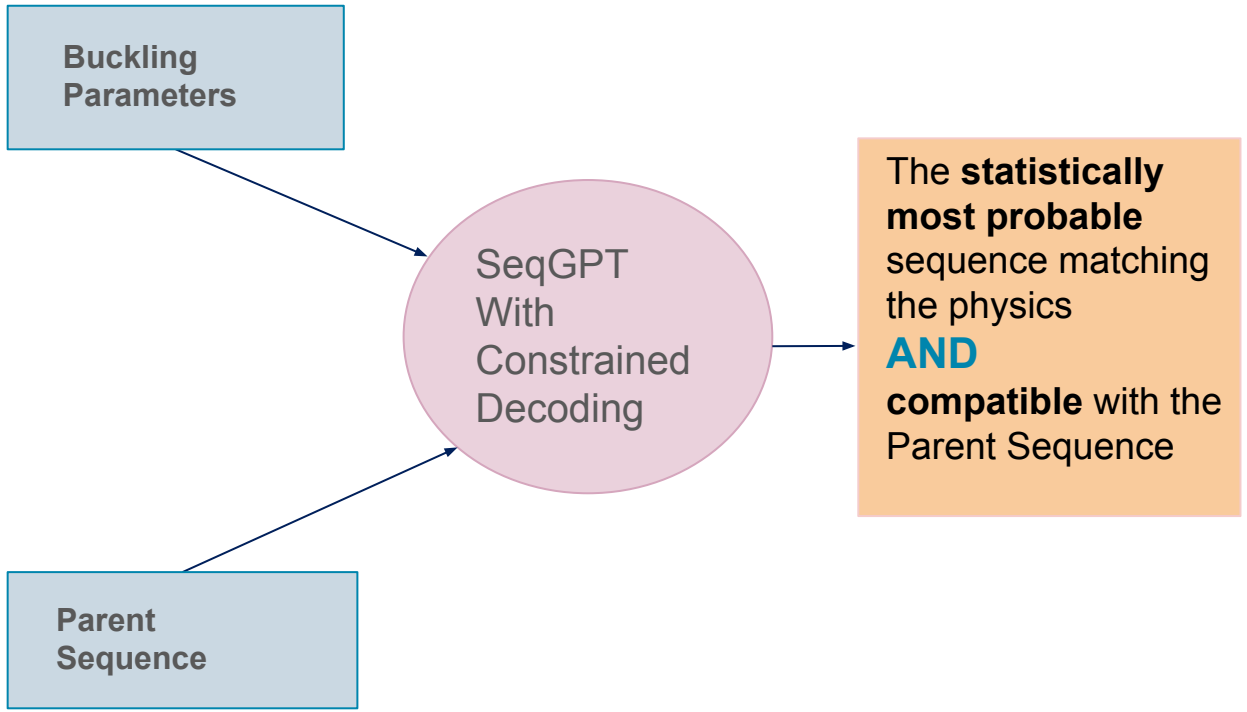
- *ply drop*
- *ply insertion*

From Independent Generation to Constrained Decoding : Idea

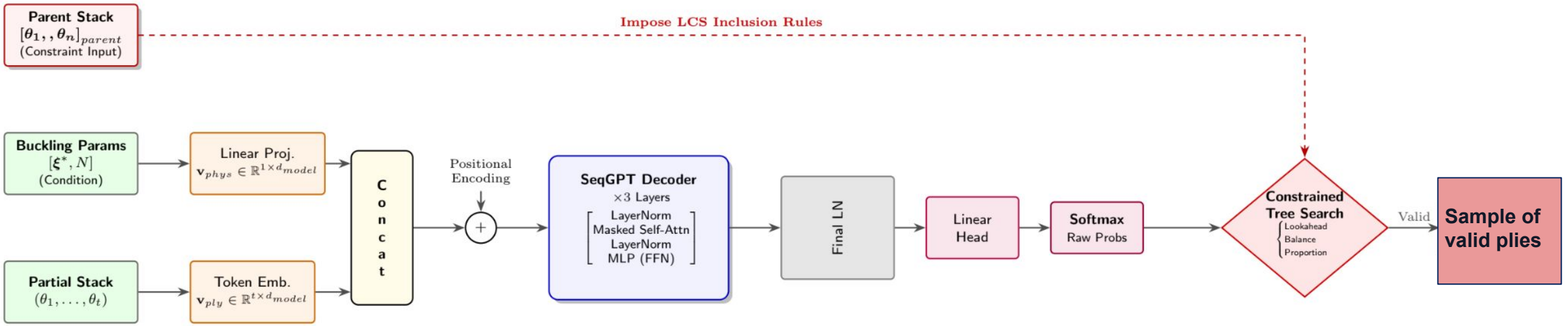
What we have developed for a single panel



What we need for the multi-panel case using SPD



From Independent Generation to Constrained Decoding : Architecture



If one of those rules is broken the final sequence will not be feasible or compatible with the parent sequence

AIRBUS

Optimal design of laminated composite structures

Horseshoe Benchmark

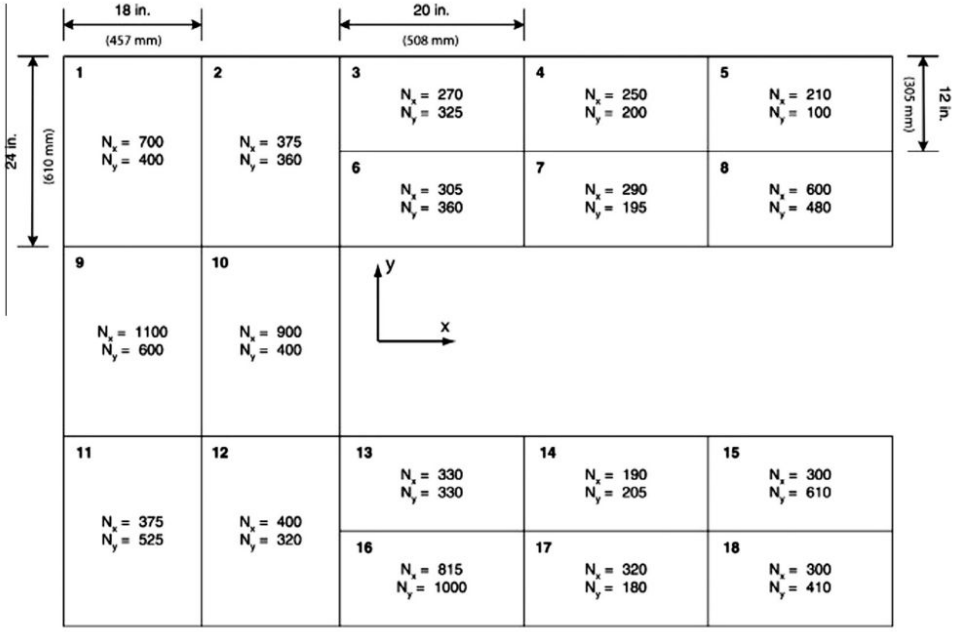
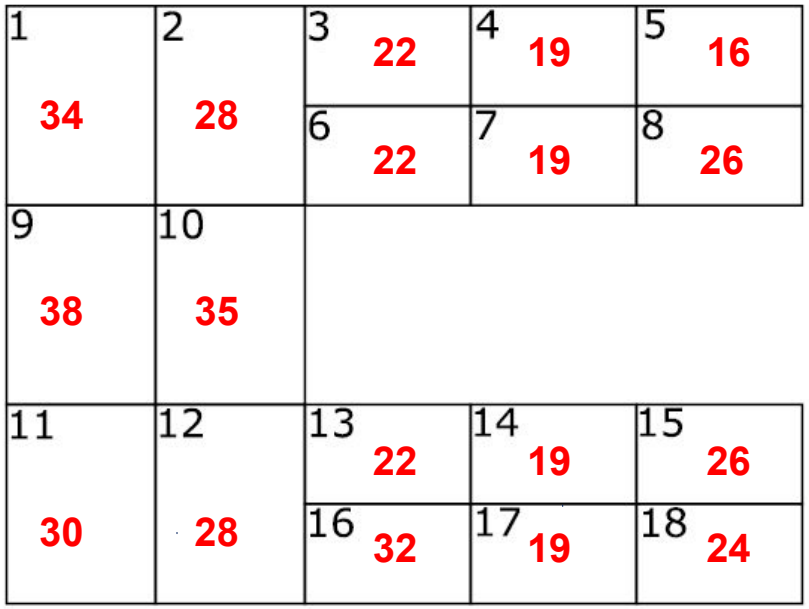
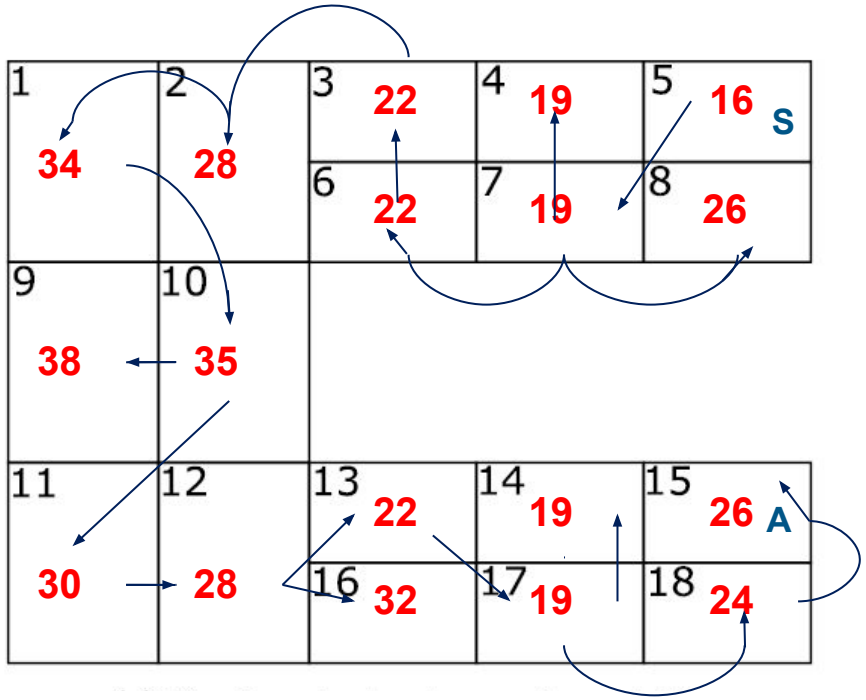


Fig. 8. Eighteen-panel test problem [8,26], all loads in lbf/in (×175.1 for N/m).

SPD definition



(a) Benchmark structure and panel IDs



Generation with **SeqGPT + Beam Search**

1 First Step Optimization
(we don't plot the nemeth parameters here)

2 Propagation maps definition

3 Second step Retrieval

AIRBUS

Results with SPD

$$\lambda_{(m,n)} = \frac{\pi^2 \left[D_{11}\left(\frac{m}{a}\right)^4 + 2(D_{12}+2D_{66})\left(\frac{m}{a}\right)^2\left(\frac{n}{b}\right)^2 + D_{22}\left(\frac{n}{b}\right)^4 \right]}{\left(\frac{m}{a}\right)^2 N_x + \left(\frac{n}{b}\right)^2 N_y} \qquad (*)$$

$$\text{Reserve Factor (RF)} = \min_{m,n} \lambda_{(m,n)}$$

Delta plies = 0

Training time of seqGPT :
5-10 min

Inference time to generate this table:
4.5 seconds

Panel	Target	SeqGPT (Proposed)		EA Reference
	N_{target}	$N_{generated}$	Margin (%)	Margin (%)
1	34	34	16.1	16.6
2	28	28	3.1	3.7
3	22	22	13.4	14.5
4	19	19	6.6	7.7
5	16	16	5.3	6.5
6	22	22	1.9	2.9
7	19	19	3.2	4.3
8	26	26	8.5	9.6
9	38	38	3.4	4.8
10	35	35	3.6	4.5
11	30	30	10.4	10.3
12	28	28	4.3	3.3
13	22	22	2.7	7.8
14	19	19	9.4	14.3
15	26	26	6.2	6.2
16	32	32	8.5	7.9
17	19	19	1.3	5.8
18	24	24	16.2	20.5

(*) François-Xavier Irisarri, Agathe Lasseigne, Fabien Simon, and Rodolphe Le Riche. Evolutionary optimization of stacking sequence tables for composite structures with blending and ply-drop constraints. Journal of Composite Materials, 48(26):3285–3303, 2014.

