

# Un réseau de neurones augmenté pour simuler l'évolution de la distribution de matière noire

Jean PROST   Pierre-Antoine THOUVENIN   Jenny SORCE   Pierre CHAINAIS  
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

**Résumé** – Pour étudier l'Univers, il est essentiel de pouvoir simuler rapidement l'évolution de la distribution de matière noire. Ce travail implémente un réseau de neurones 3D-UNet pour prédire efficacement le résultat d'une coûteuse simulation numérique à  $N$  corps de l'évolution de cette distribution dans un volume donné. Une nouvelle stratégie pour prédire conjointement position et vitesse des particules de la simulation est proposée, avec un terme de couplage dans la fonction d'entraînement pour assurer la cohérence entre position et vitesse. Nos résultats atteignent une précision état de l'art avec un gain significatif en temps d'exécution.

**Abstract** – To study the Universe, it is essential to rapidly simulate the evolution of the dark matter distribution. This work implements a 3D-UNet neural network to efficiently predict the output of a costly  $N$ -body numerical simulation of this distribution evolution. A novel training strategy is introduced to jointly predict the position and velocity of the simulation particles, along with a coupling term to ensure their consistency. Our results achieve state-of-the-art accuracy with a significant speedup.

## 1 Introduction

Selon le modèle cosmologique standard  $\Lambda$ CDM, la matière dans l'Univers était initialement répartie comme un champ aléatoire Gaussien, suivant le spectre de puissance mesuré dans le fond diffus cosmologique. Sous l'effet de la gravité, ces fluctuations ont mené à la formation des structures à grandes échelles observables aujourd'hui [14]. Les simulations numériques de l'évolution de la matière noire permettent d'étudier la formation de ces structures et d'interpréter les observations en les comparant aux structures simulées [16, 8].

Les simulations sont notamment employées au sein d'algorithmes de Monte-Carlo par chaînes de Markov (MCMC) pour estimer la valeur de paramètres physiques [4]. Ces algorithmes sont néanmoins coûteux, en ce qu'ils nécessitent de réaliser une simulation pour chaque échantillon généré. Il est donc essentiel d'avoir accès à un simulateur efficace.

Les simulations de matière noire sont des simulations à  $N$  corps, faisant évoluer un ensemble de  $N$  particules traçant la distribution de la matière noire [7]. Le résultat de ces simulations peut être prédit par un réseau de neurones pour obtenir un modèle approché efficace [9, 10]. Les vitesses prédites sont en particulier nécessaires pour confronter les modèles aux catalogues de vitesses mesurées [16, 17]. L'approche existante consiste à utiliser un réseau ne prédisant que la position des particules au cours du temps et d'utiliser la différentiation automatique (DA) pour en déduire la vitesse [10]. L'utilisation de la DA induit néanmoins un surcoût en mémoire et en temps de calcul lors de l'évaluation.

Ce travail propose d'entraîner un réseau à prédire de manière conjointe le déplacement et la vitesse des particules de matière noire issus d'une simulation numérique. Nous introduisons un terme de couplage additionnel dans la fonction de coût du réseau afin de garantir la cohérence entre la dérivée temporelle de la position et la vitesse prédite. Nos résultats expérimentaux valident l'efficacité du terme de couplage, et montrent que notre méthode obtient une précision similaire aux méthodes de l'état de l'art tout en étant plus rapide.

L'article est organisé comme suit. La partie 2 introduit brièvement les simulations de matière noire. La partie 3 décrit l'approche utilisée pour entraîner le réseau proposé. Les performances de cette approche sont évaluées en partie 4, en comparaison avec d'autres approches de la littérature. Conclusions et perspectives sont résumées en partie 5.

