Interpolating between cosmological simulation snapshots using machine learning

being a dissertation submitted in fulfillment of

requirements of degree of **Masters of Artificial Intelligence and Data Science** in the **University of Hull**

by **Daniyal Kaleem May 2025**

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr Aoife Curran for her invaluable guidance, encouragement, and support throughout this project. Their insights and expertise in both cosmology and machine learning were instrumental in shaping the direction of my research and helping me overcome numerous challenges along the way.

I am also grateful to the Centre of excellence for Data Science, Artificial Intelligence and Modelling at the University of Hull for providing the resources and environment necessary to carry out this work. The analysis presented in this report was facilitated by the University of Hull's High Performance Computing facility, Viper. Specifically, the large dataset processing and complex model training were enabled by Viper's specialized hardware, which significantly reduced processing time compared to traditional computing methods.

Abstract

This project explores the application of convolutional neural networks (CNNs) to interpolate between snapshots of N-body cosmological simulations, aiming to reduce data storage requirements without compromising the fidelity of simulated structures. Using the CAMELS dataset, particle distributions were converted into 2D density maps via histogramming and normalization, allowing for efficient training of deep learning models. U-Net architecture was implemented and progressively refined through hyperparameter tuning, deeper layers, and skip connections to improve accuracy. The final model achieved a substantial reduction in Root Mean Square Error (RMSE) and a significant increase in Peak Signal to Noise Ratio (PSNR), accurately reconstructing intermediate states between simulation snapshots. The findings demonstrate the potential of CNN-based interpolation as a scalable and efficient tool for cosmological modeling, contributing to computational astrophysics by alleviating storage and processing demands while preserving scientific validity.

Table of Content

List of Figures

List of Tables

1. Introduction

N-body simulations are fundamental tools in computational astrophysics, allowing researchers to model the evolution of cosmic structures under the influence of gravity. These simulations track the motion of dark matter across cosmic time, generating large-scale structures such as galaxy clusters and filaments. However, due to the immense computational resources required to store and process simulation outputs, alternative methods for reducing data storage without sacrificing accuracy have become a crucial area of research. Each particle in an N-body simulation requires storage for its position and velocity, which can require 24/48 bytes per particle (B) if stored as single/double precision floating numbers (Chen et al., 2020). The bytes per particle (B) for a single or double precision is calculated by the following rule:

$$Single\ Precision = 24\ bytes\ (3\ positions + 3\ velocities * 4\ bytes\ each)$$

$$Double\ Precision = 48\ bytes\ (3\ positions + 3\ velocities * 4\ bytes\ each)$$

The total storage per particle can then be calculated by multiplying the number of particles (N_p) with the number of timesteps or snapshots T with Bytes per particle B

$$Storage\ per\ particle = N_p * T * B$$

Traditional approaches to mitigating storage constraints involve reducing the temporal frequency of saved snapshots, leading to potential loss of information. Recent advancements in machine learning, particularly deep learning, offer promising solutions to this problem. Convolutional neural networks (CNNs) have shown success in various scientific applications, including image recognition (Pahwa et al., 2019) and physical simulations (Ciresan et al., 2011), due to their ability to capture spatial patterns effectively.

This project builds on the work of Chen et al. (2020), which proposed using deep learning techniques to interpolate between N-body simulation snapshots. The paper demonstrated that a CNN-based approach could predict intermediate snapshots with high accuracy, thus significantly reducing storage demand without losing essential physical information. Their model treated the interpolation as a regression task on particle properties, illustrating that machine learning can bridge sparse simulation outputs efficiently.

Further supporting the utility of deep learning in the cosmological context, Jamieson et al. (2023) developed a field-level neural network emulator capable of predicting nonlinear displacements and velocities directly from linear inputs across a range of cosmologies. Their emulator, incorporating style parameters encoding cosmological dependence, achieved high fidelity compared to full N-body simulations, outperforming traditional fast approximators like COLA. Importantly, their findings highlight the potential of CNN architectures to not only compress simulation outputs but also accelerate large-scale cosmological analyses by enabling fast and differentiable predictions.

The ability to generalize across cosmological parameters and maintain accuracy at small scales (k $\sim 1\ h\ Mpc^{-1}$) further justifies the application of CNN-based methods to datasets like CAMELS.

Beyond astrophysics, innovations from other domains inform architectural choices for volumetric data. For example, Sobirov et al. (2023) introduced the concept of "super images" in medical imaging analysis, effectively converting 3D volumetric data into 2D images to leverage the simplicity and computational efficiency of 2D CNNs. Their method achieved performance comparable to full 3D networks while significantly reducing computational cost and model complexity. This strategy underscores a general principle highly relevant to the current project: dimensionality reduction techniques, when carefully designed, can preserve critical spatial information while easing computational demands, a principle similarly employed when designing 2D CNN-based interpolation for N-body snapshots.

Moreover, the broader field of machine learning cosmology has shown that ML models can emulate nonlinear gravitational evolution with considerable speed and accuracy. The review by Lucie-Smith et al. (2018) emphasizes that, while machine learning methods can drastically reduce computational demands, careful consideration must be given to the generalization ability of models across different cosmological parameters. Overfitting the model to specific training simulations can lead to biased predictions if not properly addressed.

Foundational to the use of CNNs in spatial prediction tasks is the U-Net architecture, originally developed by Ronneberger et al. (2015) for biomedical image segmentation. U-Net's encoder-decoder structure, with skip connections to recover spatial precision, has proven to be exceptionally effective when training data are limited. Its success in tasks requiring fine-grained spatial accuracy suggests that similar architectures could be well-suited to interpolating N-body simulations, where capturing the intricate gravitational evolution at small scales is crucial.

In this project, I apply and extend the methodology proposed by Chen et al. (2020), training CNN models on 3D grid-based representations of the CAMELS particle data to predict intermediate simulation snapshots. I experiment with different CNN architectures, informed by the successes of field-level emulation (Jamieson et al., 2023), efficient 2D approximations of 3D data (Sobirov et al., 2023), and encoder-decoder segmentation networks (Ronneberger et al., 2015), to optimize both predictive accuracy and computational efficiency.

