



HAL
open science

Intégration de méthodes de Machine Learning pour le Post-Processing renforcé de l'Optimisation Topologique.

Baptiste Logghe, Sergio Octavio Esparza Martinez, Juan-Pedro Berro Ramirez, Thomas Guffroy, Christophe Bailly

► To cite this version:

Baptiste Logghe, Sergio Octavio Esparza Martinez, Juan-Pedro Berro Ramirez, Thomas Guffroy, Christophe Bailly. Intégration de méthodes de Machine Learning pour le Post-Processing renforcé de l'Optimisation Topologique.. CSMA 2024, CNRS, CSMA, ENS Paris-Saclay, Centrale Supélec, May 2024, Giens, France. hal-04822991

HAL Id: hal-04822991

<https://hal.science/hal-04822991v1>

Submitted on 6 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Intégration de méthodes de Machine Learning pour le Post-Processing renforcé de l'Optimisation Topologique.

B. Logghe^{1*}, S.O. Esparza Martinez¹, T. Guffroy¹,
C. Bailly¹, J.P. Berro Ramirez¹,

^{1*} Altair Engineering France, baptiste.logghe@gmail.com

¹ Altair Engineering France, {sesparza,tguffroy,bailly,jpbramirez}@altair.com

Résumé — Face aux défis de l'optimisation topologique, notre méthodologie repose sur l'utilisation d'un modèle de Geometric Deep Learning (GDL) pour prédire des réponses associées à des modifications du champ de densité. Cet article présente les premières pistes d'exploration nécessaires à la réalisation d'un potentiel post-processing interactif des résultats de topologie permettant aux concepteurs de prédire des performances sans relancer de calculs éléments finis de vérification.

Mots clés — Optimisation Topologique, Clustering, Post-Processing, SIMP, Cutting-Edge Geometric Deep Learning.

1 Introduction

Dans le domaine de l'ingénierie des structures mécaniques, la quête de l'optimalité persiste, alors que les ingénieurs naviguent parmi des contraintes multidimensionnelles, incluant des considérations économiques, de masse, et de méthodes de fabrication. L'une des approches prépondérantes pour aborder cette complexité réside dans l'application de la modélisation mathématique, avec une attention particulière portée à l'optimisation topologique. Cette méthodologie se révèle être un outil puissant, capable de dévoiler des voies d'efforts innovantes qui peuvent échapper à l'intuition des concepteurs, enrichissant ainsi le champ des possibles. Toutefois, l'un des défis persistants dans l'application de cette technologie réside souvent de la transition des structures optimales révélées par des méthodes numériques, vers des conceptions industrielles concrètes. Le post-processing (interprétation, reconstruction, etc.) des résultats de l'optimisation topologique s'avère être une tâche complexe et chronophage, exigeant un niveau d'expertise substantiel et une allocation de ressources significative.

C'est dans cette optique que notre recherche s'intéresse à une approche novatrice et prometteuse : l'exploitation de l'apprentissage automatique pour enrichir le processus de post-processing de l'optimisation topologique. Plus spécifiquement, notre étude se concentre sur la possibilité de prédire avec précision les modifications des performances structurelles induites par des variations topologiques (modifications du champ de densité) appliquées au résultat. Pour ce faire, un modèle d'apprentissage automatique calibré sur les itérations de l'optimisation topologique a été utilisé. L'objectif de ce document est de démontrer la faisabilité de cette démarche novatrice. Si cette approche se révèle concluante, elle pourrait ouvrir la voie à une optimisation topologique interactive, autorisant les concepteurs à intervenir de manière éclairée sur les designs issus de l'optimisation. Cette synergie entre la technologie d'optimisation topologique et l'expertise humaine représente une avancée significative dans la démarche de pré-design et d'optimisation des structures mécaniques, en permettant une adaptation plus précise des conceptions.

2 Revue bibliographique

2.1 Optimisation Topologique

L'optimisation topologique se consacre à la détermination de la distribution optimale de matière au sein d'une structure dans un cas de chargement donné. Cela revient à traiter un problème d'optimisation sous-contraintes en minimisant une fonction objective donnée sous des contraintes spécifiques. Afin de

faciliter cette optimisation et de permettre l'usage des méthodes de gradients, la méthode SIMP (Solid Isotropic Material with Penalization) est couramment employée. Elle repose sur la relaxation des variables binaires en variables continues, interprétant ainsi les variables de conception comme la densité de matière de chaque élément. Indépendamment développée par Bendsoe (1989) [1] et Zhou & Rozvany (1991) [8], cette approche se sert de techniques de pénalisation pour converger vers une solution discrète. En pratique, la densité de matière dans un élément influence le module de Young de celui-ci, permettant une modulation adaptative des propriétés mécaniques. Pour une exploration détaillée de la méthode SIMP, les travaux de Bendsoe & Sigmund (2003) [2] peuvent être utilisés comme document de référence.