## 2 Simulation de matière noire

### 2.1 Modèle physique

La matière noire est modélisée comme un fluide sans collision qui évolue sous l'effet des interactions gravitationnelles. Une manière efficace de simuler numériquement un fluide est de faire évoluer de manière conjointe un ensemble de particules traçant son évolution [13]. En coordonnées comobiles, la trajectoire d'une particule  $\mathbf{x}$  de masse  $m$  est décrite par les équations du mouvement

$$\mathbf{p} = ma^2\dot{\mathbf{x}}, \quad \dot{\mathbf{p}} = -m\nabla\phi, \quad (1)$$

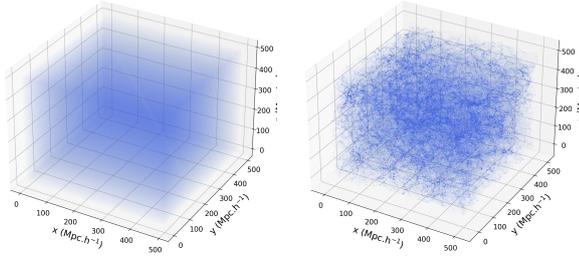
où  $a$  est le facteur d'échelle dépendant du temps, lié à l'expansion de l'Univers, et  $\mathbf{p}$  est la quantité de mouvement [14]. Le potentiel gravitationnel  $\phi$  est solution de l'équation de Poisson

$$\nabla^2\phi = 4\pi G a^2 (\rho - \rho_b), \quad (2)$$

où  $G$  est la constante gravitationnelle de Newton, et  $\rho = \rho(\mathbf{x})$  est la densité du fluide au point  $\mathbf{x}$ , et  $\rho_b$  est la densité moyenne du fluide. Le potentiel  $\phi$  est ainsi relié à la densité  $\rho$  via une intégrale spatiale, ce qui rend ces simulations très coûteuses.

### 2.2 Simulations à $N$ corps

En pratique, les simulations de matière noire sont réalisées dans un volume cubique, avec des conditions périodiques aux bords (voir figure 1).



(a) Début,  $t = 12.6$  M années (b) Fin,  $t = 13.7$  giga-années

FIGURE 1 : Simulation à  $N$  corps de la distribution de matière noire. Gauche : au début de la simulation, les particules sont réparties uniformément sur un maillage régulier. Droite : à la fin de la simulation, les particules sont regroupées en amas sous l'effet de la gravité.

**Initialisation.** La simulation est initialisée en échantillonnant la densité de matière initiale  $\rho_{init}$  comme une réalisation d'un champ aléatoire Gaussien dont les statistiques sont données par le modèle  $\Lambda$ CDM. Les  $n^3$  particules forment initialement un maillage régulier en 3 dimensions couvrant le volume de la simulation. Le déplacement et la vitesse initiale des particules sont calculés avec la théorie Lagrangienne de la perturbation du premier ordre, applicable aux origines de l'univers quand les fluctuations spatiales de la densité sont faibles [19, 2, 7].

**Calcul des forces.** La simulation à  $N$  corps se base sur une estimation de Monte-Carlo de la densité  $\rho$  dans l'équation (2), calculée à partir des  $N$  particules  $\{\mathbf{x}_i\}_{i=1,\dots,N}$  de masse respective  $m_i$  :

$$\hat{\rho}(\mathbf{x}) = \sum_i m_i \delta_D(\mathbf{x} - \mathbf{x}_i). \quad (3)$$

La force qui s'applique à une particule est alors donnée par la somme des interactions gravitationnelles avec les autres particules,

$$\dot{\mathbf{p}}_i = -m_i G \sum_j \frac{m_j (\mathbf{x}_i - \mathbf{x}_j)}{|\mathbf{x}_i - \mathbf{x}_j|^3 + \epsilon}, \quad (4)$$

où  $\epsilon > 0$  est un paramètre introduit pour éviter les singularités quand  $\mathbf{x}_i = \mathbf{x}_j$ . Une fois les forces calculées, les positions et les quantités de mouvement des particules sont mises à jour en utilisant un schéma d'intégration numérique.