Optimal design of laminated composite structures

References :

[1] Optimization of Long Anisotropic Laminated Fiber Composite Panels with T-Shaped Stiffeners

[2] Optimal design of laminated composite structures with ply drops using stacking sequence tables

[3] NEUROLOGIC DECODING: (Un)supervised Neural Text Generation with Predicate Logic Constraints

[4] The design of blended laminates regardless of the stack: The search propagation direction

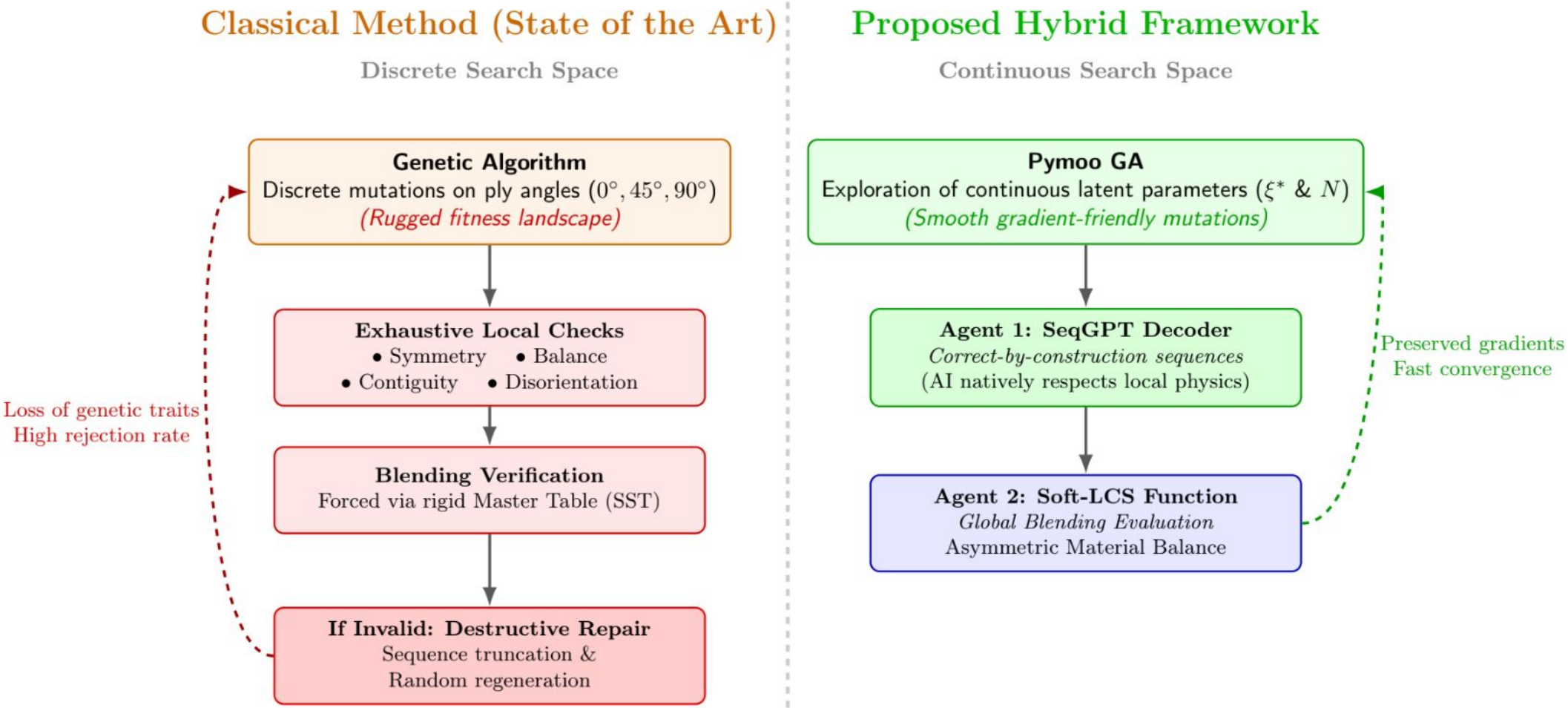
Annexes

Optimal design of laminated composite structures

Another method without the SPD

A quick recall on genetic algorithm

Multiagent Optimization



45min-1 Hour
Use of an SST

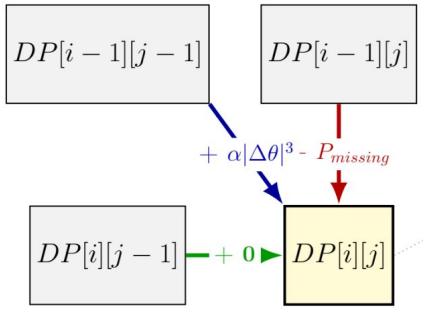
10 min
Multiple families of solutions

AIRBUS

Multiagent Optimization

Agent 2: Soft-LCS Function
Global Blending Evaluation
Asymmetric Material Balance

Soft-LCS: Asymmetric Dynamic Programming for Blending



1. Skip in Thick Panel (Ply Drop) Cost = 0
Standard ply termination. The thin laminate is a subset of the thick one.

2. Ply Substitution (Alignment) Cost = $\alpha|\Delta\theta|^3$
Penalizes angular mismatch cubically to preserve a **smooth gradient**.

3. Skip in Thin Panel (Missing) Cost = $P_{missing}$
Physical Error: Thin panel contains a ply missing in the thick panel.

General Bellman Equation:

$$DP[i][j] = \min \begin{cases} DP[i-1][j-1] + \alpha|S_1[i] - S_2[j]|^3 \\ DP[i][j-1] + 0 \\ DP[i-1][j] + P_{missing} \end{cases}$$

Conservation of Material for the Thickest Sequence

$$\mathcal{P}_{mat} = \sum_{\theta \in \Theta} \max\left(0,\ N_\theta^{thin} - N_\theta^{thick}\right) \times P_{missing} \times C_{rarity}(\theta)$$

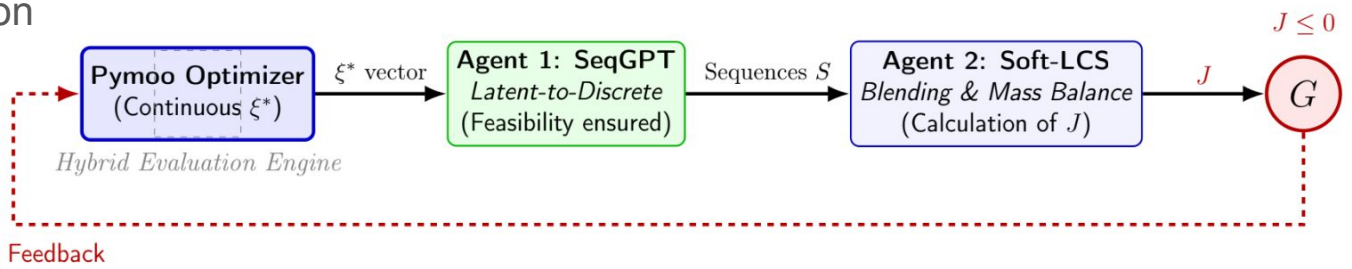
N_θ : Number of plies at angle θ $P_{missing}$: Base penalty C_{rarity} : Structural multiplier

$$\sum$$

$$J = D(m,n) + P_{mat}$$

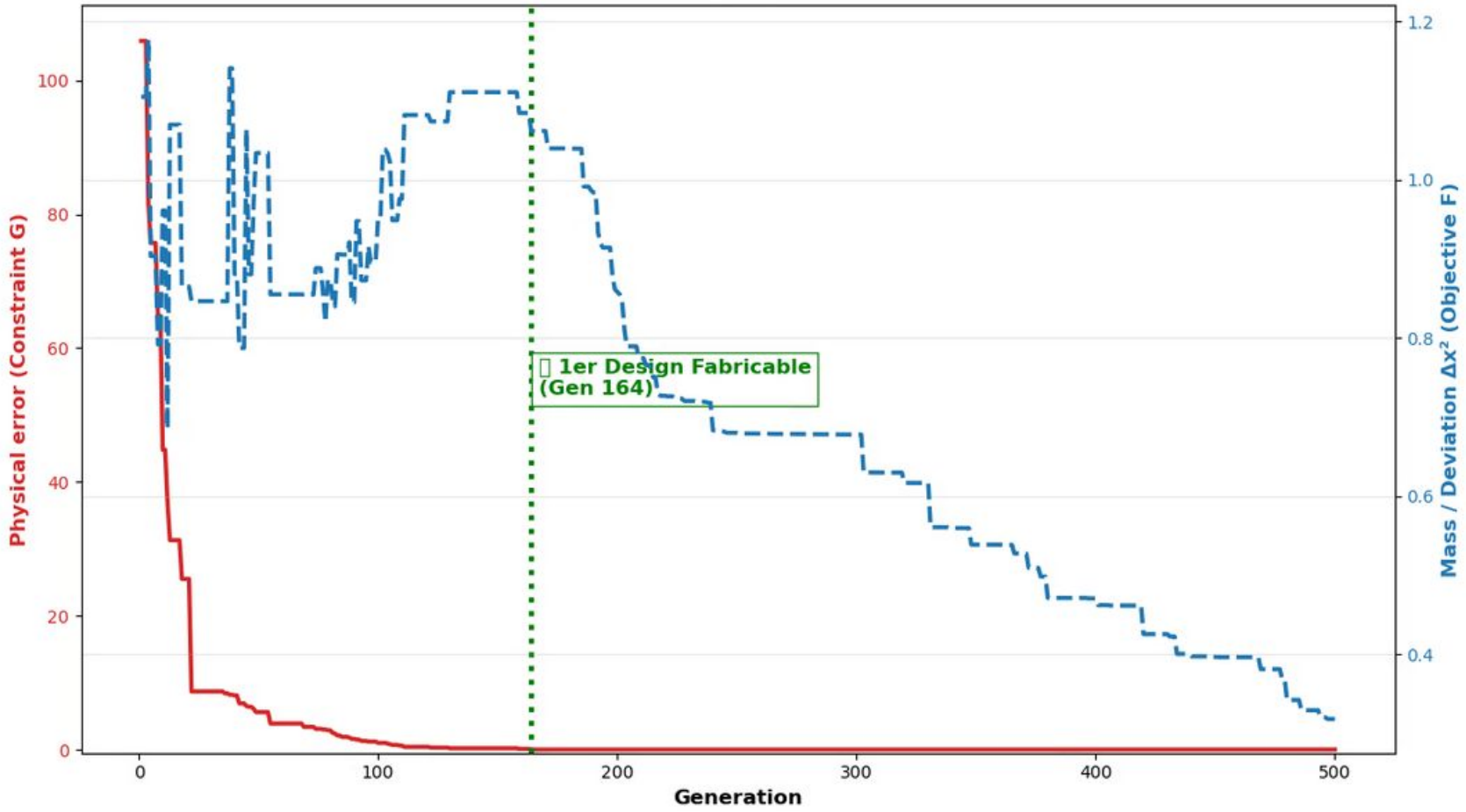
Objective function

$$F = \mu * \sum \Delta\xi^{*2}$$



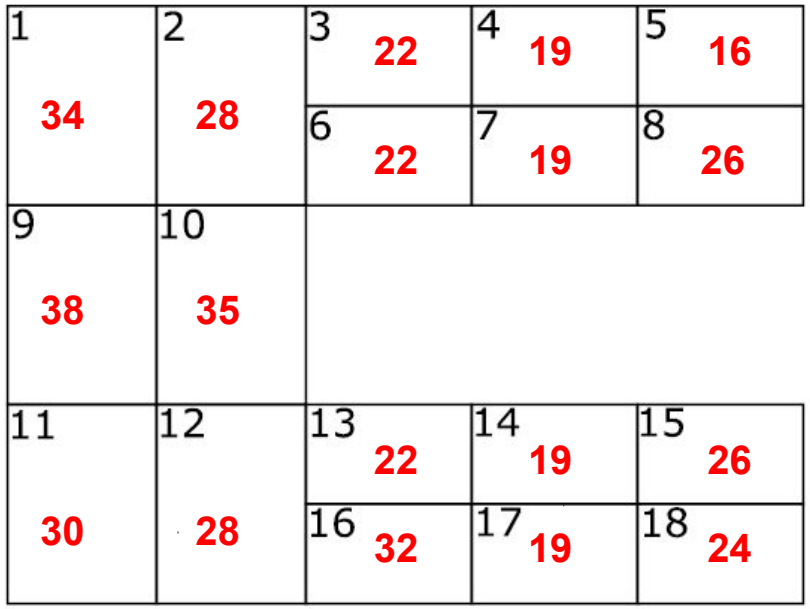
AIRBUS

Multiagent Optimization



AIRBUS

Multiagent Optimization



(a) Benchmark structure and panel IDs

Problem..

Panel_ID	Target_N	total_plies
1	34	34
2	28	28
3	22	22
4	19	23
5	16	16
6	22	34
7	19	19
8	26	26
9	38	38
10	35	35
11	30	30
12	28	28
13	22	32
14	19	19
15	26	32
16	32	38
17	19	23
18	24	24

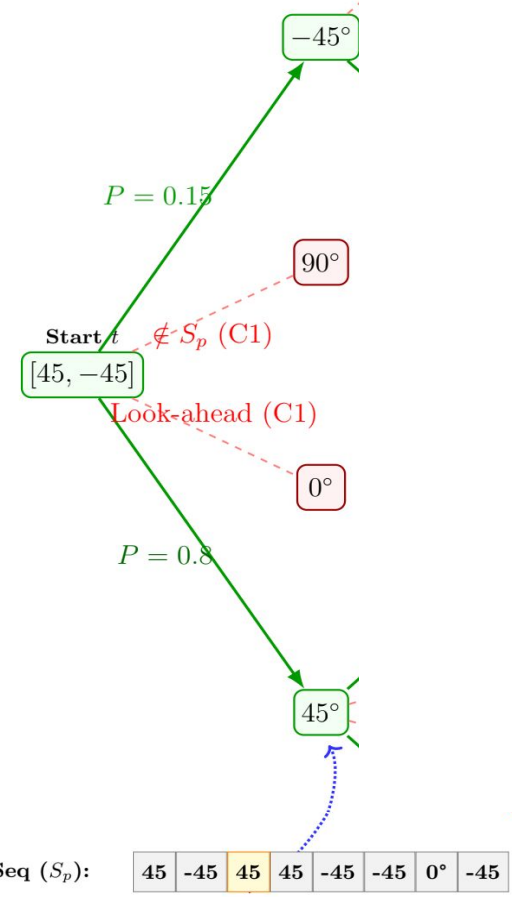
edges = [(0, 1), (0, 8), (1, 9), (1, 2), (1, 5), (8,9), (8,10), (9,11), (2,5), (2,3), (5,6), (3,6), (3,4), (4,7), (10,11), (11,12), (11,15), (12,13), (12,15), (15,16), (13,16), (13,14), (14,17)]