Malgré les avancées considérables dans l'optimisation topologique, l'aspect souvent théorique et organique des résultats ainsi que la présence d'éléments semi-denses engendre des limitations dans leur réinterprétation. Le post-processing est donc une phase complexe et, bien que des outils géométriques sophistiqués aient vu le jour tel que Altair Inspire [9], la transition vers un design industriel final demeure une étape coûteuse et cruciale pour les concepteurs.

2.2 Application des méthodes de Machine Learning pour l'Optimisation Topologique

L'intégration récente des techniques de machine learning (ML) dans l'optimisation topologique pourrait être une piste permettant d'apporter des solutions à ce problème. Shin & al. (2021) [5] ont fourni une revue exhaustive des avancées dans ce domaine, mettant en évidence des orientations telles que l'accélération des itérations, la minimisation du temps de calcul grâce à une réduction adaptative de l'espace de conception, et notamment le post-processing. Cette dernière orientation est la pierre angulaire de notre étude. Les recherches de Wang et al. (2021) [7] illustrent l'utilisation d'un réseau neuronal convolutif (CNN) pour discrétiser les résultats de l'optimisation, supprimant les éléments semi-denses. Cette avancée représente un progrès significatif pour accroître la fiabilité des performances prédites par les solveurs, une préoccupation centrale pour les éditeurs de logiciels. En complément, Sosnovik & Oseledets (2019) [6] ont également déployé un modèle de ML sur des itérations de SIMP pour une discrétisation efficace via un réseau neuronal. Ces avancées soulignent le potentiel croissant du ML pour affiner et améliorer les complexités inhérentes à l'optimisation topologique et les défis qui lui sont associés.

3 Méthodologie

L'objectif principal de la méthode est de prédire les performances associées aux modifications de design à partir d'un modèle d'apprentissage, sans nécessiter une refonte complète et une re-simulation du modèle éléments finis pour chaque modification de design que l'on souhaite tester. Approche traditionnellement utilisée mais qui est extrêmement chronophage et peu flexible.

L'approche (fig. 1) repose sur l'utilisation d'un réseau neuronal convolutif pour prédire les performances liées aux modifications de la topologie (design). Pour former ce modèle, nous avons généré des jeux de données à partir de simulations d'optimisation topologique lancées avec le solveur Optistruct [10], en utilisant la méthode SIMP. Chaque itération de chaque simulation a fourni des données de densité pour chaque élément du maillage et de déplacement associé à chaque nœud. L'intérêt de l'approche est qu'elle peut fournir un grand jeu de données en récupérant des informations de chaque itération de chaque optimisation topologique. Cependant, cette génération automatique cause certains problèmes comme l'apparition de certains biais et répétitions (voir Section 3.2). Pour éviter ces derniers dans les données, nous avons réduit les datasets en utilisant un algorithme K-Means. Une fois les datasets réduits, nous les avons utilisés pour entraîner un modèle de deep learning dans PhysicsAI v2023. En fournissant les densités en tant qu'entrées et les champs de contour de déplacement correspondants en tant que sorties à notre réseau neuronal convolutif on cherche à prédire des performances associées à un design.

3.1 Génération de jeux de données issus de plans d'expérience à partir de la méthode SIMP sous Optistruct

Pour entraîner notre modèle d'apprentissage automatique, nous avons généré trois jeux de données distincts à partir de simulations conduites avec Optistruct, en appliquant une logique de plan d'expé-