**Stratégies d'accélération.** L'étape la plus coûteuse dans la simulation est le calcul des forces (4), dont la complexité est de l'ordre de  $O(N^2)$ . En pratique, il est possible de calculer une bonne approximation de ces forces en  $O(N \log N)$  opérations, en utilisant une structure d'arbre hiérarchique. Par ailleurs, les méthodes Particle-Mesh (PM) calculent les forces de manière rapide en résolvant l'équation de Poisson (2) sur un maillage, au prix d'une perte de précision due à la résolution limitée du maillage [7]. Malgré ces stratégies d'accélération, les simulations restent très coûteuses, en particulier pour des simulations de grande taille impliquant  $N = 512^3$  particules ou plus.

### 3 Réseau de neurones pour l'émulation de matière noire

L'objectif est d'entraîner un réseau de neurones dont l'évaluation sera beaucoup moins coûteuse pour prédire rapidement l'état final des particules issu d'une simulation de matière noire à partir des conditions initiales de la simulation. Les détails techniques de l'approche proposée sont détaillés ci-dessous.

**Données d'entraînement.** Les résultats de 200 simulations de  $N = 512^3$  particules dans un domaine cubique de taille  $500 \text{ Mpc}^3 \cdot \text{h}^{-3}$  ont été utilisées pour entraîner et valider le réseau proposé. Les simulations ont été réalisées avec le code OpenGadget-3 [5], en utilisant les paramètres du modèle  $\Lambda$ CDM déterminés par la collaboration *Planck* [1]. Le coût d'une simulation représente 300 heures CPU. Pour tester notre approche, les simulations sont réduites à  $N = 128^3$  particules en sous-échantillonnant les champs de déplacement et de vitesse d'un facteur  $4^3$ . Chaque simulation comporte 22 sauvegardes d'état capturées tout au long de la simulation, incluant les positions et les vitesses des particules. Nous entraînons le réseau seulement sur les 3 dernières sauvegardes de chacune des 200 simulations, correspondant aux instants  $t$  à compter de la formation de l'Univers  $t \in \{13, 13.4, 13.7\} \times 10^9$  années, auxquels le réseau sera utilisé par la suite. Les données sont normalisées pour l'entraînement.

**Champs de déplacement et de vitesse.** Dans la suite de cet article, les particules de la simulation sont indexées par leur position sur le maillage à l'initialisation, en notant  $\mathbf{x}_{i,j,k}(t)$  la position au temps  $t$  de la particule initialement localisée sur le sommet  $(i, j, k)$  du maillage. Le champ de déplacement  $\Psi_t : \mathbb{R} \mapsto \mathbb{R}^{n \times n \times n \times 3}$  est construit de sorte que

$$\mathbf{x}_{i,j,k}(t) = \mathbf{x}_{i,j,k}(t_{init}) + \Psi_t[i, j, k], \quad (5)$$

et le champ de vitesse  $V_t \in \mathbb{R}^{n \times n \times n \times 3}$  est défini par

$$V_t[i, j, k] := \dot{\mathbf{x}}_{i,j,k}(t). \quad (6)$$

Représenter les données sous la forme d'un tenseur permet d'utiliser un réseau de neurones de type UNet, qui a démontré des résultats au niveau de l'état de l'art pour de nombreuses applications (voir par exemple [15]).

**Architecture du réseau.** Nous utilisons un réseau 3D-UNet [3] noté  $F$ , prenant en entrée le déplacement initial  $\Psi_0 := \Psi_{t_{init}}$  et prédisant les champs de déplacement et de vitesse associés :

$$F : \mathbb{R}^{n \times n \times n \times 3} \times [t_0 : T] \longrightarrow \mathbb{R}^{n \times n \times n \times 3} \times \mathbb{R}^{n \times n \times n \times 3} \\ \Psi_0, t \longmapsto \Psi_t^{NN}, V_t^{NN}. \quad (7)$$

Les simulations ayant été réalisées avec des conditions aux bords périodiques, ces mêmes conditions aux bords sont utilisées pour les opérations de convolutions. La dépendance en  $t$  est réalisée à partir d'un mécanisme de modulation des couches d'attributs styleGAN [11], dont les statistiques sont calculées dans chaque bloc par un perceptron multi-couches prenant en entrée un vecteur calculé par *sinusoidal positional embedding* à partir de  $t$  [18].