The significance of this research lies in its potential to alleviate the computational burden associated with high-resolution simulations. By reducing storage demands without compromising accuracy, CNN-based interpolation methods could transform the feasibility of storing and analyzing large simulation datasets, facilitating faster and more cost-effective research in cosmology. Furthermore, such approaches align with a broader trend toward machine learning-driven acceleration of scientific simulations, potentially opening new pathways for real-time analysis and inference from observational data.

2. Methodology and Approach

2.1. Research Design and Methodological Framework

This project adopts a quantitative, experimental, and design-based methodological framework to predict the temporal evolution of dark matter distributions in cosmological N-body simulations using deep learning. A convolutional neural network, specifically a U-Net architecture, is trained to interpolate between particle distributions at different simulation time steps, learning spatial-temporal transformations from numerical data. Model performance is evaluated through statistical metrics including Root Mean Squared Error and Peak Signal to Noise Ratio. The design-based component focuses on structuring the data pipeline and model architecture to effectively capture spatial density variations. By combining domain-specific physical data with machine learning techniques, this approach enables efficient exploration of predictive capabilities for simulation-based cosmology while maintaining rigorous computational standards.

2.2. Data Processing and Model Design

To implement this project, several interrelated computational methods and tools were employed, each serving a specific purpose in the broader workflow. The first step involved extracting particle position data from HDF5 simulation files from the CAMELS Multifield simulation dataset, a widely used data format in astrophysics due to its efficiency in handling large, hierarchical datasets. Using the h5py Python library, the three-dimensional coordinates of dark matter particles were extracted from the PartType1/Coordinates group in each snapshot file. These files correspond to discrete time steps in an N-body simulation. Nine snapshots ranging from snapshot_040 to snapshot_088, at intervals of 6 snapshots, were used to construct the training and validation datasets, while snapshot_090, the final snapshot in the simulation, was reserved for testing.

Once the 3D coordinates were extracted, 2D density maps were constructed. Given the isotropy in particle distribution along the z-axis and the focus on large-scale 2D filamentary patterns, the x and y coordinates were deemed sufficient for representing the density field. This was done using numpy.histogram2d, which binned the x and y coordinates into a 256×256 grid, thereby producing a pixel-based image representation of particle density. The 2D density map was constructed by binning the projected coordinates into a discrete grid. Mathematically, for a particle i with 2D coordinates (xi, yi), the density $\rho_{j,k}$ at grid bin (j, k) is computed as:

$$\rho_{j,k} = \sum_{i=1}^{N} \delta_{j,k}(xi, yi)$$

Where $\delta_{j,k}(xi, yi)$ is an indicator function equal to 1 if particle i's coordinates fall into bin (j, k), and 0 otherwise. This effectively counts the number of particles in each grid cell, generating a spatially resolved particle density distribution.

To address the wide dynamic range in particle counts across different regions of the simulation box, a logarithmic transformation was applied to the raw density values. This transformation compresses high-density peaks and enhances lower-density structures. Additionally, to normalize the transformed values into the [0, 1] range required by the neural network, min-max normalization was performed. The final preprocessed density value $D'_{j,k}$ for bin (j, k) is given by:

$$D'_{j,k} = \frac{\log_{10}(\rho_{j,k} + \epsilon) - min}{max - min}$$

Where $\epsilon = 10^{-5}$ is a small constant to avoid taking the logarithm of zero, and min and max are the minimum and maximum log-transformed values across the dataset. This preprocessing pipeline ensures numerical stability during training and allows the network to effectively learn from both high- and low-density regions in the cosmological simulation data.

Following map construction, data normalization was applied to prepare the maps for neural network input. To handle the high variance in particle densities, a logarithmic transformation was applied using log10(x + 1e-5), which compressed the dynamic range and avoided zero values. Subsequently, min-max normalization was used to rescale the maps to the [0, 1] range. These transformations helped stabilize training and maintain numerical consistency.

The dataset was then split into input-target pairs for supervised learning, where each map at time (t) was the input and the map at t+1 served as the target. These were encapsulated in a custom N-Body Dataset class based on PyTorch's Dataset interface, enabling efficient mini-batch loading via the DataLoader class. A 90/10 split was used for training and validation, with a separate test set held out.

The supervised learning task in this project is framed as a regression problem, where the model learns to predict the particle density map at a future time step given the map from a preceding time. Each training sample consists of a pair of consecutive simulation snapshots, with the earlier snapshot serving as the input and the later one as the target. Formally, let X_t represent the normalized 2D density map at time step t. The model's goal is to learn a function f_θ, parameterized by a set of trainable weights θ, that maps X_t to its corresponding future state $\hat{X}_{t+1}$ such that:

$$\hat{X}_{t+1} = f_\theta(X_t)$$

In this formulation, $\hat{X}_{t+1}$ is the predicted density map at the next simulation time step. The function f_θ is realized by a convolutional neural network (CNN), which learns to approximate the underlying physical transformation encoded in the simulation data. Through this formulation, the model effectively interpolates between two discrete simulation outputs, aiming to generalize the temporal evolution of cosmic structures across the training set. This abstraction captures the essence of the interpolation task as one of learning temporal mappings governed by physical principles embedded in spatial patterns.

The modeling component used a convolutional neural network based on the U-Net architecture, known for preserving spatial detail and capturing filamentary features in density maps. The encoder consisted of convolutional blocks with ReLU activations and max-pooling. A bottleneck layer encoded abstract features, and the decoder used transposed convolutions and skip connections to reconstruct the maps. The model was trained in PyTorch using Mean Squared Error loss, the Adam optimizer, and evaluated with RMSE and PSNR metrics.

U-Net architecture used in this study is designed to capture and reconstruct fine-grained spatial features through an encoder-decoder structure. The encoder progressively reduces spatial resolution while extracting abstract representations, and the decoder upsamples these representations to reconstruct the original resolution. The core operation underlying both encoding and decoding is the convolution.