AIRBUS

From Independent Generation to Constrained Decoding : An example

Rank	Seq $(t+1)$	Sc.
1	$[\ldots, 45, 45]$	**3.4**
2	$[\ldots, 45, -45]$	2.7
3	$[\ldots, -45, 45]$	2.4
4	$[\ldots, -45, -45]$	*1.9*

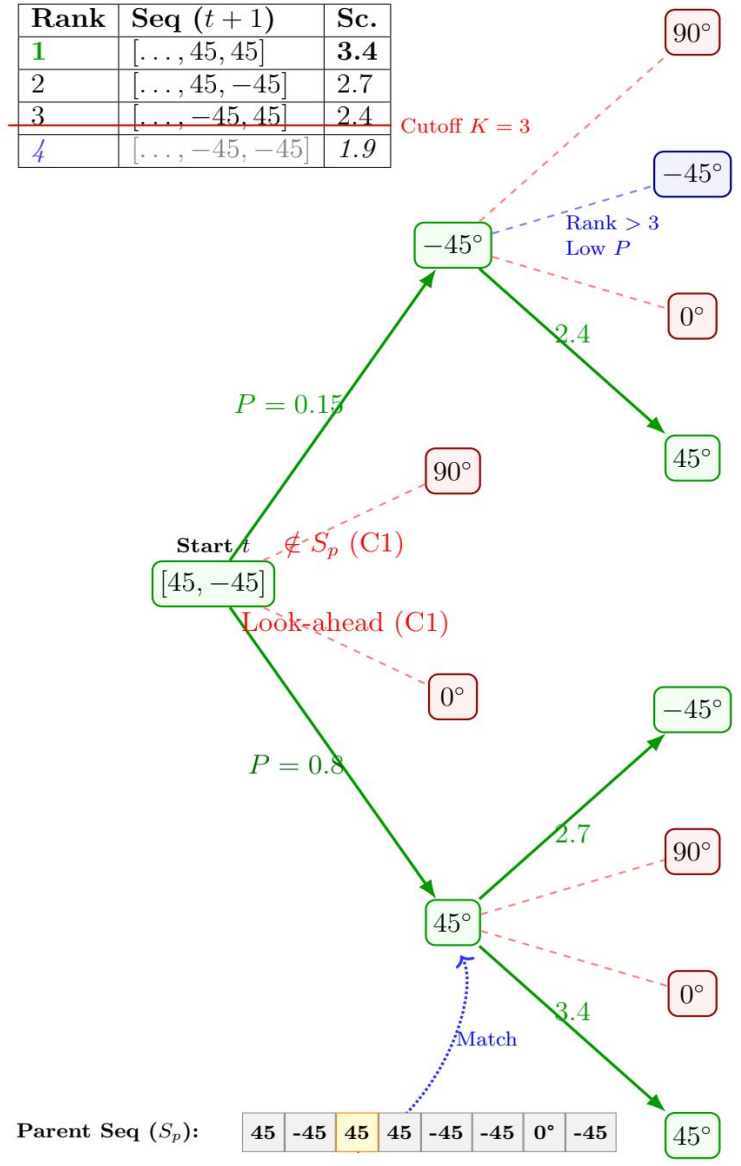
Cutoff K



AIRBUS

From Independent Generation to Constrained Decoding : An example

Rank	Seq $(t+1)$	Sc.
1	$[\ldots, 45, 45]$	**3.4**
2	$[\ldots, 45, -45]$	2.7
3	$[\ldots, -45, 45]$	2.4
4	$[\ldots, -45, -45]$	*1.9*



AIRBUS

Annexes

More details on seqGPT results

Optimal design of laminated composite structures

Benchmark of three different solutions

Comparison between three different approaches

Feature	Irisarri et al. (2014)	Scardaoni & Montemurro (2022)	SeqGPT
Paradigm	Monolithic Global Optimization	Optimization Bi-step	Optimization Bi-step
Phase 1 (Continuous)	N/A	Continuous parameter optimization	Continuous parameter optimization
Phase 2 (Discrete)	Evolutionary Algorithm with Repair	Multi-start Non-Linear Optimization (guided by SPD)	Neural Beam Search (using seqgpt)
Blending	Costly constraint within the loop	Constructed via SPD	Constructed via SPD
Time Complexity Analysis	$O(N_{gen} \cdot N_{pop})$	$O(N_{starts} \cdot N_{iter})$	$O(N_{panel} * N_{steps} * B)$
Scalability ?	**Combinatorial Explosion:** Thicker sequences exponentially expand the discrete search space 4^N. To maintain **constant quality**, Ngen and Npop mus increase.	**Dimensionality Curse:** Thicker sequences expand the search space, drastically increasing the density of **local minima**. **=>** To avoid premature convergence, N_start and N_iter must increase non-linearly, leading to **Unpredictable Performance**.	**Linear Scalability:** The constructive approach is immune to combinatorial explosion. The inference path is direct. **=>** Doubling the thickness exactly doubles the inference time, guaranteeing **predictable performance** with **no hidden costs**
Time Complexity Analysis (Benchmark example)	$T_{SLP\_Irisarri} \approx \underbrace{N_{gen}}_{8\,000} \times \underbrace{N_{pop}}_{30} \times (\underbrace{T_{repair} + T_{FEM}}_{\text{Simulation } (\approx 0.015s)})$ $\approx \mathbf{3\,600s}\ (1\ \text{h})$	$T_{SLP\_SPD} \approx [\underbrace{N_{starts}}_{50-100} \times \underbrace{N_{iter}}_{200} \times \underbrace{T_{\mathcal{R}}}_{0.03s}]$ $\approx \mathbf{300s - 600s}\ (\text{Minutes})$	$T_{SLPGPT} \approx \underbrace{T_{init}}_{30ms} + \underbrace{N_{panel}}_{18} \times \underbrace{N_{steps}}_{21} \times (\underbrace{T_{GPU}}_{3ms} + \underbrace{B}_{30} * \underbrace{T_{CPU}}_{0.1ms})$ $\approx 2.3s$

From Independent Generation to Constrained Decoding

What Standard SeqGPT Does	What we need for blending (using SPD)
Input : • Targets $\alpha, \beta, \gamma, \delta$ -> Buckling parameters • Target Thickness N	Input : • Targets $\alpha, \beta, \gamma, \delta$ • Target Thickness N • Parent sequence S_parent
Process : ● Generates **the statistically most probable** sequence matching the physics.	Process : ● Must generate the best sequence that is strictly **a subset** of S_parent (case N <= len(S_parent)) or **a supersequence** of S_parent (case N > len(S_parent)) AND The **statistically most probable** sequence matching the physics.

Optimal design of laminated composite structures

SPD Approach

[2] The design of blended laminates regardless of the stack: The search propagation direction
Marco Picchi Scardaoni a, Marco Montemurro b,

Inspiration: Constrained Decoding via Logic (Lu et al., 2021) [3]

Core Idea: Instead of retraining the model to learn complex rules, we constrain the **inference process** (Decoding).

Inspiration: *NeuroLogic Decoding* (Lu et al., 2021).

- Goal of the example: Force a language model to include specific words (logic constraints) without retraining.
- Method: **Constrained Beam Search** with **Lookahead Pruning**. If a branch cannot satisfy the logic in the future, it is cut immediately.

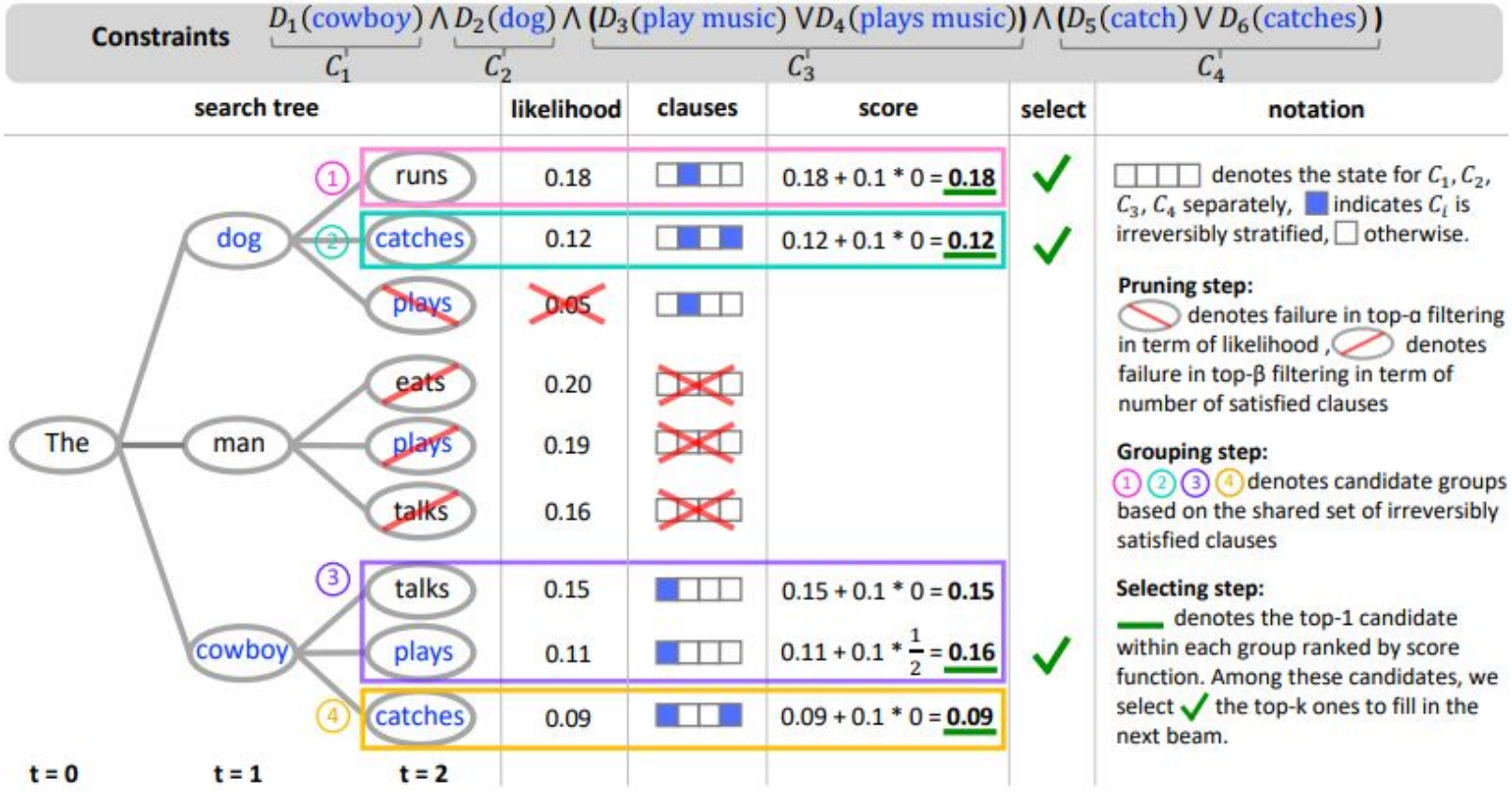


Illustration of the NEUROLOGIC decoding procedure. In this example, $k = 3$, $\alpha = 8$, $\beta = 2$, $\lambda = 0.1$

Adaptation: Constrained Decoding with seqGPT

We transpose this logic from **Semantics** to **Topology**.

First filter based on the constraints

Hard Constraints

C1	SST Topology : We must ensure this ply exists in the remaining parent sequence **AND** that choosing it leaves enough plies to reach the target thickness.
C2	Balance : We verify that the remaining parent sequence contains a sufficient inventory of counter-angles to resolve any orientation imbalance before the end of the stack
C3	Proportion : we must have at least 10 % (min_percent) of each angles

Soft Constraints

SC 1	We lower the likelihood of choosing an angle if we have to skip many others to get to it

Second filter based on the score

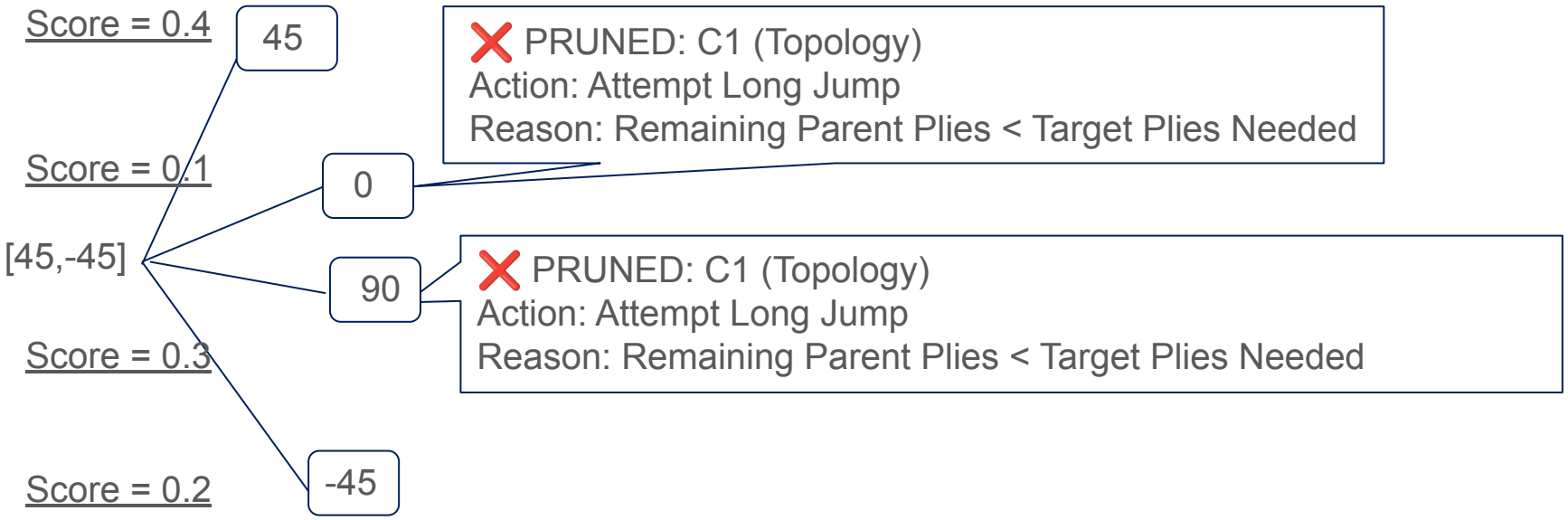
$$\text{Score}_t = \underbrace{\text{Score}_{t-1}} + \underbrace{\log(P_\theta(\text{token}))} - \underbrace{\text{Penalty}}$$