**Critère d’entraînement.** En suivant l’approche proposée dans [9], nous entraînons notre réseau à prédire le résidu de l’approximation de Zel’dovich [19]. Cette approximation permet de calculer à peu de frais une première approximation du déplacement et de la vitesse des particules :

$$\Psi_t^{ZA} := D(t)\Psi_0, \quad V_t^{ZA} := \dot{D}(t)\Psi_0, \quad (8)$$

où  $D(t) \in \mathbb{R}$  est le facteur de croissance linéaire connu en forme close (voir par exemple [6]), et  $\Psi_0$  est le champ de déplacement initial  $\Psi_0 := \Psi_{t_{init}}$ . (8) produit une bonne approximation des composantes à grandes échelles des champs. L’entraînement résiduel permet ainsi de concentrer l’apprentissage sur la prédiction des détails hautes-résolutions du champ. Les termes d’entraînement supervisé pour la prédiction du déplacement  $\Psi$  et de la vitesse  $V$  sont définis comme :

$$\mathcal{L}_\Psi(\Psi_t^{GT}, \Psi_t^{NN}) = \|\Psi_t^{GT} - \Psi_t^{ZA} - \Psi_t^{NN}\|_2^2, \quad (9)$$

$$\mathcal{L}_V(V_t^{GT}, V_t^{NN}) = \|V_t^{GT} - V_t^{ZA} - V_t^{NN}\|_2^2. \quad (10)$$

De plus, nous proposons d’introduire un terme de couplage pour assurer la cohérence entre vitesse et déplacement,

$$\mathcal{L}_c(\Psi_t^{NN}, V_t^{NN}) = \|\dot{\Psi}_t^{NN} - V_t^{NN}\|_2^2, \quad (11)$$

où  $\dot{\Psi}_t^{NN}$  est calculé avec la différentiation automatique directe. Finalement, les paramètres de notre réseau sont optimisés pour minimiser la somme pondérée de ces 3 critères sur le jeu de données d’entraînement :

$$\mathcal{L}_{tot} = \omega_\Psi \mathcal{L}_\Psi(\Psi_t^{GT}, \Psi_t^{NN}) + \omega_V \mathcal{L}_V(V_t^{GT}, V_t^{NN}) + \omega_c \mathcal{L}_c(\Psi_t^{NN}, V_t^{NN}), \quad (12)$$

avec  $\omega_\Psi$ ,  $\omega_V$  et  $\omega_c > 0$ . En pratique, nous fixons  $\omega_\Psi = \omega_V = 1$ , et ajustons  $\omega_c$  pour contrôler le compromis entre précision des prédictions et cohérence entre position et vitesse prédite.

## 4 Expériences numériques

**Méthodes comparées.** L’approche proposée est comparée avec d’autres émulateurs rapides de l’état de l’art utilisés dans des algorithmes MCMC. Nous considérons l’approximation de Zel’dovich (ZA), une implémentation GPU d’une méthode Particle-mesh (flowPM) [12], et une méthode d’apprentissage profond qui utilise la différentiation automatique pour calculer la vitesse des particules (NN-DA) [10]. Pour flowPM, le nombre d’itérations est fixé à 10, avec un maillage de résolution  $128^3$  pour le calcul des forces. L’approche NN-DA est implémentée avec la même architecture 3D-UNet que la méthode proposée, et entraînée avec les mêmes données et le même nombre d’itérations afin d’isoler l’effet de l’implémentation du calcul de la vitesse sur les résultats obtenus. Toutes les méthodes sont évaluées avec un GPU Nvidia RTX A6000, disposant d’une capacité mémoire de 48 GB.