A 2D convolution can be mathematically expressed as:

$$y_{i,j} = \sum_{m,n} x_{i+m,j+n} * w_{m,n} + b$$

Where $x_{i+m,j+n}$ is the input value from the local neighborhood around pixel (i,j), $w_{m,n}$ represents the weights of the convolutional kernel, and b is the bias term. This operation allows the network to learn localized spatial features such as edges, gradients, and textures within the density maps.

One of the key strengths of the U-Net architecture lies in its **skip connections**, which enable the model to recover spatial details lost during downsampling. These connections directly link the encoder layers to their corresponding decoder layers.

Mathematically, if E_l is the feature map from the encoder at level l, and D_{l+1} is the decoder output from the level below, then the skip connection is implemented as:

$$D_l = Upsample(D_{l+1}) + E_l$$

Here, the decoder feature map is upsampled (typically via transposed convolutions or interpolation), and then added to the encoder feature map from the corresponding level. This mechanism preserves high-resolution information and improves the accuracy of the reconstructed outputs, particularly in regions with fine structures such as filaments or dense clusters. The additive formulation also facilitates more stable gradient propagation, aiding in faster and more effective training.

2.3. Rationale for Method Selection

Each method and tool used in this project was selected based on both theoretical rationale and practical considerations. The choice of HDF5 for data storage is standard in astrophysical simulations due to its efficiency in handling large-scale structured data. Using h5py allows direct and memory-efficient access to the required datasets without full in-memory loading.

Projecting the 3D particle distributions into 2D density maps was motivated by computational efficiency. While 3D CNNs are more expressive for volumetric data, they are also significantly more resource-intensive, making training and inference on limited hardware infeasible. Prior research in medical imaging has shown that 2D projections can preserve essential structural information while significantly reducing computational costs (Sobirov et al., 2023).

The use of histogram binning to construct density maps was selected for its simplicity, speed, and ability to represent the continuous mass distribution in a grid-based format compatible with CNNs. Alternative methods such as kernel density estimation were considered but deemed too computationally expensive for large datasets.

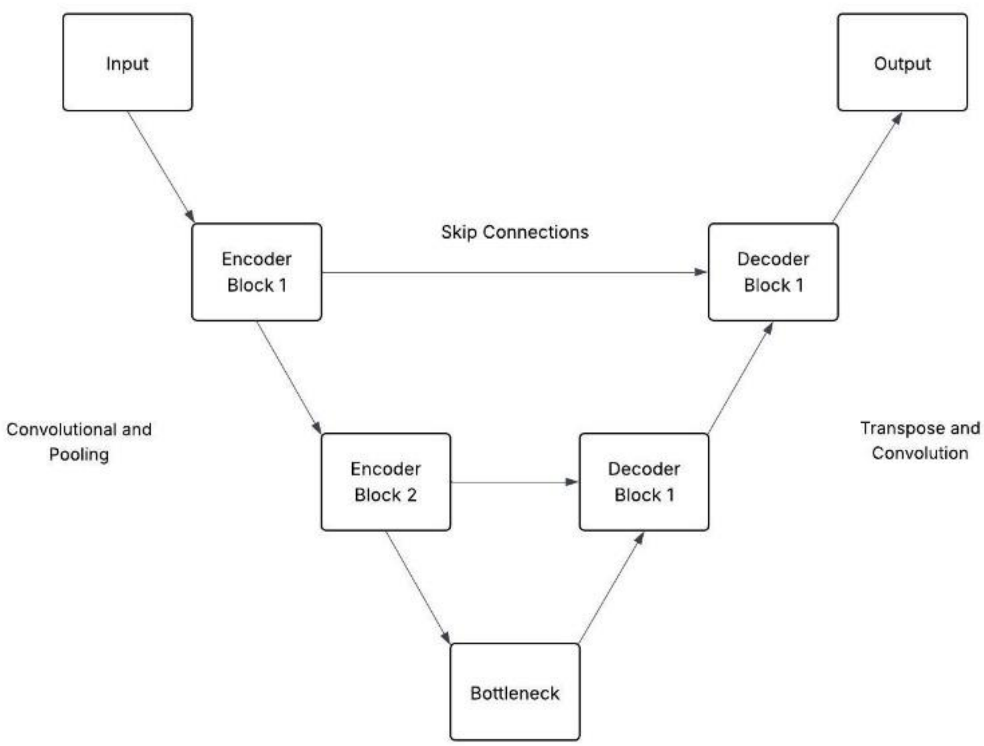

Figure 1: U-Net Model Architecture Layout

The U-Net architecture was chosen based on its strong performance in tasks involving spatial reconstruction and transformation (Ronneberger et al., 2015). U-Net's combination of down sampling and up sampling layers allows the network to capture both fine-grained details and large-scale context, which is essential in modeling the multi-scale structures of the cosmic web.

Compared to standard CNNs or fully connected networks, U-Net offers a better balance of performance and parameter efficiency. In the implemented U-Net model, skip connections are used to concatenate feature maps from the encoder directly to their corresponding decoder layers. This helps preserve important spatial details that might be lost during max-pooling. By combining these high-resolution features with up-sampled decoder outputs, the model is better able to reconstruct fine structures in the 2D density maps, such as cosmic filaments. These connections also support more stable training by improving gradient flow. As a result, the model achieves more accurate and detailed predictions.

To quantitatively evaluate the model's performance during training and testing, standard image reconstruction metrics were employed. The **Mean Squared Error (MSE)** was selected as the primary loss function due to its sensitivity to large deviations between predicted and ground truth pixel values. MSE penalizes errors quadratically, which is particularly effective for ensuring accuracy in high-density regions of the cosmological maps. It is mathematically defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Where y_i denotes the true pixel intensity, $\hat{y}_i$ is predicted value, and N is the total number of pixels in the image. This formulation measures the average squared difference across all pixels.

To make the metric more interpretable, particularly in the context of pixel intensity values, the **Root Mean Squared Error (RMSE)** was also computed:

$$RMSE = \sqrt{MSE}$$

RMSE has the advantage of being expressed in the same units as the original data, making it easier to compare absolute error magnitudes in reconstructed images.

