Efficient Physically Aware Generation of Dark Matter Distributions

Mattéo Santini
matteo.santini@epfl.ch

Louis James Vasseur
louis.vasseur@epfl.ch

Julien Barozet-Golbery
julien.barozet-golbery@epfl.ch

December 2025

1 Introduction

Approximately 85 percent of the matter currently observed in the universe is dark matter, matter which we cannot observe by light yet causes gravitational anomalies (such as in motion and mass distribution shifts) unexplainable with just ordinary mass alone. It has been theorized that they are composed of massive particles which interact weakly with the physics Standard Model and interacts more strongly with itself, but such models are yet to be proven and the exact composition of dark matter remains an open question.

As physicists aim to understand more of the universe, one key method to reliably study dark matter has been found: the study of galaxy clusters. These clusters are extremely heavy in mass (about $10^{14} - 10^{15}$ times the mass of our sun), most of which is dark matter. This immense mass curves spacetime, bending light from background galaxies and creating distortions, arcs, and multiple views of the same galaxy within an image [MKR10]. Since dark matter dominates the cluster mass, we can use these distortions to infer the total mass distribution along the line of sight, effectively using galaxy clusters as cosmic lenses to map dark matter.

2 Problem Statement and Objective

Recent studies on the use of neural networks (NN) to classify physical properties of dark matter are promising [Har24]. However, training these networks relies on a lot of data that is currently simulated using large numerical simulations (2×1024^3 particles) that are computationally expensive, so gathering enough data to train neural networks is challenging. Our task is to assess whether generative methods can learn to produce realistic galaxy cluster mass maps, bypassing the need for expensive new particle simulations.

For this project, our dataset was comprised of the results of numerical simulations using different techniques. Given the limited time and resources we only made use of the data generated by the BAHAMAS simulations [MSBLB16] (left as-is, no preprocessing was needed). The data consists of 14400 annotated "images" with a sample shown in Figure 12, and features ranging from total mass in the cluster, ellipticity of the brightest galaxy in cluster and the self-interaction of dark matter per unit mass, denoted σ_{DM}/m, that takes values in $\{0.01, 0.1, 0.3, 1.0\}$ cm^2/g. Although the cross-section of dark matter is only represented as 4 distinct values (due to simulation's limitations to produce more diverse values), we chose not to treat that feature as categorical since it is a continuous physical quantity and that we wanted our models to be capable to generalize more easily if more data was provided.

Therefore our inference pipeline can be seen in Figure 1 and goes as follows, given a pair of mass and σ_{DM}/m, in order to get the conditioning embeddings (which we need to compress the information in the data optimally in order to condition a generative model) one has the two following options:

- Find the image that has the closest features from the queried ones and extract the embeddings through a nanoGPT-based model finetuned on galaxy density map images: AstroPT [SRAHC24].

- Sample the embeddings generated by the Flow-Matching model using the (mass, σ_{DM}/m) pair.

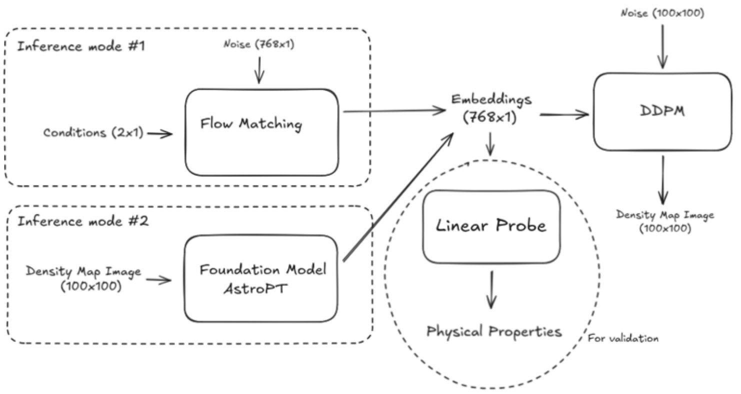


Figure 1: Our Inference pipeline

Finally, we pass the embeddings as conditions for the Diffusion Model to obtain a generated image.

3 AstroPT Finetuning

AstroPT was a natural choice for our use case. It is an autoregressive transformer pretrained on 8.6M galaxy images from the Dark Energy Spectroscopic Instrument (DESI) Legacy Survey. It is one of the main foundation models in astrophysics due to its powerful ability to adapt to downstream tasks (no excessive fine-tuning necessary) and its embeddings being capable to efficiently capture underlying physical processes (e.g. ellipticity, mass, redshift...).