**Métriques.** Pour évaluer les méthodes comparées, nous mesurons l’erreur absolue moyenne entre les champs proposés et la vérité terrain. Pour un champ  $X \in \mathbb{R}^{n \times n \times n \times 3}$  de vérité terrain associée  $X^{GT}$ , l’erreur absolue moyenne est définie par

$$L_1(X) = \frac{1}{3n^3} \|\text{vec}(X - X^{GT})\|_1, \quad (13)$$

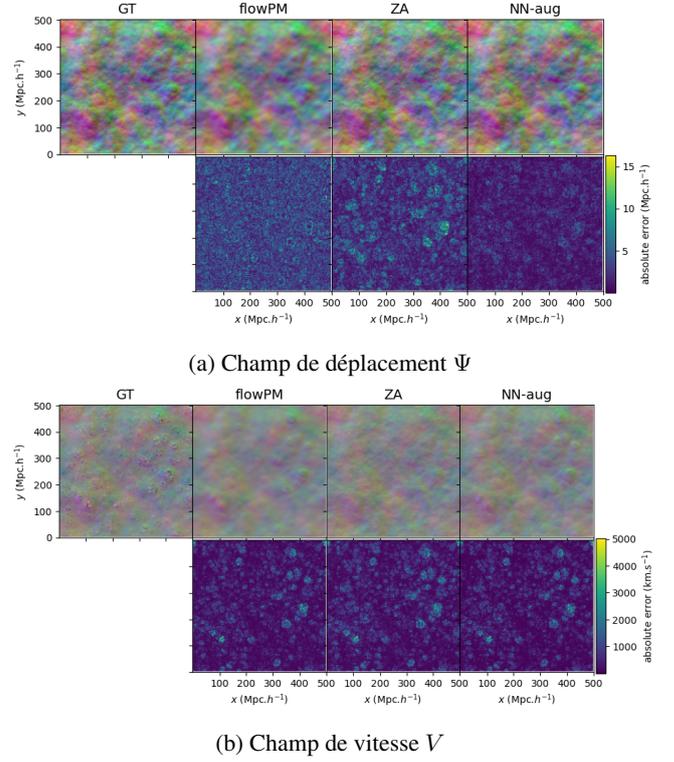


FIGURE 2 : Reconstruction du déplacement et de la vitesse des particules de matière noire par les différentes méthodes évaluées (GT : vérité terrain).

TABLE 1 : Résultats selon la valeur du terme de couplage  $\omega_c$ .

$\omega_c$	$L_1(\Psi_t)$ (kpc.h <sup>-1</sup> )	$L_1(V_t)$ (km.s <sup>-1</sup> )	$\mathcal{L}_c$ (km.s <sup>-1</sup> )
0.00	430.72	114.89	100.06
1.00	429.24	116.49	35.66
10.00	429.88	122.88	14.92
100.00	431.96	134.62	4.92

avec vec l’opérateur de vectorisation d’un tenseur, et  $X^{GT}$  le résultat de la simulation à  $N$  corps obtenu avec le code OpenGadget-3 [5]. Les résultats reportés par la suite correspondent à la valeur moyenne des erreurs sur un ensemble de test, sur les mêmes temps  $t$  que ceux utilisés pour entraîner le réseau. Par ailleurs, on mesure l’utilisation mémoire GPU et le temps d’exécution d’une évaluation.

**Effet du terme de couplage.** Nous étudions l’effet du poids du terme de couplage  $\omega_c$  sur la précision de la reconstruction et la cohérence position-vitesse. Les résultats rapportés en table 1 montrent l’effet positif du terme de couplage (11) sur la cohérence entre le déplacement et la vitesse prédits par le réseau. Comme attendu, pour  $\omega_\Psi$  et  $\omega_V$  fixés ( $\omega_\Psi = \omega_V = 1$ ), une augmentation du poids du terme de couplage  $\omega_c$  entraîne bien une amélioration de la cohérence position-vitesse, au prix d’une légère perte de précision sur la prédiction de la vitesse.