Beam search schema

Constrained Decoding with seqGPT : Ply drop Example

Parent Sequence : [45, -45, 45, 45, -45, 45, -45, 0, -45, -45, 45, 90, -45] N = 13

Generated Sequence : [45, -45, 45, 45, -45, 0, -45, -45, 45, 90] N = 10



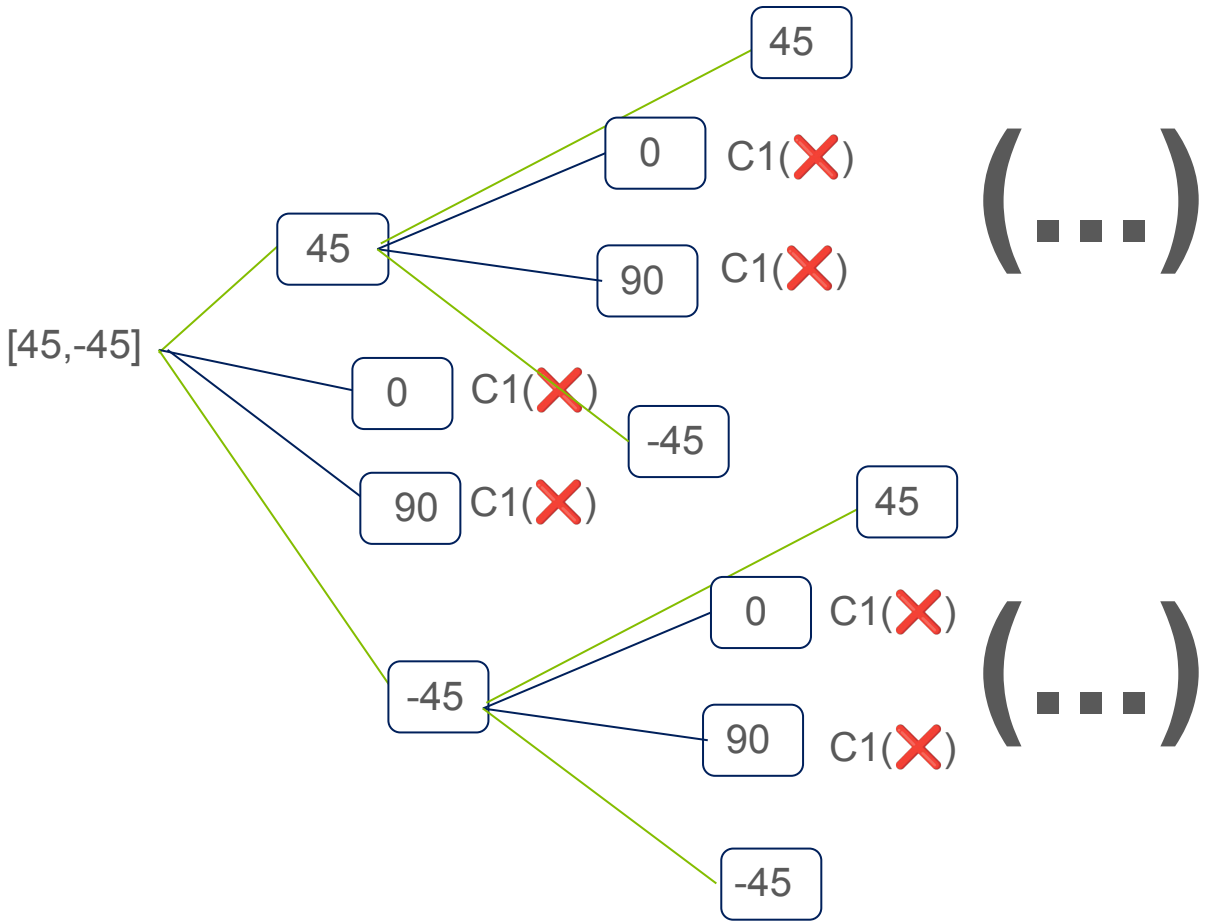
AIRBUS

Constrained Decoding with seqGPT : Ply drop Example (Second Filter

Parent Sequence : [45, -45, 45, 45, -45, 45, -45, 0, -45, -45, 45, 90, -45] N = 13

$$\text{Score}_t = \underbrace{\text{Score}_{t-1}} + \underbrace{\log(P_\theta(\text{token}))} - \underbrace{\text{Penalty}}$$

Generated Sequence : [45, -45, 45, 45, -45, 0, -45, -45, 45, 90] N = 10

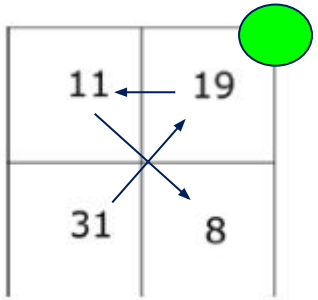


Candidates	Sequence	Cumulative Score
1	[45, -45, 45, 45, -45, 0, -45, -45, 45, 90]	3,4
2	[45, -45, 45, 45, -45, 0, -45, 45, 90, -45]	2,7
3	[45, -45, 45, 45, 0, -45, -45, 45, 90,-45]	2,4
4	[45, 45, 45, -45, 0, -45, -45, 45, 90, -45]	1,9

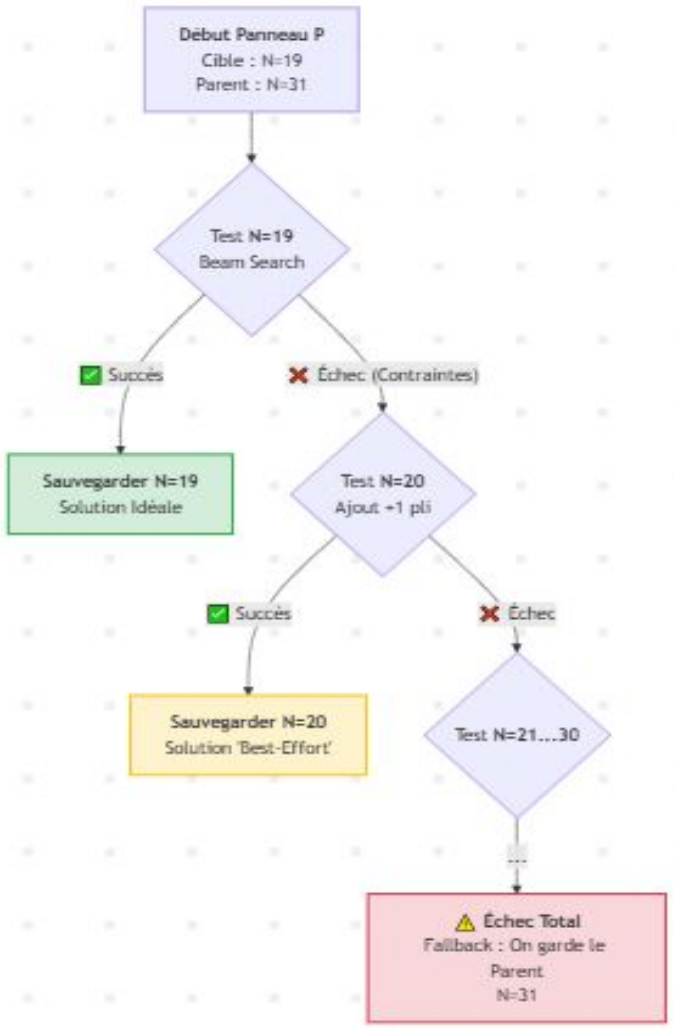
AIRBUS

SST : Ply-drop (Inspired from Irisarri work) [2]

Nmax = 31 Independent variables



Ply-drop : A particular spd



Note (*) :

- Irisarri imposed a rigid Master Guide because Genetic Algorithms could not handle the combinatorial explosion of a flexible topology.

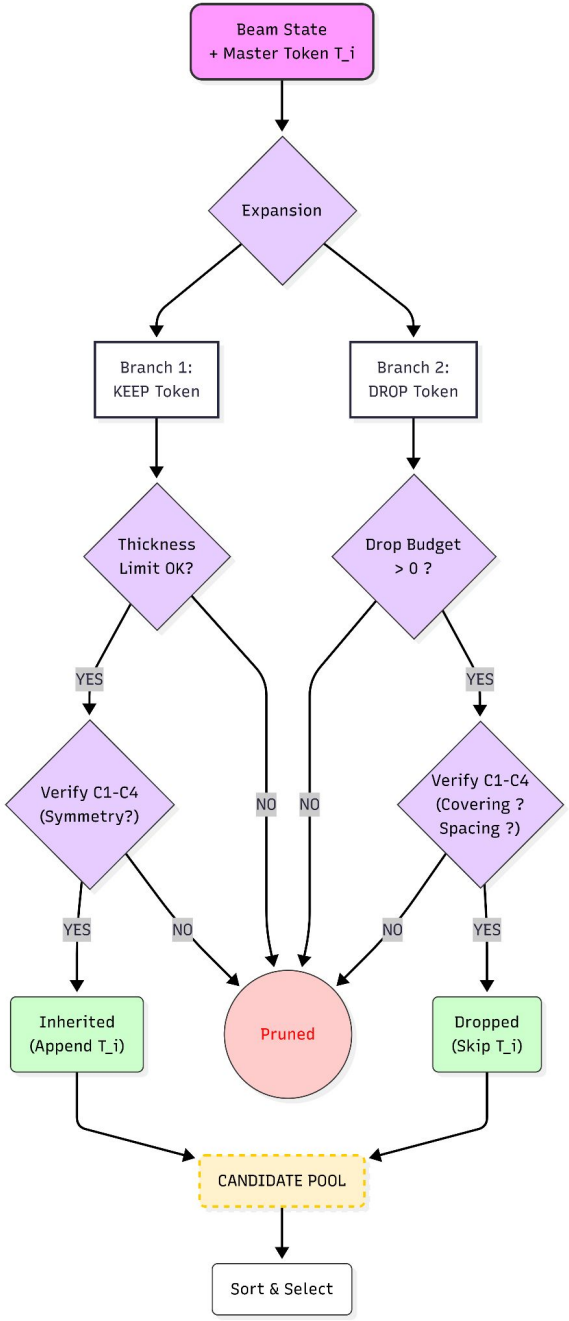
=> He had to sacrifice **design freedom** for **computational solvability**.

- With SeqGPT it is possible to try different PD's in a reasonable time.