Using AstroPT, our key idea was to leverage its capacity and attempt to fine-tune its embeddings to efficiently learn two specific features: mass and σ_{DM}/m. Our finetuning aims to create a manifold which can efficiently encode both of these features, with the aim of creating a strong embedding to serve as a condition for the downstream diffusion model. Our method fully unfreezes AstroPT and learns with a dual-task objective:

- A linear regression head to predict multiple target properties (mass, label, Brightest Cluster Galaxy shape, ellipticity components, stellar concentration all defined in Appendix A), implementing a mean square error (MSE) loss for stable learning across all continuous targets

- A supervised contrastive loss (supervised InfoNCE [KTW$^+$20]) on the σ_{DM}/m label, which unlike regression losses allows to optimize the geometry of the latent space (samples with the same σ_{DM}/m get pulled together, different classes get pushed apart) so the downstream diffusion model receives well-structured conditioning vectors. The contrastive weight λ_c is linearly interpolated from zero to its target value λ_c^* over the first N_w epochs for training stability:

$$\lambda_c(e) = \lambda_c^* \cdot \min\left(1, \frac{e}{N_w}\right) \tag{1}$$

where e is the current epoch.

These two objectives lead to a strong embedding space capable of mapping images to their associated mass/label pairs in an effective way for downstream image generative modeling.

4 Flow Matching for Embeddings Generation

Now that we have a good embedding space obtained through the finetuning of AstroPT, we need to generate new embeddings in order to condition our diffusion model. However, using AstroPT is not viable since we would need an input image (when our task is to generate the image from just the (mass, σ_{DM}/m) pair. Therefore, we have turned to a generative model,conditional flow matching [LCBH+23], in order to generate embeddings given a mass and σ_{DM}/m.

Similar to the diffusion process shown in Section 5, this method aims to create a bridge between a known analytical distribution ($x_0 \sim \mathcal{N}(0, I)$ in our case) and a target distribution ($x_1 \sim q(x_1)$, our distribution in the embeddings manifold). Our backbone for this is based on a simple multi-layer perceptron (MLP) as seen in Sub-Figure 2b. This model is used to predict a conditioned (mass and σ_{DM}/m) time-dependent velocity field to displace our noise sample to a valid point that represents a generated embedding. More concretely, given k scalar conditions (or features), and a distribution of dimension d, our neural vector field $v_\theta : [0,1] \times \mathbb{R}^k \times \mathbb{R}^d \to \mathbb{R}^d$ generates a flow $\phi : [0,1] \times \mathbb{R}^k \times \mathbb{R}^d \to \mathbb{R}^d$ through the following differential equation:

$$\begin{cases} \frac{d}{dt}\phi_t(x) & = v_t(\phi_t(x)) \\ \phi_0(x) & = x \end{cases} \tag{2}$$

Solving this ODE from a sampled point x_0 leads it to a point x_1 in our target distribution (here the embedding manifold). In order to train our neural velocity field, we made use of the conditional flow matching loss [LCBH+23], that allows the creation of an analytical velocity field $u_t(x|x_1)$ trained from our training samples. This way, we minimize the following expectation on $t \sim Unif(0,1)$, $x_1 \sim q(x_1)$, $x \sim p_t(x|x_1)$:

$$\mathbb{E}_{t,q(x_1),p_t(x|x_1)}||v_\theta(x) - u_t(x|x_1)||^2, \tag{3}$$

and

$$p_t(x|x_1) = \mathcal{N}(x|\mu_t(x_1), \sigma_t(x_1)^2 I). \tag{4}$$

From here, multiple types of analytical paths are possible depending on the values of $\mu_t(x_1)$ and $\sigma_t(x_1)$. We used a PyTorch library `torchCFM` [AT25] that grouped a few implementations for those paths, and these are described in the Appendix B.

5 Diffusion Model for Density Map

5.1 Denoising Diffusion Probabilistic Model

In this section we will describe the Denoising Diffusion Probabilistic Model (DDPM) framework [HJA20] that we use to generate the density maps.

The DDPM framework constrains us to use the Gaussian distribution as the latent distribution. We then want to bridge this distribution with the distribution of our density map. In order to model this bridge, we consider the following noising process:

$$\mathbf{x}_t(\mathbf{x}_0, \epsilon) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \tag{5}$$

With $\epsilon \sim \mathcal{N}(0, \mathbb{I})$, $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ the input data sample and $\bar{\alpha}_t$ the cumulative product of the variance schedule.

Now, we define the reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \sim \mathcal{N}(\mathbf{x}_{t-1}, \mu_\theta(\mathbf{x}_t, t), \sigma_t^2\mathbb{I})$, where the challenge is to predict the mean without leveraging its tractable formula that relies on the conditioning on $\mathbf{x}_0$, not available during sampling.

We can derive the following expression for $\mu_\theta(\mathbf{x}_t, t)$:

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) \tag{6}$$

with $\beta_t = 1 - \alpha_t$ and $\epsilon_\theta(\mathbf{x}_t, t)$ the noise our backbone model will predict.