**Résultats numériques.** La table 2 présente une comparaison numérique des différentes méthodes évaluées. L’approximation de ZA est la méthode la plus efficace en temps et mémoire

TABLE 2 : Comparaison quantitative de performances de notre réseau (NN-aug) avec d’autres émulateurs de matière noire (**meilleur**, second).

	$L_1(\Psi_t)$ (kpc.h <sup>-1</sup> )	$L_1(V_t)$ (km.s <sup>-1</sup> )	temps (ms)	GPU mem (GB)
ZA	909	147	<b>1</b>	<b>0.1</b>
FlowPM	1159	130	24000	47
NN-DA [10]	<b>428</b>	<u>117</u>	95	7.5
NN-aug	<u>429</u>	<b>116</b>	<u>35</u>	<u>7.2</u>

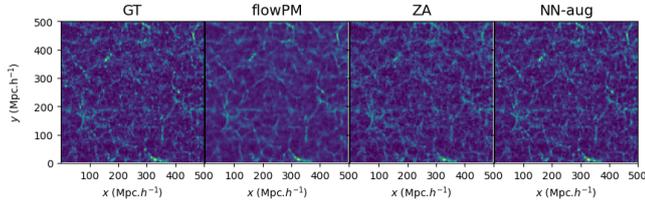


FIGURE 3 : Densité de matière noire  $\rho$  à  $t = 13.7 \times 10^9$  simulé par les méthodes évaluées. Chaque figure correspond à une tranche de  $3.9 \text{Mpc.h}^{-1}$  du volume 3D.

avec les prédictions les moins précises, hormis pour la prédiction du déplacement meilleure qu’avec flowPM. Les méthodes NN produisent les meilleurs résultats, en étant plus de 2 fois plus précises que flowPM et ZA sur le déplacement, et en offrant aussi une précision supérieure pour la vitesse. Les approches NN-aug et NN-DA sont aussi précises sur la position et la vitesse, mais NN-aug est environ 3 fois plus rapide et demande légèrement moins de mémoire GPU que NN-DA, du fait que NN-aug n’utilise pas la DA pour calculer la vitesse. En revanche, NN-DA garantit une cohérence position vitesse par construction. Il serait possible d’améliorer la précision de flowPM en augmentant la résolution du maillage utilisé pour le calcul des forces, mais cette opération est trop coûteuse pour être réalisée avec un GPU. Cette option n’a pas été explorée, dans la mesure où l’objectif de cette étude reste d’évaluer des méthodes légères déployables sur un seul GPU.

**Comparaison visuelle** Une comparaison visuelle des résultats montre que la méthode proposée NN-aug permet d’obtenir une reconstruction plus précise du champ de déplacement (figure 2a), ce qui se traduit par une meilleure reconstruction du champ de densité (figure 3). Les méthodes NN-aug et flowPM surpassent toutes deux l’approximation linéaire de ZA pour la prédiction de la vitesse (figure 2b).

## 5 Conclusion

Nous avons présenté un réseau de neurones capable de prédire en quelques dizaines de millisecondes le résultat d’une simulation à  $N$  corps de la distribution de matière noire dans l’Univers. La stratégie d’entraînement augmenté proposée permet d’obtenir jusqu’à 3 fois plus rapidement des résultats d’une précision équivalente aux méthodes de l’état de l’art. Ce gain en efficacité permet d’envisager une extension des méthodes réseaux neurones pour des simulations de résolution plus importante.

**Remerciements** Ce travail est soutenu par la Chaire IA Sherlock ANR-20-CHIA-0031-01 portée par P. Chainais, ainsi que par le programme national d’investissement d’avenir ANR-16-IDEX-0004 ULNE et la Région Hauts-de-France. Ces travaux ont bénéficié d’un accès aux ressources de calcul et de stockage au TGCC au travers des allocations de ressources 2023-A0150414664 et 2024-AD010415422 attribuées par GENCI sur les partitions ROME et V100 du calculateur Joliot Curie.