AIRBUS

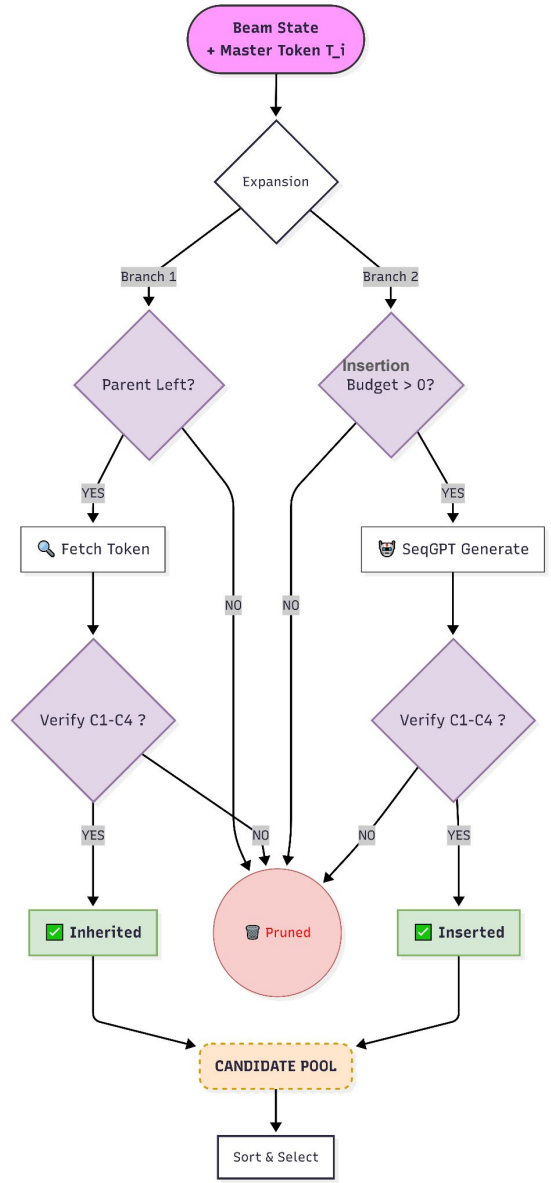
Constrained Decoding with seqGPT

Ply drop algorithm



Budget = | Size_son - Size_parent |

Ply Insertion algorithm

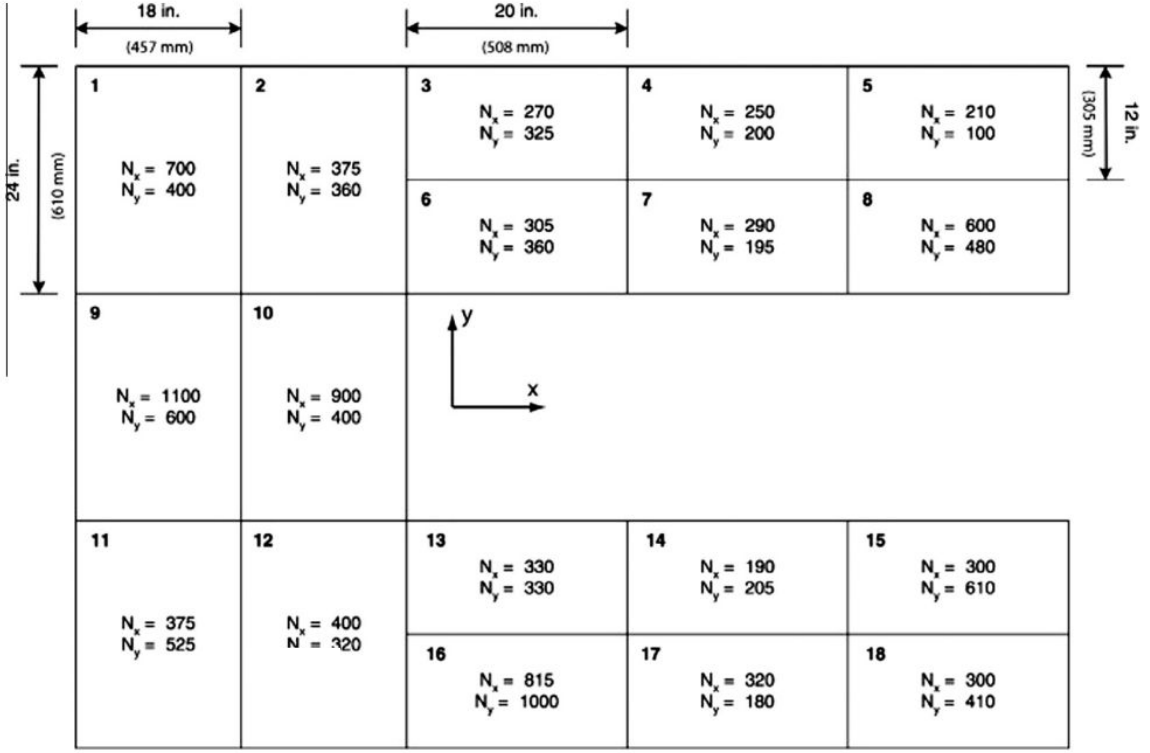


Airbus Amber

AIRBUS

The Benchmark & The Baseline Solution

Baseline problem



Eighteen-panel test problem [8,26], all loads in lbf/in (×175.1 for N/m).

Solution (Ply Drop SPD)

Airbus Amber

N_{min}		18		22		26		30		34		38					N_{max}
45	45	**45**	45	**45**	45	**45**	45	**45**	45	**45**	45	**45**	45	45	45	45	45
					45	**45**	45	**45**	45	**45**	45	**45**	45	45	45	45	45
		60	60	**60**	60	**60**	60	**60**	60	**60**	60	**60**	60	60	60	60	60
												30	30	30	30	30	30
45	45	**45**	45	**45**	45	**45**	45	**45**	45	**45**	45	**45**	45	45	45	45	45
										30	30	**30**	30	30	30	30	30
													30	30	30	30	30
						45	**45**	45	**45**	45	**45**	45	45	45	45	45	45
90	90	**90**	90	**90**	90	**90**	90	**90**	90	**90**	90	**90**	90	90	90	90	90
								-45	-45	**-45**	-45	**-45**	-45	-45	-45	-45	-45
											-30	**-30**	-30	-30	-30	-30	-30
						-45	-45	**-45**	-45	**-45**	-45	**-45**	-45	-45	-45	-45	-45
90	90	**90**	90	**90**	90	**90**	90	**90**	90	**90**	90	**90**	90	90	90	90	90
															90	90	90
			-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	-45	-45	-45	-45
-45	-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	-45	-45	-45	-45
																90	90
	-60	**-60**	-60	**-60**	-60	**-60**	-60	**-60**	-60	**-60**	-60	**-60**	-60	-60	-60	-60	-60
													-30	-30	-30	-30	
-45	-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	**-45**	-45	-45	-45	-45	-45
																	0
									-30	**-30**	-30	**-30**	-30	-30	-30	-30	-30
0	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	**0**	0	0	0	0	0
				45	45	**45**	45	**45**	45	**45**	45	**45**	45	45	45	45	45
	I	I	II	II	III	III	IV	IV	V	V	VI	VI	VII	VII	VIII	X	XI

Buckling Constraints

SST of Solution 1 in Table 6. Columns corresponding to the laminates of the panels are marked in bold letters. SB-cycles are numbered with Roman numerals.

This is done in the first step (FLP)

For our test base, we generate for each sequence of Irisarri solution, the buckling parameters and the number of plies.

AIRBUS

Annexes

Cette 12ème édition d'APIA se tiendra à Arras, dans le cadre de la Plate-Forme Intelligence Artificielle. Nous recherchons des contributions pour illustrer des applications concrètes de l'IA et partager des travaux dont l'objet d'étude concerne des problèmes opérationnels ou des données réelles, et nous sommes particulièrement intéressés par les propositions qui mettront en avant les sujets suivants, en lien avec l'IA Agentique :

- Orchestration de systèmes complexes, collaboratifs
- Recherche de résultat optimum
- Délégation de raisonnement
- Traitement du langage multimodal, nouvelles applications

Soumissions APIA

Format des soumissions:

Nous attendons des articles et des résumés mettant en avant des applications concrètes de l'intelligence artificielle. Nous apprécierons particulièrement que ceux-ci s'attachent à préciser le contexte opérationnel dans lequel ces applications se situent, à décrire en détail la solution proposée de sorte à en assurer la reproductibilité, et à justifier l'adéquation des choix scientifiques et techniques effectués aux objectifs applicatifs visés.

Les types de soumissions acceptés sont les suivants :

- **articles longs**, de 10 pages au maximum, références comprises, présentant des travaux originaux sur les thèmes de la conférence APIA ;
- **articles courts**, démos et positionnement, de 6 pages au maximum, références comprises. Ces articles peuvent présenter soit:
  1. des travaux préliminaires ou en cours ;
  2. des démonstrations d'applications ou de réalisations concrètes - inclure un lien vers une vidéo de présentation de l'outil ou du logiciel décrit;
  3. un état des lieux pour un domaine précis en lien avec les thèmes de la conférence et les défis applicatifs ou les verrous scientifiques importants dans ce domaine.
- **résumés**, de 2 pages au maximum, références comprises. Ces résumés peuvent présenter soit:
  1. des articles déjà publiés dans des conférences ou des revues internationales mais inédits en français - donner la référence de l'article publié dans le résumé ;
  2. des travaux préliminaires ou en cours, que vous souhaitez présenter en poster. Les résumés de posters acceptés feront l'objet d'une présentation flash pendant la conférence.

Dates APIA

- Envoi des résumés des soumissions : 15 février 2026 (10 lignes au maximum, dans le style PFIA)
- Soumission des articles : 1er mars 2026
- Notification aux auteurs : 15 avril 2026
- Réception des versions définitives : 8 mai 2026
- Dates de la conférence : du 29 juin au 1er juillet 2026

Input Strategy: We solve the inverse problem by injecting the targets at the start: `Input = [ Buckling_Targets | Ply_1, Ply_2, ]`

Attention

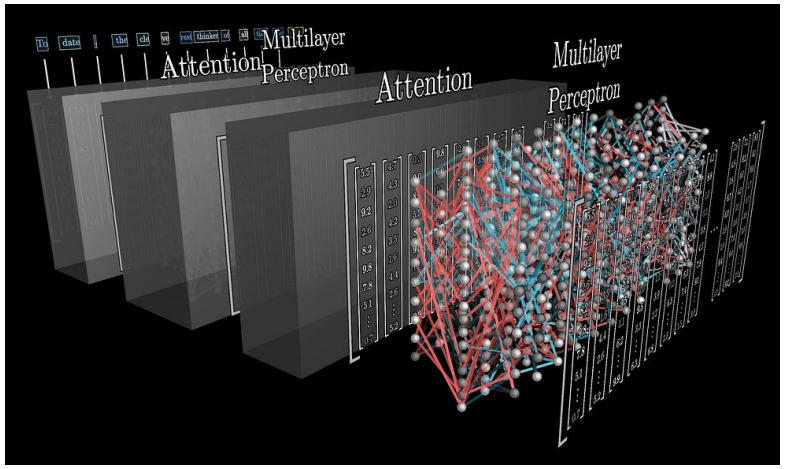
- **Role:** The attention mechanism creates a global context by calculating relationships between the current position and all previous tokens.
- **Pattern Recognition:** It detects prohibited patterns (like checking if placing a 90 next to 0 would violate a learned constraint) while simultaneously tracking the Mechanical Targets.
- **Contextual Alert:** It effectively highlights discrepancies: *"We are below the required % of 90° plies"*, “We are unbalanced” or *"We need more stiffness to meet Buckling Target 1."*
- **Inverse Mapping:** Instead of just looking at past plies, the model constantly "looks back" at the target parameters (P1-P4). It asks: *"Given these targets, what ply is needed?"*

MLP

- **Conflict Resolution:** The MLP processes this complex context where physical targets and manufacturing rules might conflict.
- **Implicit Logic:** It has learned to prioritize **valid designs**. For instance, even if a 0° ply is best for buckling, the MLP will output a 45° ply if the "max contiguous" or "no 0/90" rule requires it.
- **Final Output:** It refines the embedding to strictly favor the optimal token (ply) that satisfies both mechanical and manufacturing constraints

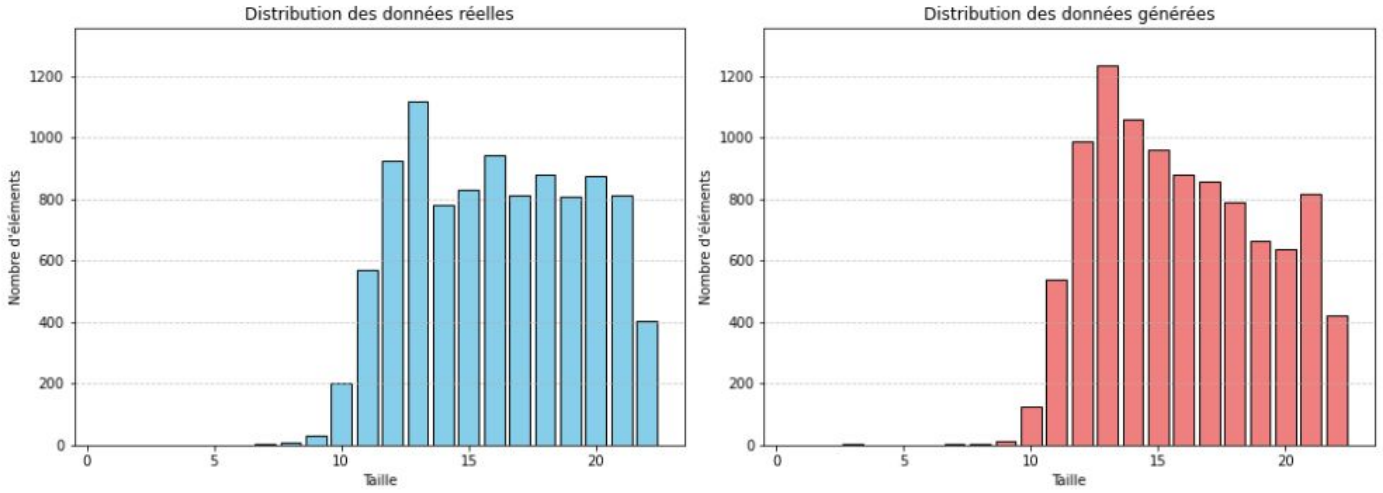
The model is composed of **multiple layers (or blocks), each containing an attention mechanism followed by an MLP**.

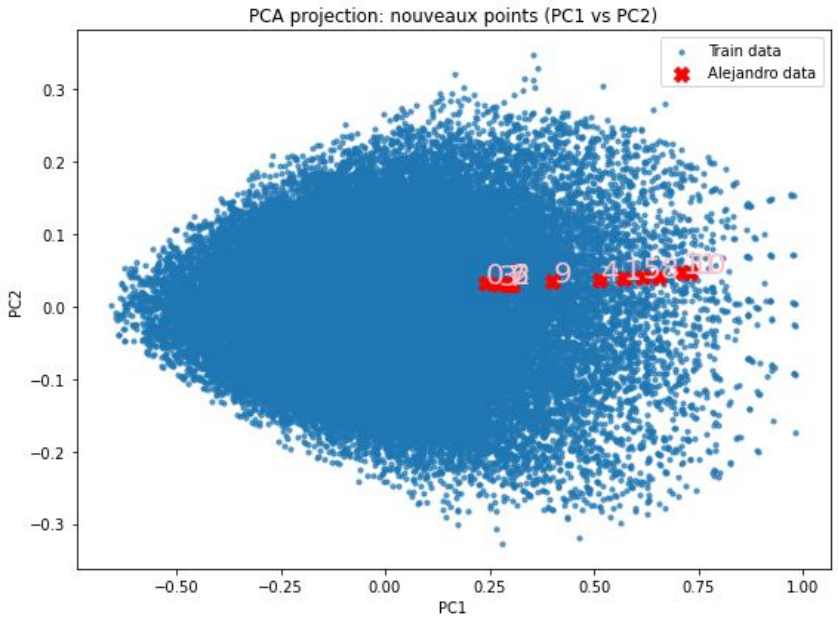
➔ *By **stacking several of these attention + MLP blocks**, the model can build increasingly sophisticated and nuanced representations, which leads to more accurate and coherent sequence generation.*



AIRBUS

Data distribution comparison





=> Sequences with length under 10 are out of the ODD.