In addition to RMSE, the **Peak Signal-to-Noise Ratio (PSNR)** was used as a supplementary metric to quantify the fidelity of image reconstruction. PSNR is widely used in image processing and compares the maximum possible signal power to the power of corrupting noise (in this case, the reconstruction error). It is defined as:

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX_i}{\sqrt{MSE}}\right)$$

RMSE, used for evaluation, shares the same units as the target, enhancing interpretability. PSNR complements RMSE by assessing reconstruction fidelity, measuring how closely predicted maps match target pixel intensities. Higher PSNR values reflect better preservation of fine structures such as dense regions and cosmic filaments.

Where MAX_i is the maximum possible pixel intensity, which is 1.0 in the case of normalized input images. A higher PSNR indicates that the predicted map closely matches the target image, particularly preserving fine spatial features and structures. This metric complements RMSE by offering an intuitive, logarithmic-scale assessment of reconstruction quality.

2.4. Methodological Reflections and Limitations

The approach used in this project was effective, with the U-Net architecture successfully modeling spatial transformations and capturing the non-linear evolution of density distributions. The 2D projection strategy preserved key structural features, and histogramming enabled efficient conversion of raw data into image-like inputs. However, reducing the data to 2D omitted potential z-axis information, limiting generalizability in 3D contexts. The model also assumed evenly spaced time steps and lacked cosmological context, which could hinder long-term predictions. Computational demands remained high due to resolution and network depth, though increased training epochs improved accuracy and enhanced qualitative performance despite these constraints.

3. Results

3.1. Summary of Key Findings

This project aimed to develop and refine a deep learning model capable of predicting future states in N-body cosmological simulations using 2D density maps derived from simulation snapshots. The baseline model, a simple U-Net with two encoder and decoder layers, served as a reference point. Through progressive tuning of architectural components and training parameters, a significantly more accurate version of the U-Net was developed, with deeper layers, skip connections, and an extended training period. The best model demonstrated strong predictive capability, with markedly improved metrics such as RMSE and PSNR on unseen test data. The final version achieved a test RMSE of **0.0051** and an average PSNR of **45.79 dB**. These improvements reflect the impact of deeper architecture, extended training, and skip connections on the model's ability to capture spatial patterns in cosmological simulations.

3.2. Experimental Results and Model Evolution

3.2.1. Baseline Performance Analysis

The initial model featured a relatively simple U-Net configuration: two convolutional layers in the encoder and decoder each, with max pooling and transposed convolutions for downsampling and upsampling, respectively. The model was trained for 10 epochs on 700 synthetic density maps, derived from simulation data via logarithmic normalization of histogrammed particle coordinates in a 256×256 grid. The resulting model achieved a test RMSE of **0.0563** and a PSNR of **24.98 dB**. Visually, predictions preserved large-scale density features but lacked finer structural fidelity. The output images were noticeably blurred compared to their corresponding targets, indicating the model's limitations in recovering high-frequency spatial details after only limited training. The output image as compared to the target image is highlighted in Figure 2 below.

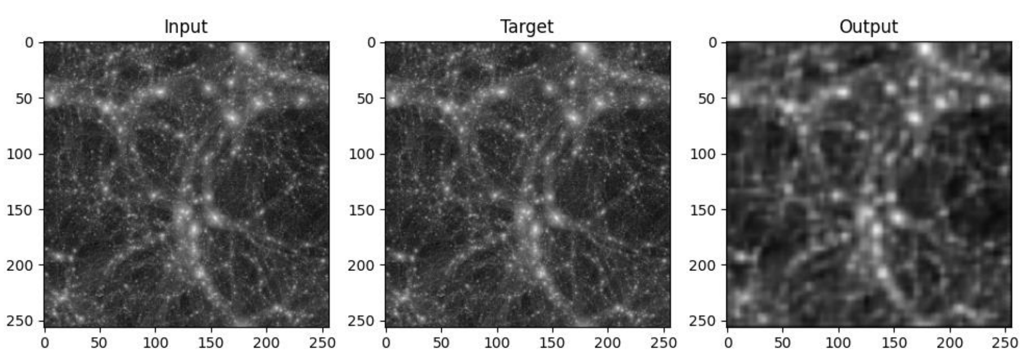

Figure 2: Simple U-Net Architecture: Input to Output

3.2.2. Enhanced Architecture and training strategy

Extensive experimentation led to substantial model improvements. The architecture was revised to include:

- Larger convolutional kernels (5×5 instead of 3×3),
- Additional skip connections to preserve spatial detail during upsampling,
- Deeper layers to allow for more expressive feature extraction,
- Training extended to 100 epochs to ensure convergence.

Also, different architectures were tried and tested, namely V-Net and ResNet architectures. Key architectural changes included concatenated feature maps from earlier encoder layers directly into the decoder via skip connections. This allowed the decoder to access high-resolution features otherwise lost during downsampling, resulting in sharper, more structurally faithful outputs.

The results from hyperparameter tuning highlight several additional trials involving variations in learning rate, kernel sizes, and optimizer choices. Learning rates of 1e-4 and the Adam optimizer consistently yielded the most stable convergence. Attempts with higher learning rates (1e-3) led to divergence, while lower learning rates (1e-5) resulted in underfitting, even with extended epochs.

The initial model was a basic U-Net without skip connections, trained using both 3×3 and 5×5 convolutional kernels. With a 3×3 kernel and 10 epochs of training, it achieved an RMSE of **0.0563**, improving slightly to **0.0487** after 20 epochs, suggesting the benefits of extended training. Switching to a 5×5 kernel improved spatial feature extraction, especially at higher epochs. With 100 epochs and ReLU activation, the model reached an RMSE of **0.0430** and a PSNR of **27.43 dB**, showing clear gains in predictive accuracy. The effect of using a LeakyRELU activation function was minimal on the performance of the model. The results of the hyperparameter tuning on this model are listed in Table 1.