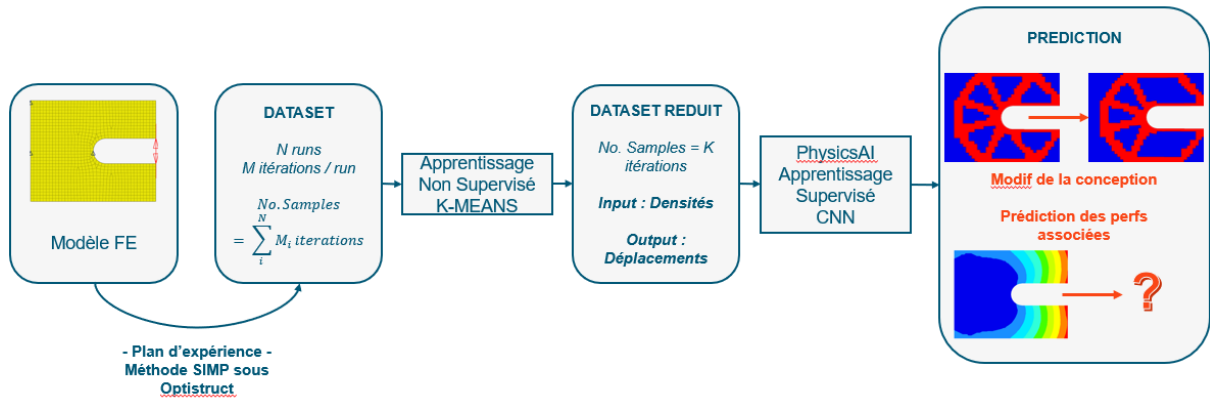


FIGURE 1 – Schéma de la méthodologie appliquée

rience. Ces simulations ont exploité la méthode SIMP pour l’optimisation topologique. Chaque itération produite par cette méthode constitue un échantillon de notre jeu de données, avec chaque échantillon contenant des informations de densité pour chaque élément et de déplacement pour chaque nœud (fig. 2).

- **DATASET 1** : Issu de $N = 27$ runs d’optimisation basés sur trois paramètres distincts à trois valeurs, ce jeu de données a engendré 877 échantillons (877 itérations contenant chacune une information de densité pour chacun des 1120 éléments et la norme du déplacement pour chacun des 1207 noeuds).
- **DATASET 2** : Il reprend les 27 runs d’optimisation du DATASET 1, mais se concentre uniquement sur les 11 dernières itérations de chaque run, qui sont généralement plus stables, plus discrètes et plus proches des designs finaux sur lesquelles on souhaite réaliser la prédiction. Ce jeu de données compte ainsi 297 échantillons, il est plus spécifique au problème (les champs de densité contenus dans ces échantillons sont proches des modèles de prédictions sur lesquels l’utilisateur souhaitera prédire les réponses associées).
- **DATASET 3** : Dans l’optique de maximiser l’exploration de l’espace de conception, nous avons enrichi le DATASET 1 avec 54 runs d’optimisation supplémentaires, suivant un DOE de $3 \times 3 \times 3$. Cela nous donne un total de $N = 81$ runs d’optimisation et 2161 échantillons pour ce jeu de données. Il est plus large, plus diversifié et plus exploratoire que les deux autres datasets (puisque’il est issu de formulations de calculs plus diverses réalisées, dans le but de mesurer l’impact de l’exploration sur la qualité des prédictions finales).

On a donc : $\text{DATASET 2} \subset \text{DATASET 1} \subset \text{DATASET 3}$

3.2 Sous-échantillonnage des datasets par clustering K-Means

En raison de l’algorithme utilisé (SIMP sous Optistruct) pour générer les données d’entrées du modèle de machine learning, beaucoup d’itérations (samples du dataset) partagent le même champ de densités et de déplacements. La convergence intrinsèque de l’algorithme d’optimisation génère à la fois des itérations de calculs similaires pour deux runs différents, mais aussi pour deux itérations consécutives d’un même run. En plus d’augmenter le temps d’apprentissage du modèle de Deep Learning, ces itérations similaires créent des biais, affectant la précision finale du modèle.

Afin d’améliorer la qualité des datasets, une démarche de sous-échantillonnage est mise en place. Cette dernière a pour objectif de créer des datasets représentatifs, couvrant au maximum les différents cas de nos ensembles de données générés automatiquement par

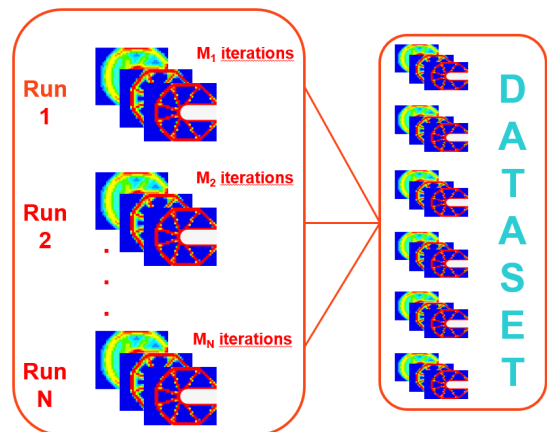


FIGURE 2 – Exemple de construction de dataset

l'optimisation. A l'aide de cette démarche, il est possible d'obtenir un meilleur contrôle et regard sur la qualité finale des jeux de données utilisés pour entraîner notre modèle de Deep Learning.