## Références

- [1] P. a. R. ADE, N. AGHANIM, C. ARMITAGE-CAPLAN et al. : Planck 2013 results. XVI. Cosmological parameters. *Astronomy & Astrophysics* 571 (novembre 2014).
- [2] F. BERNARDEAU, S. COLOMBI, E. GAZTANAGA et al. : Large-Scale Structure of the Universe and Cosmological Perturbation Theory. *Physics Reports* 367.1-3 (septembre 2002).
- [3] Özgün ÇIÇEK, Ahmed ABDULKADIR, Soeren S. LIENKAMP et al. : 3D U-Net : Learning Dense Volumetric Segmentation from Sparse Annotation. Cham, 2016.
- [4] R. GRAZIANI, H M. COURTOIS, G. LAVAUX et al. : The peculiar velocity field up to  $z \sim 0.05$  by forward-modelling Cosmicflows-3 data. *Monthly Notices of the Royal Astronomical Society* 488.4 (octobre 2019).
- [5] Frederick GROTH, Ulrich P. STEINWANDEL, Milena VALENTINI et al. : The Cosmological Simulation Code OpenGadget3 – Implementation of Meshless Finite Mass. *Monthly Notices of the Royal Astronomical Society* 526.1 (septembre 2023).
- [6] A. J. S. HAMILTON. : Formulae for growth factors in expanding universes containing matter and a cosmological constant. *Monthly Notices of the Royal Astronomical Society* 322.2 (avril 2001).
- [7] Wojciech HELLWING. : A short introduction to numerical methods used in cosmological N-body simulations ().
- [8] Yehuda HOFFMAN, Edoardo CARLESI, Daniel POMARÈDE et al. : The quasi-linear nearby Universe. *Nature Astronomy* 2.8 (juillet 2018).
- [9] Drew JAMIESON, Yin LI, Renan Alves de OLIVEIRA et al. : Field Level Neural Network Emulator for Cosmological N-body Simulations. *The Astrophysical Journal* 952.2 (août 2023).
- [10] Drew JAMIESON, Yin LI, Francisco VILLAESCUSA-NAVARRO et al. : Field-level Emulation of Cosmic Structure Formation with Cosmology and Redshift Dependence. Août 2024.
- [11] Tero KARRAS, Samuli LAINE et Timo AILA. : A Style-Based Generator Architecture for Generative Adversarial Networks. 2019.
- [12] Chirag MODI, Francois LANUSSE et Uros SELJAK. : FlowPM : Distributed TensorFlow implementation of the FastPM cosmological N-body solver. *Astronomy and Computing* 37 (2021).
- [13] J J MONAGHAN. : Smoothed particle hydrodynamics. *Reports on Progress in Physics* 68.8 (août 2005).
- [14] Peebles P.J.E. : The Large-Scale Structure of the Universe. Princeton university press, novembre 1980.
- [15] Satya P. SINGH, Lipo WANG, Sukrit GUPTA et al. : 3D Deep Learning on Medical Images : A Review. *Sensors* 20.18 (janvier 2020).
- [16] Jenny G. SORCE, Stefan GOTTLÖBER, Gustavo YEPES et al. : Cosmicflows Constrained Local UniversE Simulations. *Monthly Notices of the Royal Astronomical Society* 455.2 (janvier 2016).
- [17] Edward N. TAYLOR, Michelle CLUVER, Eric BELL et al. : The 4MOST Hemisphere Survey of the Nearby Universe (4HS). *Published in The Messenger vol. 190* pp. 46-48 (2023).
- [18] Ashish VASWANI, Noam SHAZEER, Niki PARMAR et al. *Attention Is All You Need*. Août 2023.
- [19] Ya. B. ZEL’DOVICH. : Gravitational instability : An approximate theory for large density perturbations. *Astronomy and Astrophysics* 5 (mars 1970).