Table 1: Simple U-Net Architecture hyperparameter tuning

Model Name	Kernel Size	Epochs	Activation Function	RMSE	PSNR
UNetReLU_model_1	3	10	ReLU	0.0563	24.98
UNetReLU_model_2	3	20	ReLU	0.0487	26.24
UNetReLU_model_3	5	10	ReLU	0.0538	25.38
UNetReLU_model_4	5	50	ReLU	0.0466	26.63
UNetReLU_model_5	5	100	ReLU	0.0430	27.43
UNetLeakyReLU_model_1	5	30	LeakyReLU	0.0459	26.77
UNetLeakyReLU_model_2	5	100	LeakyReLU	0.0432	27.46

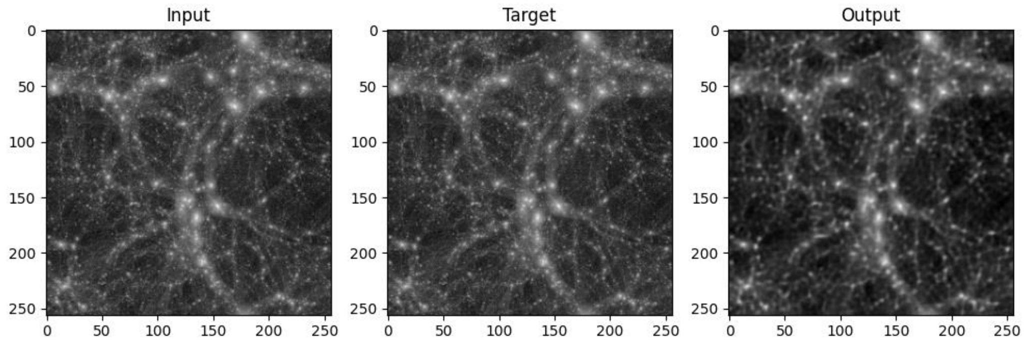

Figure 3: U-Net Architecture with LeakyReLU activation ran for 100 epochs: Input to Output

As seen from Figure 3, the output closely resembles the target, highlighting successful reconstruction of fine cosmic structures. This visual result validates the quantitative improvements in RMSE and PSNR, confirming enhanced spatial feature learning. Despite notable improvements, the output still lacks some finer details present in the target, indicating limitations in capturing high-frequency features.

A deeper U-Net architecture was tested with LeakyReLU activation function. To increase representational depth, this model introduced more convolutional layers and batch normalization. However, early results showed overfitting, with an RMSE of **0.0829**. The addition of dropout (with a probability of 0.2) significantly improved performance, reducing the RMSE to **0.0497**. This highlighted the importance of regularization in deeper networks and demonstrated that deeper architectures alone aren't always sufficient without proper training stabilization. The results are highlighted in Table 2.

As the model architecture was deepened to improve its representational capacity, the risk of overfitting increased, especially when training on relatively limited simulation data. To mitigate this, **dropout regularization** was employed. Dropout randomly deactivates a fraction of neurons during each training iteration, thereby preventing the network from becoming overly reliant on specific activations. This encourages the model to learn more generalizable features and improves robustness on unseen data. Mathematically, the dropout operation for a hidden unit h_i can be expressed as:

$$h_i^{drop} = h_i \cdot z_i, \qquad z_i \sim Bernoulli(1-p)$$

In this formulation, h_i is the original output of the neuron before dropout, and z_i is a binary random variable sampled from a Bernoulli distribution with probability $(1-p)$, where p is the dropout rate (e.g., 0.2). When $z_i = 0$, the neuron's output is effectively masked; otherwise, it is retained. This stochastic regularization technique helps reduce co-adaptation of neurons and results in more stable convergence, particularly beneficial when training deeper architectures as in the enhanced U-Net models used in this project.

Table 2: Deeper U-Net Architecture with Dropout hyperparameter tuning

Model Name	Dropout	Epochs	Activation Function	RMSE	PSNR
UNetDeep_model_1	None	20	LeakyRELU	0.0829	21.63
UNetDeep_model_2	0.2	50	LeakyRELU	0.0497	26.08

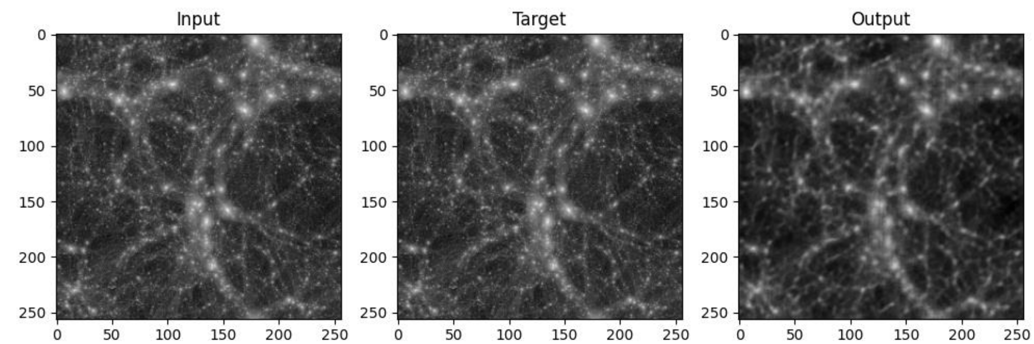

Figure 4: Deeper U-Net Reconstruction with dropout: Input to Output

Figure 4 shows that despite the added depth, the output image remains noticeably blurrier than the target, missing fine structural details. This suggests that deeper networks without sufficient training or architectural balance can struggle with precise reconstruction.

The ResNet model, named ResNet_model_1, incorporated residual connections in a traditional encoder-decoder architecture. It demonstrated reasonable performance with an RMSE of **0.0985** and PSNR of **20.13 dB** after 50 epochs. Although not outperforming the optimized U-Net models, the ResNet-style model showed better convergence behavior and reduced training time, suggesting its potential for faster deployment in scenarios prioritizing speed over maximum accuracy.

Despite residual connections, the output appears overly noisy and lacks structural clarity compared to the target, as seen in Figure 5. Fine filaments are obscured, and the overall reconstruction looks inconsistent.