Sur chacun des datasets, on applique un algorithme non-supervisé de type K-Means. Avec cet algorithme on considère un vecteur densité lié à chaque échantillon, puis on crée k clusters définis en fonction de la distance euclidienne des vecteurs densités. Finalement, de chaque cluster on conserve un échantillon aléatoirement créant ainsi un sous-dataset de taille k . La démarche est montrée sur la figure 3.

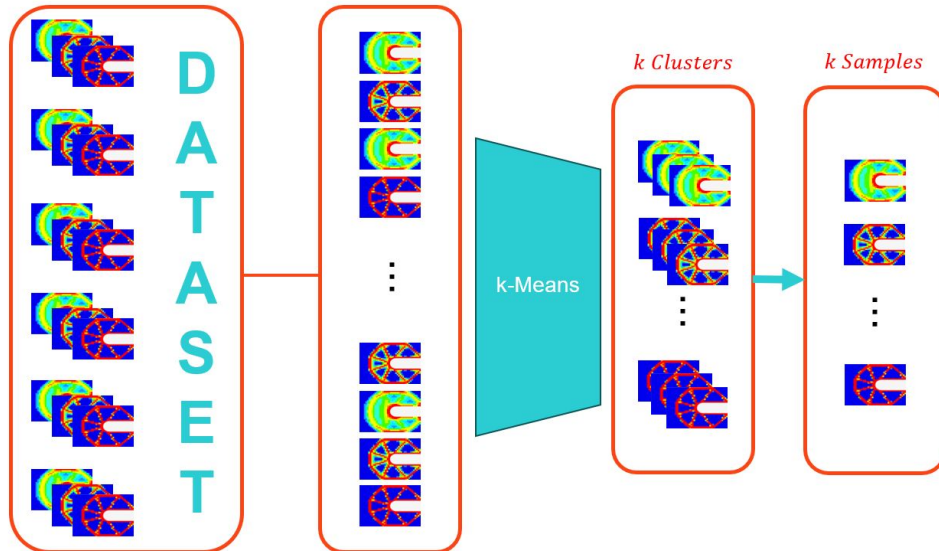


FIGURE 3 – Méthode proposée de sous-échantillonnage. A gauche un Dataset quelconque. A droite le sous-dataset correspondant

Par clustering, on constitue trois sous-datasets de chacun des trois datasets précédents :

- **CLUSTER 1** \subset **DATASET 1** : Sous-dataset avec $k = 150$, correspondant à 17% des samples
- **CLUSTER 2** \subset **DATASET 2** : Sous-dataset avec $k = 100$, correspondant à 33% des samples
- **CLUSTER 3** \subset **DATASET 3** : Sous-dataset avec $k = 250$, correspondant à 12% des samples

Le choix de k pour la réduction des modèles s'est basée sur l'expérience et l'observation pour déterminer les valeurs offrant des bons ratios temps de calcul/performance.

4 Résultats

Afin d'évaluer la fiabilité de nos modèles, nous les avons confrontés aux ensembles d'entraînement et de test. L'ensemble d'entraînement permet d'assurer la robustesse du modèle, évitant l'overfitting. Les modèles de réseaux neuronaux profonds sont très complexes et sont formés pour détecter des patterns dans les données. Toutefois, si les ensembles de données possèdent du bruit ou sont trop petits, le modèle sera capable de détecter des patterns à travers le bruit qui ne sont généralement pas extrapolable et réduiront la robustesse, la stabilité et la performance prédictive du modèle [4]. L'ensemble de test, composé de topologies modifiées proches de l'itération finale, simule des prédictions sur des designs modifiés, offrant une perspective directe de leur application réelle en post-processing.

On observe sur la figure 4 une prédiction intéressante qui conserve l'information du champs de déplacement (contour global, symétrie, etc.) et qui commet des erreurs relatives raisonnables aux alentours de 10 à 15%. Ces prédictions offrent des perspectives intéressantes puisqu'elles ne nécessitent pas de re-simuler le nouveau design que l'on souhaite tester à partir de la forme optimale. Dans une phase de pré-design, des erreurs relatives proches de 15% sont tolérables pour évaluer rapidement la performance d'un design.