AIRBUS

Annexes

More details on beam search constraints

Context & Framework - The Bi-Step Optimization Strategy

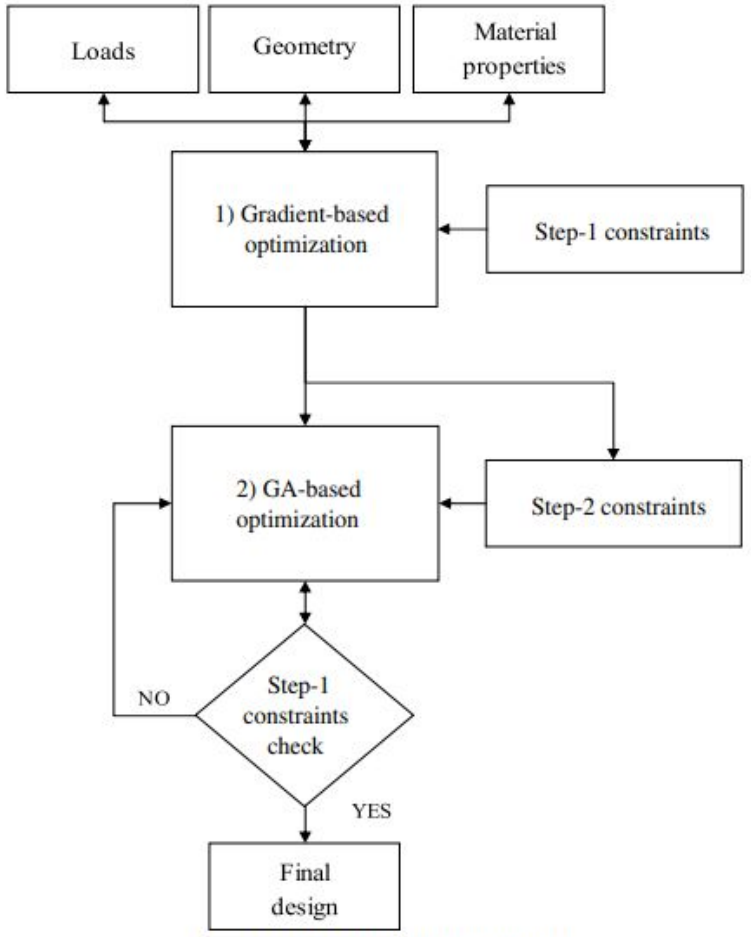
As described by **Herencia et al.[1]** and **Scardaoni and Montemurro [4]** , the optimization of complex composite structures is computationally expensive due to the nonconvex nature of the design space.

The Rules

Symmetry: Sequence mirrored around mid-plane.
Goal: $B_{ij} = 0$ (No thermal warping).
Balance: Equal number of $+45^\circ$ and -45° plies.
Goal: $A_{16}, A_{26} = 0$ (No extension-shear coupling).
Disorientation:
$\theta_{i+1} - \theta_i \neq 90^\circ$ involving 0°.
Forbidden: $0^\circ \rightarrow 90^\circ$ or $90^\circ \rightarrow 0^\circ$.
Allowed: $+45^\circ \rightarrow -45^\circ$.
Goal: Minimize interlaminar shear (Avoid delamination).
Contiguity: Max 4 identical plies consecutive.
Goal: Limit matrix cracking propagation.
Proportions: $10\% \leq$ Each orientation $\leq 66\%$.
Goal: Ensure damage tolerance & bearing strength.

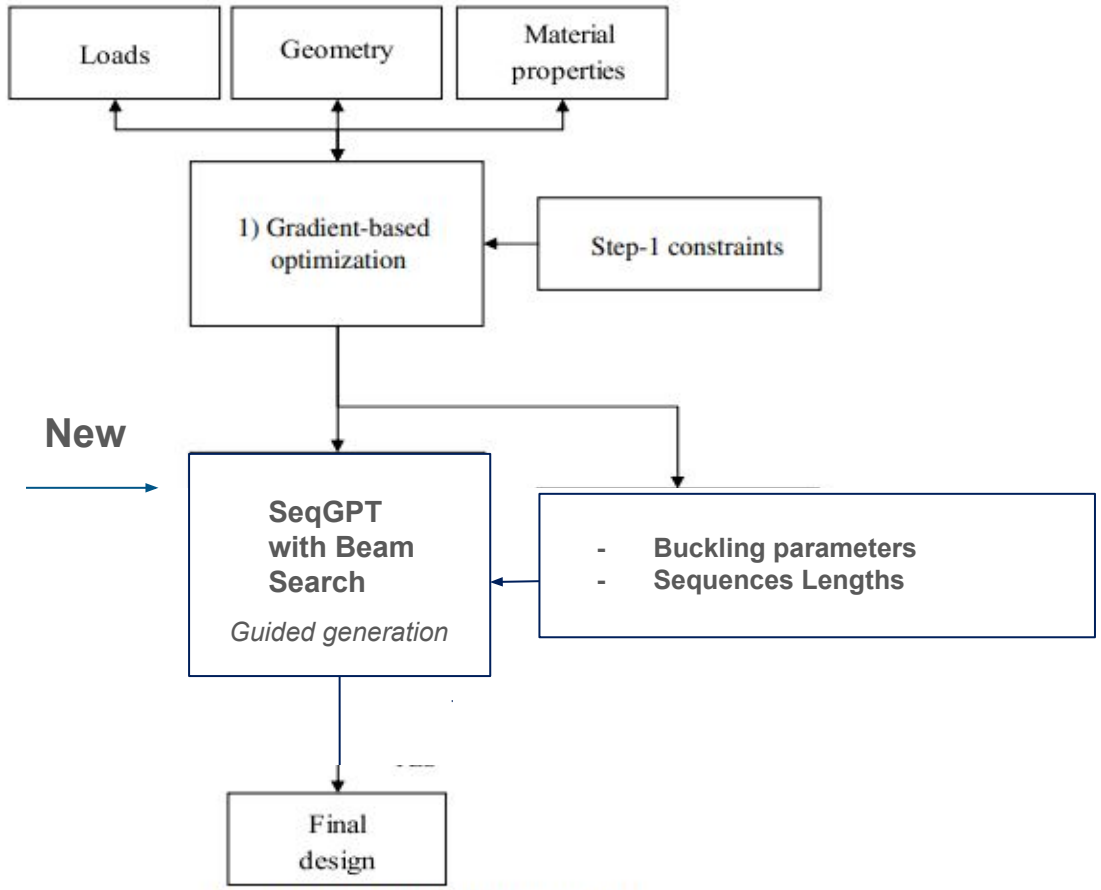
Goal : Accelerate Step 2 by replacing the slow Genetic Algorithm with a fast Neural Inference model (SeqGPT).

Context & Framework - The Bi-Step Optimization Strategy



GA: **Optimization flowchart.**

New



GenAI: **Optimization flowchart.**

→ **Adaptable**

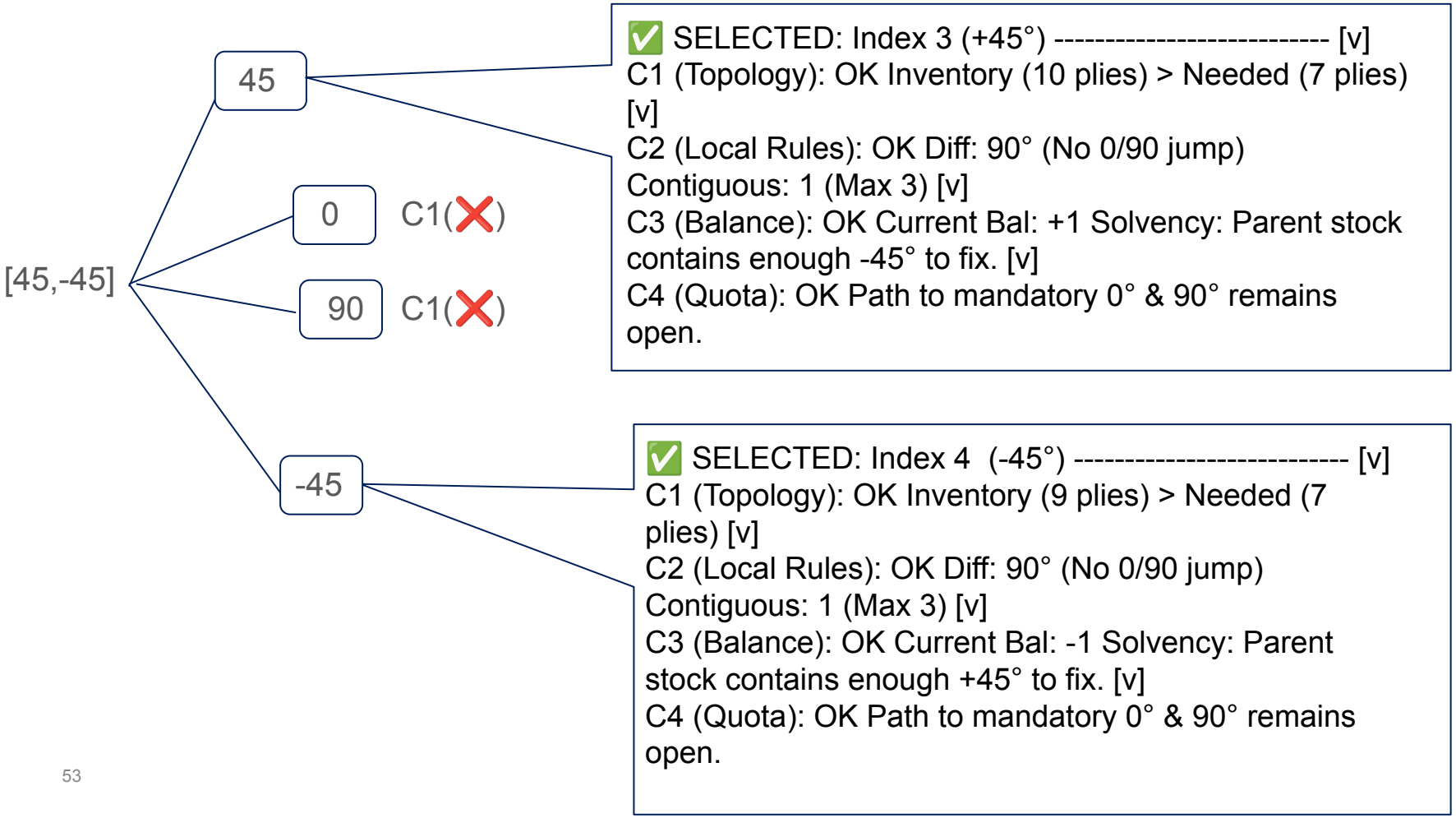
→ **Generalizable Training & Fast Inference**

→ **Compatible with manufacturing**

Constrained Decoding with seqGPT : Ply drop Example (Valid Paths)

Parent Sequence : [45, -45, 45, 45, -45, 45, -45, 0, -45, -45, 45, 90, -45] N = 13

Generated Sequence : [45, -45, 45, 45, -45, 0, -45, -45, 45, 90] N = 10



AIRBUS

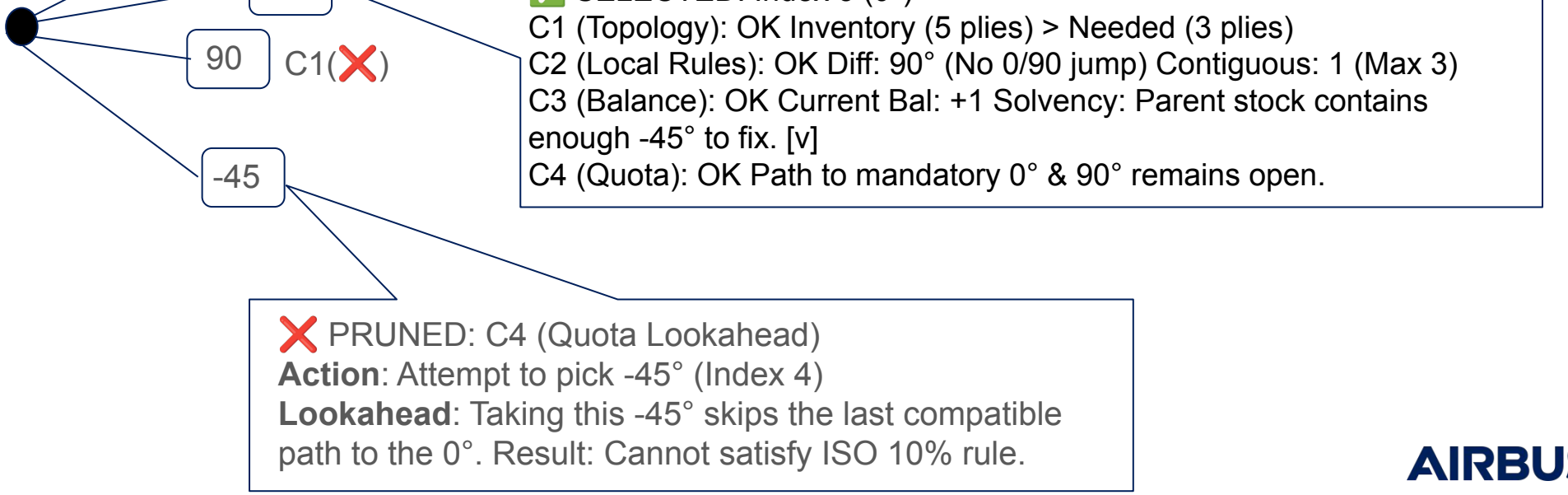
Constrained Decoding with seqGPT : Ply drop Example (Proportion rule)

Parent Sequence : [45, -45, 45, 45, -45, 45, -45, 0, -45, -45, 45, 90, -45] N = 13