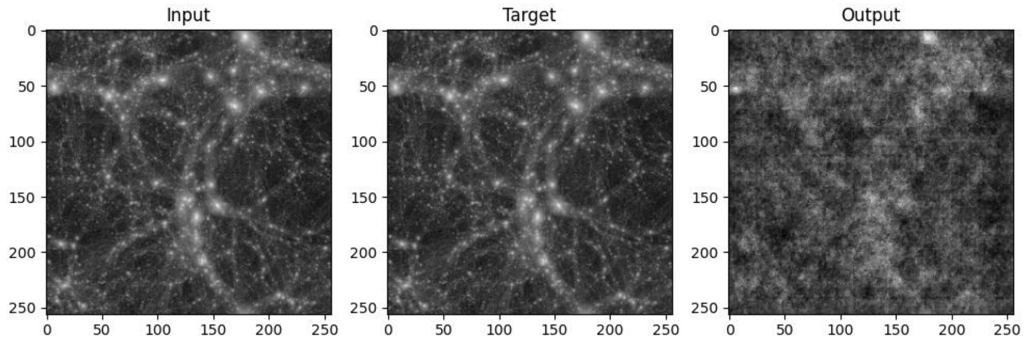

Figure 5: ResNet reconstruction: Input to Output

The V-Net model initially showed good results at 20 epochs (RMSE **0.0275**, PSNR **31.21 dB**), and after 50 epochs, its RMSE dropped to **0.0261**, highlighting the need for sufficient training time.

Table 3: V-Net Architecture Hyperparameter Tuning

Model Name	Kernel Size	Epochs	Activation Function	RMSE	PSNR
VNetmodel_1	3	20	RELU	0.0275	31.21
VNetmodel_2	3	50	RELU	0.0261	31.68

Although the network is deeper, the output is still a little blurrier, as seen in Figure 6, suggesting difficulty in capturing fine details due to insufficient training or architectural balance.

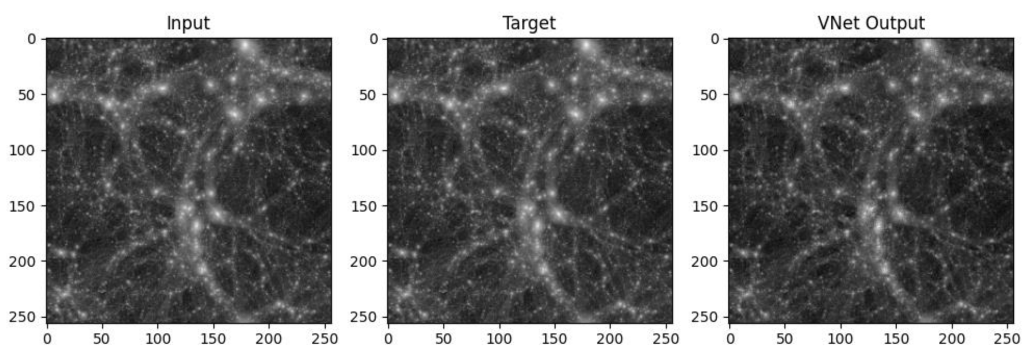

Figure 6: V-Net Reconstruction: Input to Output

The best results came from the final optimized U-Net with skip connections, named Model Skip_UNet_model, trained for 50 epochs. This model drastically improved performance with an RMSE of **0.0051** and PSNR of **45.79 dB** by far the most accurate configuration. The skip connections proved vital in retaining spatial detail across layers, ensuring both high-fidelity structure reconstruction and generalization to unseen data. The output image is nearly indistinguishable from the target image, suggesting that the model captured all the details across layers accurately, highlighted in Figure 7.

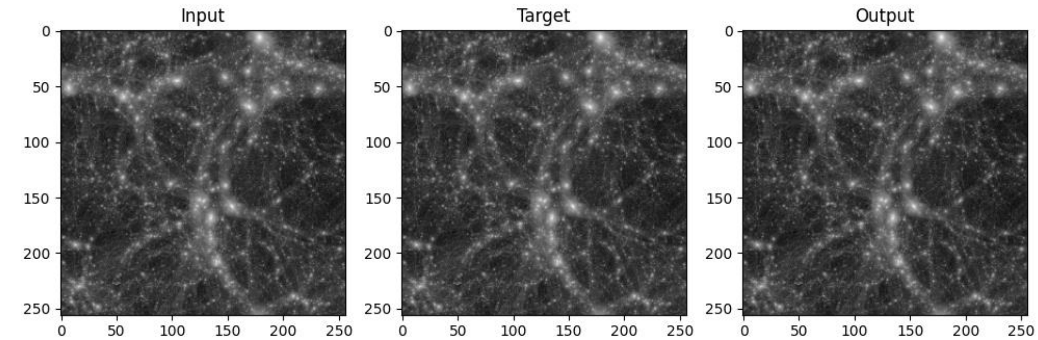

Figure 7: U-Net with skip connections: Input to Output

3.3. Analysis and Interpretation of Outcomes

The results demonstrate a clear trend: deeper architectures with better spatial information retention perform significantly better in predicting future states of cosmological simulations. The use of skip connections was especially impactful. These allowed earlier high-resolution features to be directly reused during reconstruction, thus preserving small-scale structures in the output maps.

Moreover, training for a greater number of epochs improved the model's capacity to minimize loss and generalize across batches. The drop in RMSE by over 23% and PSNR gain of nearly 2.5 dB strongly suggest that the model benefited from extended learning. This indicates that the task is complex enough to warrant significant representational depth and training time to learn predictive dynamics effectively.

Interestingly, the visual output mirrored these quantitative improvements. The target and predicted density maps were almost indistinguishable in high-density regions in the final model, whereas the baseline model often blurred or missed subtle filamentary structures. This consistency between metrics and visual inspection provides additional confidence in the model's performance.

3.4. Assessment of Robustness and Scientific Validity

From a reliability standpoint, the model consistently yielded low MSE and high PSNR across multiple test batches, indicating stable and accurate performance. Evaluations were conducted using torch's no-grad mode to prevent data leakage. The data pipeline was standardized, applying uniform binning and logarithmic normalization across all datasets, minimizing preprocessing bias. The task is scientifically valid, aligning with established practices in cosmological machine learning.