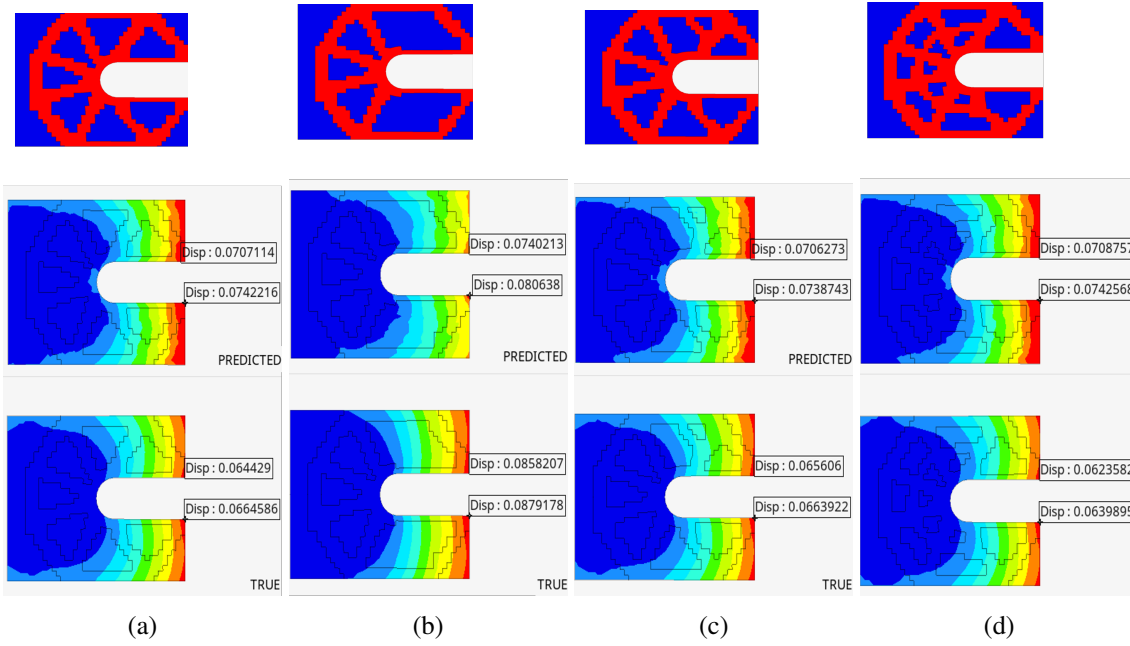


FIGURE 4 – Exemples de prédictions obtenus avec le **CLUSTER 1** : (a) Design final de l’optimisation topologique ; (b) Retrait de deux bras sur le design final ; (c) Modification de la connectivité d’un bras sur le design final ; (d) Ajout d’un renfort sur le design final

Nous avons utilisé deux indicateurs de performance pour évaluer et comparer entre eux les différents jeux de données afin d’en tirer les conclusions nécessaires à la génération de ces derniers :

MeanL2Norm(ML2) :

$$NormeL2 = \frac{\|\mathbf{pred} - \mathbf{true}\|_2}{\|\mathbf{true}\|_2} \times 100 \quad (1)$$

où **pred** et **true** sont des vecteurs de taille 1207, représentant respectivement les prédictions et les valeurs réelles de la magnitude de déplacement pour chaque nœud du maillage ; $\|\cdot\|_2$ est la norme euclidienne. Ainsi, cette norme mesure la norme euclidienne relative (en %) entre un modèle de prédiction et son champ de déplacement réel.

$$ML2 = \frac{1}{N} \sum_{i=1}^N L2_i \quad (2)$$

où N est le nombre total de modèles de prédiction, et $L2_i$ est la valeur de $L2$ obtenue pour le i ème modèle de prédiction.

MeanAverageError(MAE) :

$$AE_i = \frac{|\mathbf{pred}(i) - \mathbf{true}(i)|}{|\mathbf{true}(i)|} \quad \forall i = 1, \dots, 1207 \quad (3)$$

AE est l’erreur relative commise sur le déplacement chaque nœud du maillage.

$$MAE = \frac{1}{N} \sum_{i=1}^N AE_i \quad (4)$$

où N est le nombre total de modèles de prédiction, et AE_i est la valeur de AE obtenue pour le i ème modèle de prédiction.

On s’aperçoit que les deux indicateurs donnent les mêmes tendances (voir Tab1) :

- Les clusters (datasets réduits) fournissent des meilleures performances de prédiction que les datasets d’origine, ils ont des erreurs relatives inférieures.