Generated Sequence : [45, -45, 45, 45, -45, 0, -45, -45, 45, 90] N = 10

Let's consider one of the feasible path

[45, -45, 45, 45, -45, 45, -45]



AIRBUS

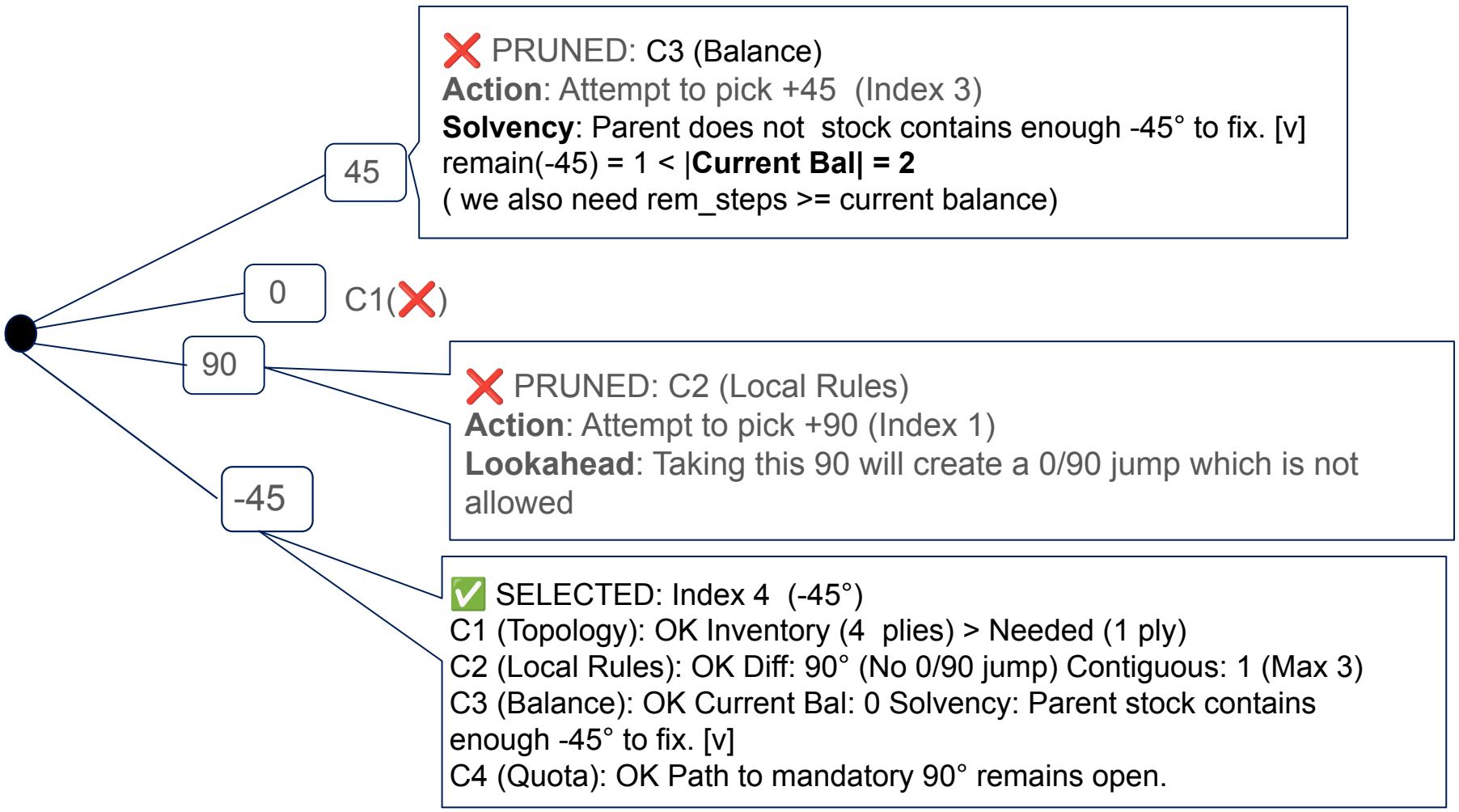
Constrained Decoding with seqGPT : Ply drop Example (Balance and local rules)

Parent Sequence : [45, -45, 45, 45, -45, 45, -45, 0, -45, -45, 45, 90, -45] N = 13

Generated Sequence : [45, -45, 45, 45, -45, 0, -45, -45, 45, 90] N = 10

Let's consider one of the feasible path

[45, -45, 45, 45, -45, 45, -45, 0]



Results with SST

Total Delta_plies = 7

	Target_N	Generated_N	Delta_Plies	IA_Margin	Ref_Margin	Diff_Margin	Sequence
9	38	38	0	4.653865	4.8	-0.146135	[45, -45, 45, 45, -45, 45, -45, 45, 0, -45, 45, 0, -45, -45, 90, 90, 45, -45, 0, 0, -45, 45, 90, 90, -45, -45, 0, 45, -45, 0, 45, -45, 45, -45, 45, 45, -45, 45]
10	35	35	0	4.839228	4.5	0.339228	[45, -45, 45, 45, -45, 45, -45, 45, 0, -45, 45, -45, -45, 90, 90, 45, -45, 0, -45, 45, 90, 90, -45, -45, 45, -45, 0, 45, -45, 45, -45, 45, 45, -45, 45]
1	34	35	1	27.507169	16.6	10.907169	[45, -45, 45, 45, -45, 45, -45, 45, 0, -45, 45, -45, -45, 90, 90, 45, -45, 0, -45, 45, 90, 90, -45, -45, 45, -45, 0, 45, -45, 45, -45, 45, 45, -45, 45]
16	32	32	0	8.915244	7.9	1.015244	[45, -45, 45, 45, -45, 45, -45, 0, -45, 45, -45, -45, 90, 90, 45, 0, 0, 45, 90, 90, -45, -45, 45, -45, 0, -45, 45, -45, 45, 45, -45, 45]
11	30	31	1	22.439088	10.3	12.139088	[45, -45, 45, 45, -45, 45, -45, 0, -45, 45, -45, -45, 90, 90, 45, 0, 45, 90, 90, -45, -45, 45, -45, 0, -45, 45, -45, 45, 45, -45, 45]
2	28	28	0	3.945989	3.7	0.245989	[45, -45, 45, 45, 45, -45, 0, -45, -45, -45, 90, 90, 45, 0, 0, 45, 90, 90, -45, -45, -45, 0, -45, 45, 45, 45, -45, 45]
12	28	28	0	3.497946	3.3	0.197946	[45, -45, 45, 45, 45, -45, 0, -45, -45, -45, 90, 90, 45, 0, 0, 45, 90, 90, -45, -45, -45, 0, -45, 45, 45, 45, -45, 45]
8	26	27	1	21.032482	9.6	11.432482	[45, -45, 45, 45, 45, -45, 0, -45, -45, -45, 90, 90, 45, 0, 45, 90, 90, -45, -45, -45, 0, -45, 45, 45, 45, -45, 45]
15	26	27	1	17.323966	6.2	11.123966	[45, -45, 45, 45, 45, -45, 0, -45, -45, -45, 90, 90, 45, 0, 45, 90, 90, -45, -45, -45, 0, -45, 45, 45, 45, -45, 45]
18	24	24	0	11.311473	20.5	-9.188527	[45, -45, 45, 45, 0, -45, -45, -45, 90, 90, 45, 0, 0, 45, 90, 90, -45, -45, -45, 0, 45, 45, -45, 45]
6	22	23	1	7.99739	2.9	5.09739	[45, -45, 45, 45, 0, -45, -45, -45, 90, 90, 45, 0, 45, 90, 90, -45, -45, -45, 0, 45, 45, -45, 45]
3	22	23	1	20.173316	14.5	5.673316	[45, -45, 45, 45, 0, -45, -45, -45, 90, 90, 45, 0, 45, 90, 90, -45, -45, -45, 0, 45, 45, -45, 45]
13	22	23	1	13.050744	7.8	5.250744	[45, -45, 45, 45, 0, -45, -45, -45, 90, 90, 45, 0, 45, 90, 90, -45, -45, -45, 0, 45, 45, -45, 45]
4	19	19	0	5.708248	7.7	-1.991752	[45, -45, 45, 45, -45, -45, -45, 90, 45, 0, 45, 90, -45, -45, -45, 45, 45, -45, 45]
7	19	19	0	2.389419	4.3	-1.910581	[45, -45, 45, 45, -45, -45, -45, 90, 45, 0, 45, 90, -45, -45, -45, 45, 45, -45, 45]
14	19	19	0	12.126525	14.3	-2.173475	[45, -45, 45, 45, -45, -45, -45, 90, 45, 0, 45, 90, -45, -45, -45, 45, 45, -45, 45]
17	19	19	0	3.846179	5.8	-1.953821	[45, -45, 45, 45, -45, -45, -45, 90, 45, 0, 45, 90, -45, -45, -45, 45, 45, -45, 45]
5	16	16	0	4.581615	6.5	-1.918385	[45, -45, 45, -45, -45, 90, 45, 0, 0, 45, 90, -45, -45, 45, -45, 45]

Training time of seqGPT :
5-10 min

Inference time to generate this table:
2 seconds

What do we do about the case where we have transition between a seq with N = 2k and a seq with N=2k+1

AIRBUS

Tokenization

Shape : (5,) (21,)

X_train[0,0] : [parameters] + [3, 4, 4, 3, 3, 1, 4, 2, 2, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]

ENCODER

nn.Linear(config.latent dim, config.n_embd),

nn.Embedding(config.vocab_size, config.n_embd)

Output shape : (1,48)

Output shape : (21,48)

Positional array : Gives information about the position of each token

[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]

nn.Embedding(config.block_size, config.n_embd)

Pos : Embedding shape : (22,48)

After **Embedding…**

X : Embedding shape : (22,48)

Addition

X = X + Pos -> because Each token needs to be "localized" in the sequence

Unembedding

X : Embedding shape : (22,48)

X = X + Pos -> because Each token needs to be "localized" in the sequence

better_X : Embedding shape : (22,48)

Attention + MLP Blocks
(n_heads attention block)
(n_layers MLP blocks)

self.lm_head = nn.Linear(config.n_embd, config.vocab_size, bias=False)

logits: Embedding shape : (22,5)

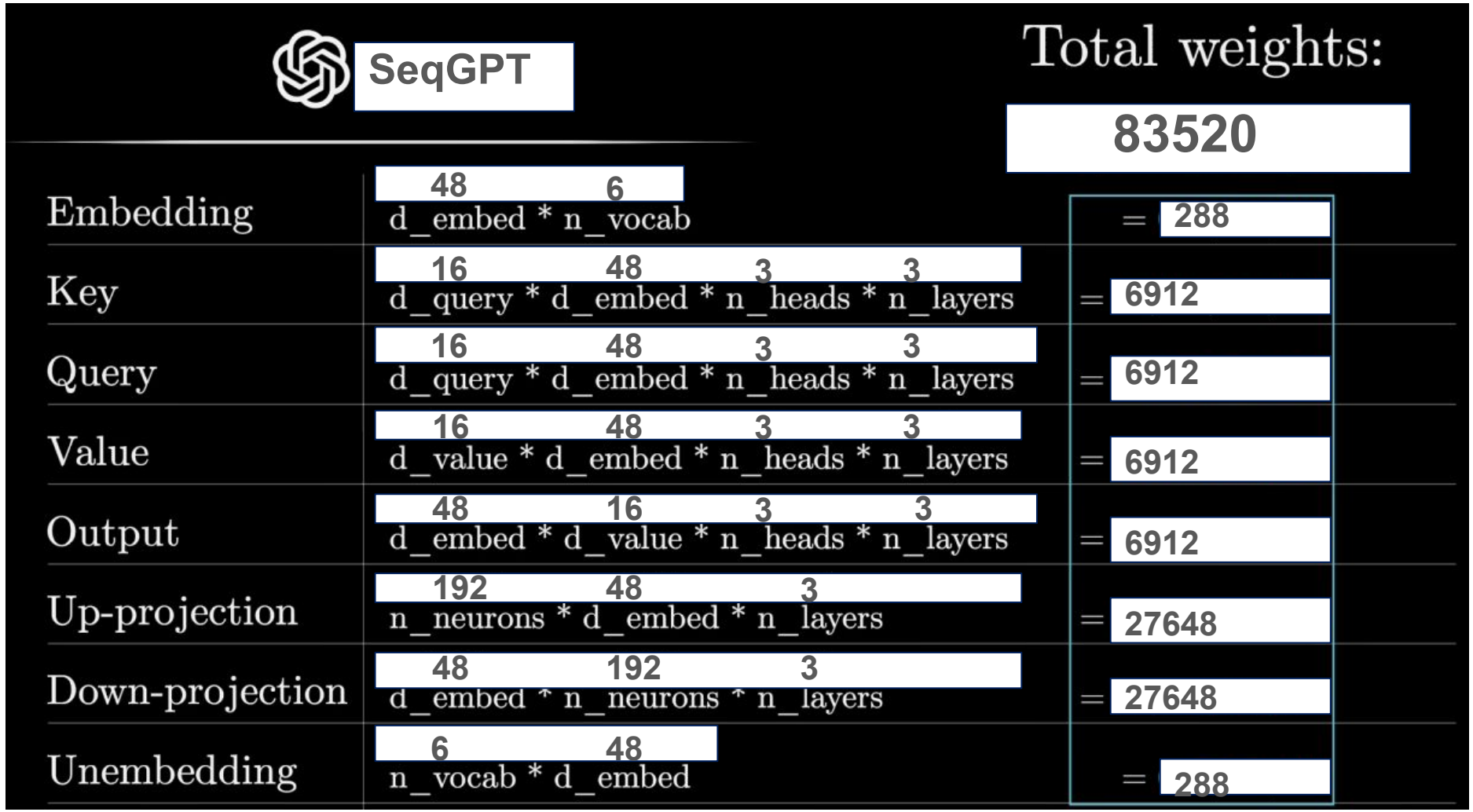
Last Layer To projects our Embedding into logits