However, the replication of snapshots may limit generalization, and normalization could obscure low-density details. Additionally, reliance on a fixed 256×256 resolution suggests that exploring multi-scale architectures may enhance both computational efficiency and predictive performance.

4. Discussion

4.1. Insights from the Interpolated Outputs

From a reliability perspective, the model demonstrated consistent performance across multiple test batches, producing outputs with low mean squared error and high PSNR, which reflects both accuracy and stability. All evaluations were conducted in torch's no-grad mode, ensuring no data leakage or gradient interference during inference. The data preparation pipeline was uniformly applied, using fixed binning and logarithmic normalization across training, validation, and test sets. This consistency minimized preprocessing bias and ensured that observed improvements were attributable to model design rather than data handling.

In terms of validity, the representation of N-body simulations as 2D density maps is well-established in cosmological machine learning. The next-frame prediction task aligns with physical principles of large-scale structure evolution. However, limitations remain, reusing the same snapshot may reduce generalization, normalization may obscure low-density variations, and fixed-resolution inputs could limit scalability and accuracy.

4.2. Alignment and Contrast with Prior Work

This work builds upon and contrasts with several recent studies applying deep learning to cosmological simulations. Most directly, the approach by Chen et al. (2020) served as the foundation for my methodology. Like their model, mine frames interpolation as a supervised learning problem using CNNs. However, my implementation incorporates several architectural modifications aimed at improving generalization and accuracy. The results affirm the viability of CNNs for interpolating simulation data, echoing Chen et al. (2020) findings.

Ulibarrena et al. (2024) use Artificial Neural Networks, including Hamiltonian Neural Networks (HNNs), to approximate the most computationally expensive parts of the simulations specifically, the gravitational interactions between particles. Their method integrates these predictions into a hybrid Symplicit integrator to maintain physical accuracy. In contrast, I use convolutional neural network (CNN), specifically a U-Net architecture, to learn the temporal evolution of cosmological structures from simulation snapshots. By converting 3D particle data into 2D density maps, my model learns to predict intermediate states between snapshots, reducing the need for frequent outputs and large storage requirements. While both approaches apply deep learning to N-body problems, their focus is different: Ulibarrena et al. (2024) aim to speed up simulation runtimes, whereas I aim to reduce the frequency of snapshot generation by enabling accurate interpolation between them.

In contrast, Lanzieri et al. (2022) presents a method to enhance the accuracy of Particle-Mesh (PM) N-body simulations by integrating a neural network-based correction into the simulation's differential equations. This approach addresses the PM method's limitation in resolving small-scale structures due to its approximation of short-range gravitational interactions. The authors introduce an effective force, parameterized by a Fourier-space neural network, to compensate for these approximations. Their results demonstrate improved matter power spectrum accuracy and robustness across different simulation settings compared to the Potential Gradient Descent (PGD) scheme.

Similarly, Chacón et al. (2023) uses machine learning to classify dark matter particles as either belonging to halos or not, based on initial condition features. They also reconstruct the Halo Mass Function (HMF) using supervised learning algorithms such as Random Forests and Neural Networks. Their approach is centered on particle-level classification and statistical reconstruction. In contrast, my work focuses on interpolating simulation snapshots using convolutional neural networks (CNNs), specifically employing a U-Net architecture to learn from and predict 2D density maps. While both projects aim to reduce the computational cost associated with N-body simulations, I concentrate on the temporal aspect reducing the number of stored outputs over time rather than on particle classification or abundance modeling. Additionally, my CNN operates on field-level representations rather than individual particle data, which gives my approach a different scale and application focus.

Perraudin et al. (2019) tackles the challenge of generating full 3D N-body simulations using scalable Generative Adversarial Networks (GANs). Their solution involves a multi-scale approach and patch-wise training to handle the memory demands of high-resolution 3D data. Where their GAN-based method aims to fully reproduce simulations, my approach predicts intermediate states using temporally adjacent snapshots as input, making it more comparable to temporal inpainting. I also avoid some of the issues they mention, such as the difficulty GANs have with capturing rare features or achieving physical accuracy, by working directly with known simulation outputs and focusing on interpolation rather than generation.

Jamieson et al. (2023) presents a sophisticated emulator that predicts nonlinear particle displacements and velocities based on linear initial conditions. Their model incorporates cosmology dependence through style parameters in the neural network, allowing it to generalize across a wide range of ΛCDM cosmologies. While their model aims to replace the simulation pipeline entirely, mine is designed to reduce the number of snapshots needed by predicting intermediate outputs. My method is simpler, less computationally demanding, and specifically intended to address the storage and data handling challenges in running and analyzing large-scale simulations like those in the CAMELS dataset.

4.3. Broader Impact and Scientific Relevance

The ability to interpolate between simulation snapshots with CNNs has significant practical and theoretical implications. From a practical standpoint, N-body simulations are computationally expensive and data-intensive, requiring substantial storage and processing resources. My results demonstrate that deep learning models can be used to reduce the frequency of stored snapshots without sacrificing crucial structural information, offering a promising solution to the storage problem in cosmological simulations.

Theoretically, my findings contribute to the ongoing discussion about the applicability of deep learning to scientific data. By focusing on interpolation rather than generation or compression, this work introduces a novel use case that complements existing efforts and shows how CNNs can be adapted for temporally continuous modeling tasks. Methodologically, it highlights the importance of architectural choices like filter sizes, regularization, and skip connections in determining the fidelity of deep learning reconstructions of complex physical systems.

4.4. Methodological Reflections and Limitations

While the results validate the core premise of CNN-based interpolation, several limitations must be acknowledged. The model's reliance on convolutional layers with fixed kernel sizes makes it sensitive to the scale of input features, reducing its adaptability to different redshifts and cosmological conditions. This is particularly problematic when the matter of distribution evolves significantly over time, affecting the model's performance.

Another limitation lies in the loss of fine-grained details during interpolation. Although the model effectively captures large-scale structures, it occasionally fails to reconstruct small-scale features accurately.