TABLE 1 – Tableau de performance des différents modèles sur le modèle de prédiction

	D1	C1	D2	C2	D3	C3
ML2 (%)	43.63	21.25	34.29	31.36	46.42	16.71
MAE (e-3)	5.93	3.51	4.05	3.87	6.47	3.02

- On retrouve que le cluster 3 est plus performant que le cluster 1 lui-même plus performant que le cluster 2. Ainsi conformément à l'établissement de nos jeux de données, on retrouve que plus le dataset est varié, plus l'exploration de l'espace de conception est maximale et meilleure est la performance en matière de prédiction.

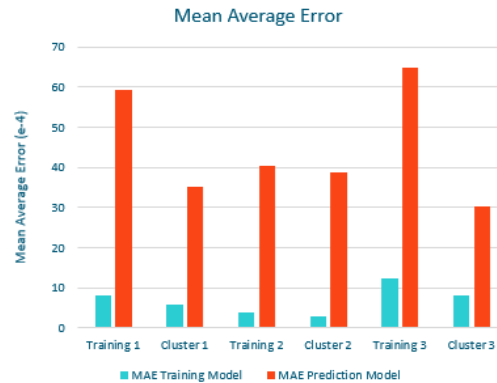


FIGURE 5 – Diagramme en barres des MAE pour chaque dataset - cluster

La Figure 5 illustre l'influence de l'overfitting sur notre modèle. Une meilleure adaptation aux données d'entraînement (MAE faible sur le Training Model) entraîne de meilleures prédictions (MAE faible sur le Prediction Model). Au vu de la complexité du modèle, la limite d'overfitting n'est pas encore atteinte, suggérant un potentiel d'amélioration par un entraînement plus poussé quelle que soit le dataset.

Pour évaluer l'effet du clustering, nous avons calculé un ratio :

$$\text{RATIO_TC} = \frac{(MAE_DATASET_i - MAE_CLUSTER_i)}{MAE_DATASET_i} \times 100$$

Un ratio positif montre que le cluster surpasse le dataset d'après l'équation précédente. Ainsi, l'analyse de la figure 6 révèle que le clustering est d'autant plus bénéfique que le dataset est grand (les ratios associés aux datasets 1 et 3 sont meilleurs que ceux du dataset 2). Ainsi, l'approche optimale serait de générer le dataset le plus vaste et diversifié possible, puis d'optimiser la valeur de K lors du clustering, pour maximiser les performances du modèle, tout en trouvant un équilibre entre temps de calcul et optimisation. Enfin d'un point de vue pré-design on peut affirmer qu'un MAE d'un ordre de grandeur autour des 1E-3 (autour des 10% d'erreur relative sur l'ouverture du clip) est convaincant pour prédire des modifications de performances associées à une modification de design, car on est à la recherche de tendances de développement pour le design, et non d'une validation de la performance du design. Cette approche par la prédiction permettrait d'éviter une phase longue de réinterprétation/reconstruction géométrique et d'un nouveau modèle FEM associé pour simuler ces modifications. Ce qui permettrait de significativement améliorer les performances de post-processing dans la phase de pré-design.

5 Limitations et perspectives

Ainsi, nous avons démontré la possibilité d'utiliser l'approche pour prédire des modifications de performance associée à des modifications de design à partir des itérations de la méthode SIMP. Cette approche présente néanmoins certaines limitations qui sont résumées dans ce tableau :

En particulier, l'un des points bloquants qu'il sera nécessaire de lever dans le futur est de réussir à améliorer la prédiction sur des itérations très discrètes (Présence de Matière : Densité = 1 | Absence de