Softmax

Probabilities Shape : (22,5)

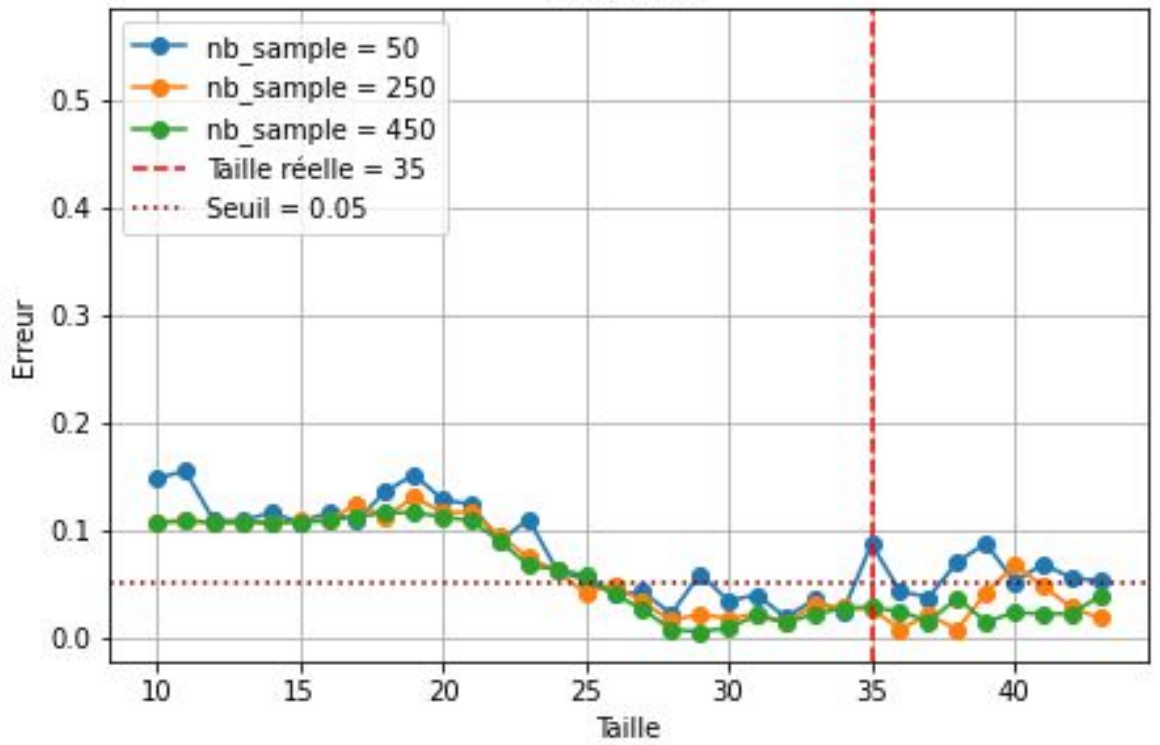
AIRBUS

How much parameters ?



Impact of Sample Size on Sequence Reconstruction Error

N = 50 is sufficient to keep the error below 10%.
Higher sample counts (N=250) stabilize the prediction, keeping the error consistently below the 5% threshold when the length is not too small.
Increasing the sample size more (N=450) offers negligible improvement.



Alejandro Example

Séquence 0 : [1.0003, 1.5773, 0.0000, 0.0000, 14.0000]

Sequences : [3, 4, 2, 1, 4, 1, 3] [3, 4, 2, 4, 1, 1, 3]
Parameters: [1.0321849 , 1.47879003, 0.03464391, 0.03251711],
[1.07005468e+00, 1.63965861e+00, -9.93e-17,-6.50e-17]
Errors: [0.0576331357046758, 0.08475433118895138]

Séquence 5 : [1.0934, 1.9508, 0.0000, 0.0000, 21.0000]

Sequences :
[3, 4, 4, 2, 3, 3, 4, 2, 1, 1, 1], [3, 4, 4, 2, 3, 3, 4, 2, 1, 2, 1], [3, 4, 4, 3, 2, 2, 4, 3, 1, 1, 1],
Parameters:[
[1.10212972, 1.93909147, 0.0000847900229, 0.0000698038421],
[1.1066388, 1.94219748, 0.0000852732905, 0.0000696307766],
[1.10777241, 1.91697368, -0.00603158, -0.00491507],
Errors :
[0.010879964824355515, 0.015802309971113322, 0.019056462069878863]

AIRBUS

A new method : seqGPT

Generate a **meaningful** Sentence that **answer** a prompt ⇔ Generate a **feasible** Sequence that **target** buckling (or lamination) parameters

minGPT

A PyTorch re-implementation of GPT, both training and inference. minGPT tries to be small, clean, interpretable and educational, as most of the currently available GPT model implementations can a bit sprawling. GPT is not a complicated model and this implementation is appropriately about 300 lines of code (see mingpt/model.py). All that's going on is that a sequence of indices feeds into a Transformer, and a probability distribution over the next index in the sequence comes out. The majority of the complexity is just being clever with batching (both across examples and over sequence length) for efficiency.

(*) *Adding lamination parameters as conditioning inputs reduces the training loss more quickly and to a lower value.*

But How to condition it ?

Idea :

- We are adding the lamination parameters-length in the beginning of each sequence : **(LP's = prompt)**
  - ➔ The model's attention mechanism uses both previous tokens and these parameters at every step.
  - ➔ This allows the model to generate sequences directly influenced by the lamination parameters.

It's a simple, effective* way to condition generation without extra architectural changes.

Training

6 tokens : 0 : Padding ; 5 : End of the sequence
1 : 0° 2 : 90° 3 : 45° 4 : -45°

Airbus Amber

Unconditional

X_train[0,0] : [3, 4, 4, 3, 3, 1, 4, 2, 2, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]

And ignore 0

y_train[0,0]: [4, 4, 3, 3, 1, 4, 2, 2, 1, 2, 2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Conditional

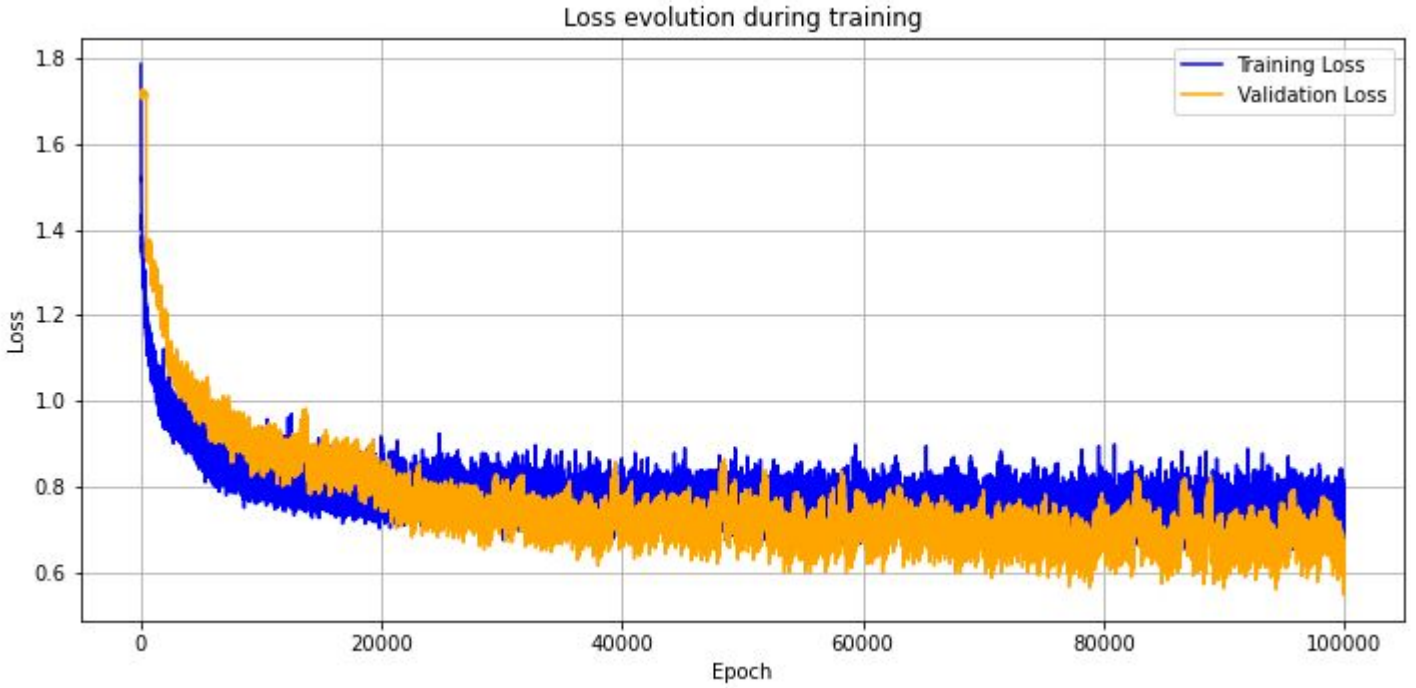
Attention

X_train[0,0] [parameters] +[3, 4, 4, 3, 3, 1, 4, 2, 2, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0]

And ignore 0

y_train[0,0] [padding] + [4, 4, 3, 3, 1, 4, 2, 2, 1, 2, 2, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0]

And ignore padding



Time = 50 min
(we could stop the training after 15 min)

The Loss: (same for both conditional and unconditional)

$$\mathcal{L} = -\sum_{i}\sum_{j\neq 0} y_{i,j}\log(p_{i,j}) \quad \text{où} \quad y_{i,j} = \begin{cases} 1 \\ 0 \end{cases}$$

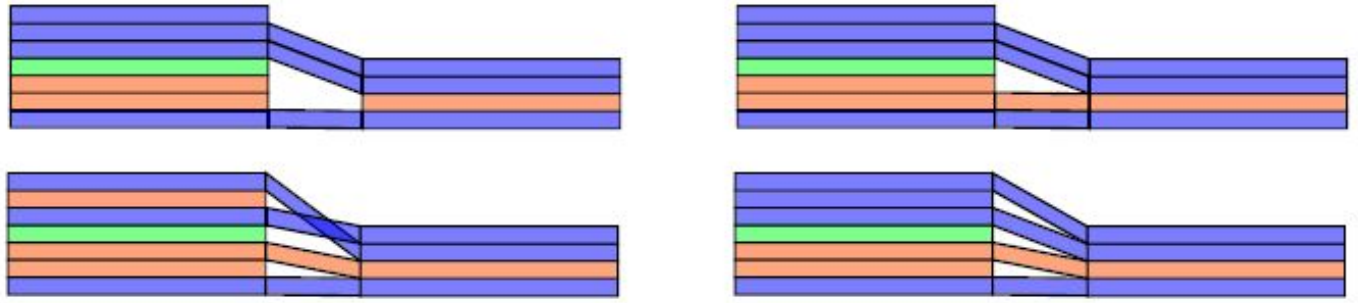
If j is the real class for the token i
If not

*With a current loss of 0.6 against a random baseline of ~1.6, the model demonstrates **strong predictive power**. While this result confirms effective learning, there is still margin for refinement*

AIRBUS

The connectivity challenge - Blending & SPD

Blending Constraint



(a) Two violations of blending map definition.

(b) Two possible blended arrangements between two given laminates

Non-blended laminates and different blending maps for the same pair of laminates (different colours mean different orientations). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

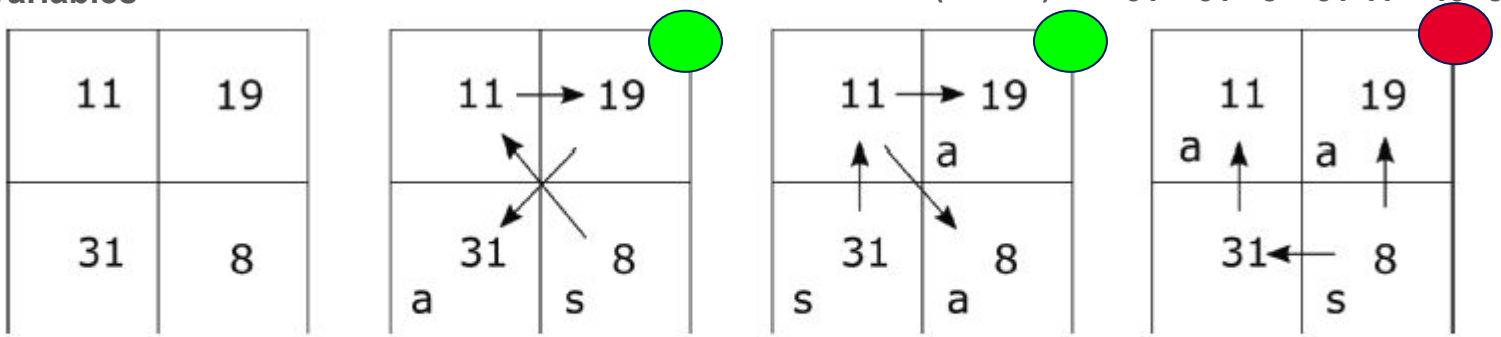
SPD (Search Propagation Direction) [4]

Number of indep variables

31 = 11 - 8 + 19-11 + 31 - 19

39 = 31 + (19 - 11)

31 = 31 - 8 + 31-11 + 19 -8



(a) A four-laminate structure

(b) Propagation 1

(c) Propagation 2

(d) Propagation 3 (counterexample)

Fig. ıetic experiment and three examples of propagation. Letters "s" and "a" stand for "start" and "arrival", respectively.

AIRBUS