Additionally, my model is purely data-driven and does not incorporate any physical constraints or prior knowledge of gravitational interactions. This omission limits its capacity to produce physically consistent outputs, particularly when extrapolating to unseen conditions.

4.5. Alternative Viewpoints and Data Considerations

An alternative interpretation of the results is that the loss of fine detail and difficulty generalizing across redshifts may not solely be a limitation of CNNs, but also a result of the way the input data is preprocessed. Using 2D projections or voxelizations of 3D particle data inherently leads to some loss of information. Future attempts to model the full 3D particle distribution directly using architecture designed for point clouds or graphs might yield improved fidelity.

Additionally, the choice to treat interpolation as a supervised learning task between fixed redshifts may oversimplify the problem. Temporal evolution in cosmological simulations is governed by complex, continuous dynamics, which may be better captured by models trained in a recurrent or sequence-aware framework. The temporal gap between training and test data could also influence the generalization performance more significantly than expected.

4.6. Future Directions and Enhancements

Several avenues exist for extending this research. First, experimenting with hybrid architecture that combines CNNs with attention mechanisms as used in transformer models could improve the model's ability to capture long-range dependencies and multi-scale features. This would address many of the shortcomings observed in modeling large temporal gaps or structurally diverse snapshots.

Second, integrating adversarial training frameworks like GANs could encourage the generation of more realistic and visually coherent interpolations.

Third, incorporating dimensionality reduction techniques, such as autoencoders, may allow the model to work in a more compact latent space, improving both efficiency and generalization.

Fourth, exploring alternative input representations such as graph-based particle interactions could help retain more physical realism. This would involve moving away from voxelized inputs toward data structures that better reflect the physical nature of the simulation.

A promising direction for future enhancement of this model involves incorporating **physics-informed loss functions** that explicitly embed physical laws into the learning process. While the current framework is entirely data-driven, cosmological simulations are governed by well-established physical principles, such as mass conservation and gravitational dynamics. By integrating these constraints into the loss function, the model can be guided not only by data accuracy but also by scientific validity. One such composite loss function can be defined as:

$$L_{total} = L_{MSE} + \lambda \cdot L_{phys}$$

In this formulation, L_{MSE} is the standard mean squared error loss used to minimize the difference between predicted and actual density maps, while L_{phys} represents a custom term designed to enforce physical consistency (e.g., conservation of mass, continuity of gravitational potential, or adherence to cosmological evolution equations). For conservation of mass, this term would be reduced to $\boldsymbol{\nabla} \cdot \vec{\boldsymbol{v}}$ captures violations of mass conservation and the hyperparameter λ controls the relative influence of this physics-based term. The whole equation would then be:

$$L_{total} = L_{MSE} + \lambda \cdot ||\boldsymbol{\nabla} \cdot \vec{\boldsymbol{v}}||$$

By blending data-driven objectives with physics-based priors, such a model could not only achieve higher fidelity reconstructions but also gain greater interpretability and scientific trustworthiness. This formulation provides a principled way to align the learning objective with the underlying physics of large-scale structure formation, encouraging the model to generate temporally interpolated outputs that are not only visually and statistically accurate, but also physically plausible. Incorporating such physics-informed regularization could significantly enhance the scientific trustworthiness of deep learning-based simulation models.

5. Conclusion

This study demonstrated that convolutional neural networks, particularly optimized U-Net architectures, can effectively interpolate between N-body simulation snapshots. The approach achieved high accuracy in reconstructing intermediate cosmological structures, evidenced by strong performance in both quantitative metrics and visual assessments. The project validates the use of deep learning to reduce snapshot frequency, providing a practical solution to the storage bottlenecks in large-scale simulations. While the current model operates on 2D projections and is limited by its fixed convolutional structure, it lays the groundwork for future research incorporating 3D data, attention mechanisms, and physics-informed architecture. Overall, this work advances the intersection of machine learning and cosmology, offering a robust framework for efficient simulation analysis.

6. References

Chacón, J., Gómez-Vargas, I., Menchaca Méndez, R., & Vázquez, J. A. (2023). Analysis of dark matter halo structure formation in N-body simulations with machine learning. *Physical Review D*, *107*(12), 123515.

Chen, C., Li, Y., Villaescusa-Navarro, F., Ho, S., & Pullen, A. (2020). Learning the Evolution of the Universe in N-body Simulations. *arXiv preprint arXiv:2012.05472*.

Ciresan, D., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). *Flexible, High Performance Convolutional Neural Networks for Image Classification.* https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-210

Jamieson, D., Li, Y., de Oliveira, R. A., Villaescusa-Navarro, F., Ho, S., & Spergel, D. N. (2023). Field-level neural network emulator for cosmological N-body simulations. *The Astrophysical Journal*, *952*(2), 145.

Lanzieri, D., Lanusse, F., & Starck, J.-L. (2022). Hybrid physical-neural odes for fast n-body simulations. *arXiv preprint arXiv:2207.05509*.

Lucie-Smith, L., Peiris, H. V., Pontzen, A., & Lochner, M. (2018). Machine learning cosmological structure formation. *Monthly Notices of the Royal Astronomical Society*, *479*(3), 3405-3414.

Pahwa, R., Arivazhagan, M. G., Garg, A., Krishnamoorthy, S., Saxena, R., & Choudhary, S. (2019). Data-driven compression of convolutional neural networks. *arXiv preprint arXiv:1911.12740*.

Perraudin, N., Srivastava, A., Lucchi, A., Kacprzak, T., Hofmann, T., & Réfrégier, A. (2019). Cosmological N-body simulations: a challenge for scalable generative models. *Computational Astrophysics and Cosmology*, *6*, 1-17.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18,

Sobirov, I., Saeed, N., & Yaqub, M. (2023). Super Images-A New 2D Perspective on 3D Medical Imaging Analysis. Annual Conference on Medical Image Understanding and Analysis,

Ulibarrena, V. S., Horn, P., Zwart, S. P., Sellentin, E., Koren, B., & Cai, M. X. (2024). A hybrid approach for solving the gravitational N-body problem with Artificial Neural Networks. *Journal of Computational Physics*, *496*, 112596.