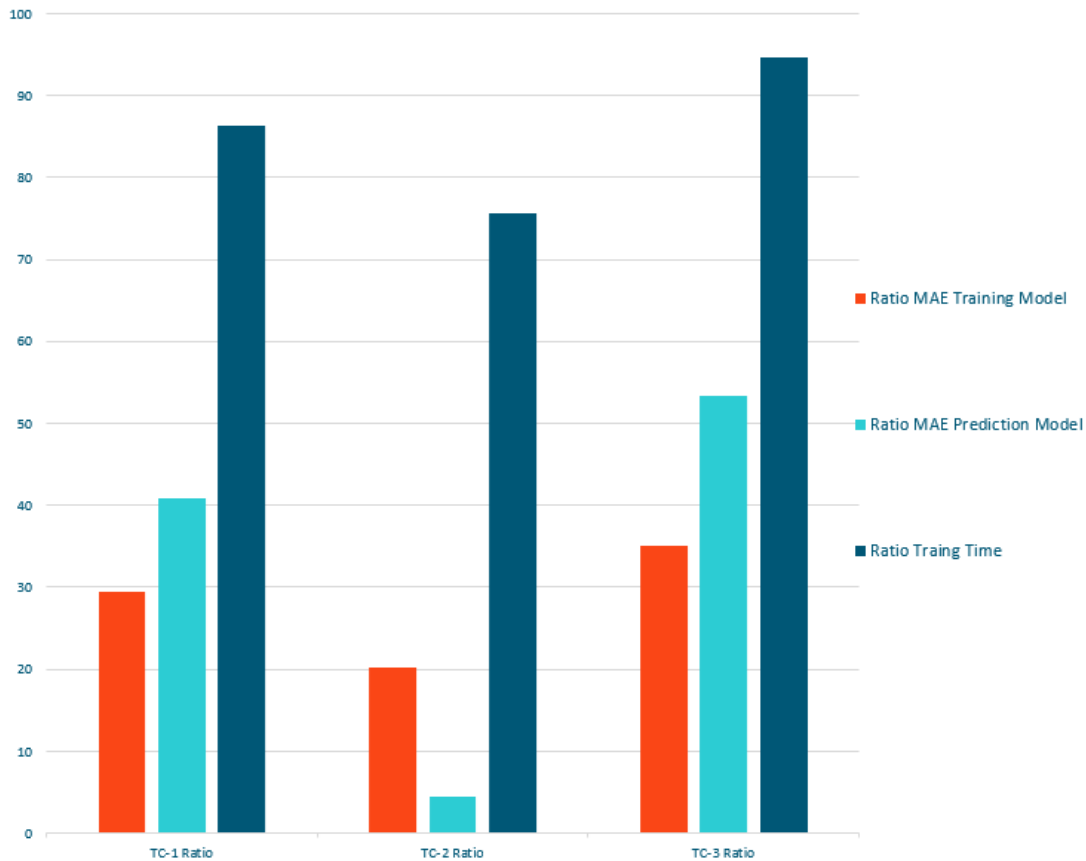


FIGURE 6 – Diagramme en barres des ratios de performance

TABLE 2 – Limitations et perspectives

Limitations	Travaux futurs
Clustering	Formulation du problème d'optimisation de la valeur de K ; Parallélisation avec clustering sur les outputs (déplacements) pour homogénéiser les datasets
Prédiction sur design discret	Implémentation d'un filtre pour améliorer la prédiction sur des résultats discrets

matière : Densité = 0.001). En effet, ces samples ne sont que très peu présentes dans les datasets fournis à partir de l'approche SIMP (itérations semi-denses) et nécessitent donc d'être traitées. C'est donc en ce sens que nous avons testé différents filtres pour améliorer la prédiction de nos modèles ; l'objectif est clair : réduire l'écart entre itérations semi-denses et discrètes.

On a donc testé un premier filtre, le plus simple possible, où les éléments représentant le design de la pièce finale (densité = 1) sont soumis à des densités plus faibles (0.8 / 0.7) et le reste des éléments représentant le vide (densité=0.001) sont soumis à des densités supérieures (0.2/0.3). Ce filtre fournit de bonnes performances en matière de prédiction par le modèle d'apprentissage, mais nous éloigne d'une représentation fiable du design final d'un point de vue industriel, car la présence d'éléments semi-denses renforce structurellement le design.

Le deuxième filtre testé par une approche des "voisins" consiste à assigner à chaque élément la moyenne des densités de ses éléments voisins. Cela permet de réduire la discrétisation des designs à prédire, et permet d'améliorer la prédiction sans dégrader la fiabilité du design final d'un point de vue industriel.

Un travail d'optimisation sur les filtres sera à réaliser pour garantir une prédiction optimale des designs discrets.

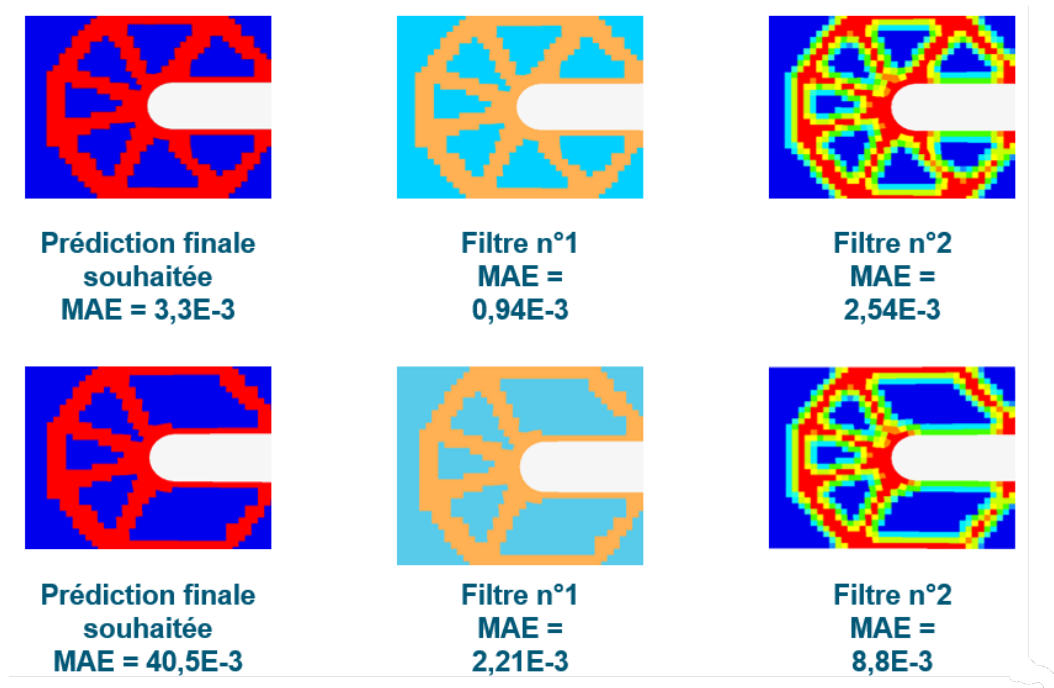


FIGURE 7 – Performances des filtres testés

6 Conclusion

Dans cette étude, nous avons exploré l'application de l'apprentissage automatique à l'optimisation topologique, en se concentrant sur la prédiction des performances structurelles suite à des modifications géométriques. L'approche adoptée, basée sur un réseau neuronal convolutif informé par la physique, a montré un potentiel prometteur, malgré certaines limitations inhérentes aux données d'entraînement semi-denses. Des filtres adaptatifs ont été mis en œuvre pour améliorer la précision sur les designs discrets. Toutefois, des améliorations et optimisations restent à réaliser pour garantir la fiabilité des prédictions. L'approche ébauchée pave la voie vers des améliorations de la phase de post-processing de l'optimisation topologique à la recherche d'un outil modulaire et interactif permettant la convergence de l'expertise humaine et de l'optimisation automatisée dans la phase de pré-design vers une conception éclairée.

Références

- [1] Bendsøe, M. P. (1989). *Optimal shape design as a material distribution problem*. *Structural and multidisciplinary optimization*, 1(4), 193–202.
- [2] Bendsøe, M. P. & Sigmund, O. (2003), *Topology optimization-theory, methods and applications*.
- [3] Chen, H., Joglekar, A., & Kara, L. B. (2023). *Topology Optimization using Neural Networks with Conditioning Field Initialization for Improved Efficiency*. <http://arxiv.org/abs/2305.10460>
- [4] Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn TensorFlow*, 26-32. O'REILLY.
- [5] Shin, S., Shin, D., & Kang, N. (2021). *Topology Optimization via Machine Learning and Deep Learning : A Review*.
- [6] Sosnovik, I., & Oseledets, I. (2019). *Neural networks for topology optimization*. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 34(4), 215–223. <https://doi.org/10.1515/rnam-2019-0018>
- [7] Wang, C., Yao, S., Wang, Z., & Hu, J. (2021). *Deep super-resolution neural network for structural topology optimization.*, *Engineering Optimization*, 53(12), 2108–2121, <https://doi.org/10.1080/0305215X.2020.1846031>
- [8] Zhou, M. & Rozvany, G. (1991). *The coc algorithm, part ii : topological, geometrical and generalized shape optimization*. *Computer Methods in Applied Mechanics and Engineering*, 89(1-3), 309–336.
- [9] Altair Inspire 2022.3 User Guide, https://2022.help.altair.com/2022/inspire/en_us/index.htm
- [10] Altair Optistruct 2022 User Guide, <https://2022.help.altair.com/2022/hwsolvers/os/index.htm>