Finally, we will train our diffusion model based on the following loss :

$$\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \tag{7}$$

With all derivations completed, we obtain the following formula for sampling :

$$\begin{aligned} \mathbf{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\mu_\theta(\mathbf{x}_t, t) + \sigma_t\mathbf{z} \\ &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t\mathbf{z} \end{aligned} \tag{8}$$

with $\mathbf{z} \sim \mathcal{N}(0, \mathbb{I})$

5.2 Diffusion Transformer

The Diffusion Transformer (DiT) [PX23] is the backbone of this Diffusion Model. The choice of the transformer in this DDPM framework imposed itself because as mentionned in this paper [PX23], it improves the baseline results obtained by U-NET networks for almost every use cases and set up. You can find in figure 2a an illustration of the DiT. We adapted the original framework to our case hence we only predict the noise and the dimensions are different.

For the specifications of the architecture, we implemented 12 DiT Block with 8 heads per block. We find this to be the optimal choice because smaller-sized model aren't complex enough for the task at hand (density map patterns do not appear during sampling) and bigger models either overfit or don't fit inside the memory limit imposed by SCITAS. The conditioning vectors are integrated into the model via Adaptive LayerNorm during each forward pass of each block as described in the figure 2a.

6 Results

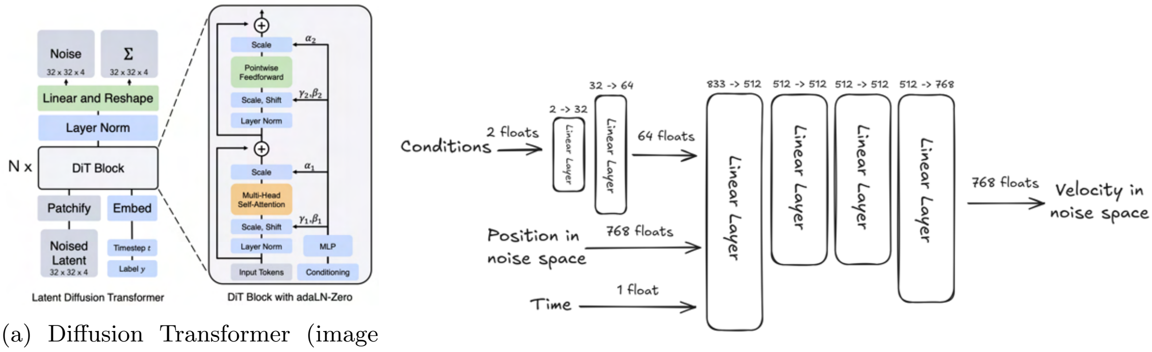
The goal of this section is to present the results of our components and pipeline, and compare to existing baselines. Please note that this work is based on new findings so no such baseline exists for the DDPM part. We will, however, show our finetuning's clear improvements over AstroPT's non-finetuned baseline.

6.1 AstroPT Finetune

Testing the dual-task objective finetuned model (cf Section 3) using linear probes yields the results in Figure 3. The finetuned model shows tighter uncertainty bounds and better class separation for $\sigma_{1.0}$ (0.85 vs 0.69). Remaining overlap between lower-σ classes likely reflects that the self-interaction cross-section is too small to meaningfully impact galaxy cluster morphology.

6.2 Flow Matching

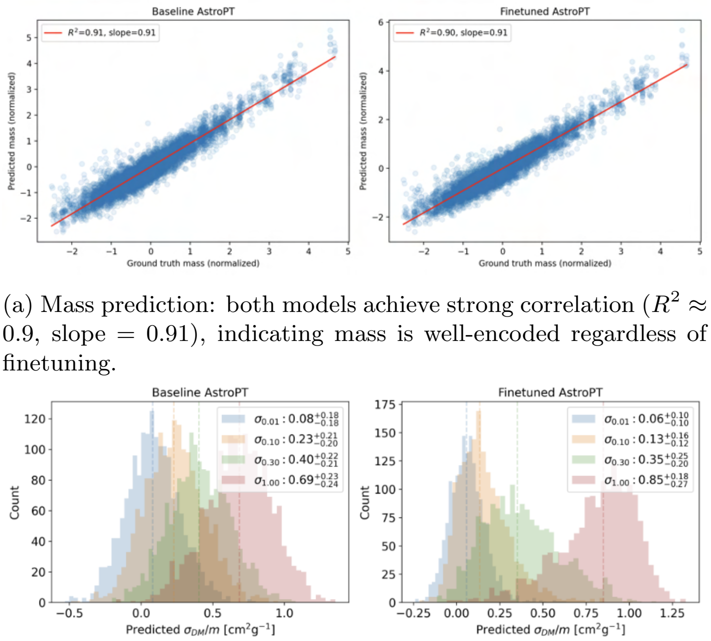
In order to test the generated embeddings, we took the embeddings of the original image dataset passed through AstroPT and computed a linear regression and UMAP projection. We



(a) Diffusion Transformer (image from [PX23])

(b) Flow Matching

Figure 2: Architectures of our Generative Models' Backbones



(a) Mass prediction: both models achieve strong correlation ($R^2 \approx 0.9$, slope $= 0.91$), indicating mass is well-encoded regardless of finetuning.

(b) σ_{DM}/m prediction: finetuning reduces uncertainty bounds and improves $\sigma_{1.0}$ recovery (0.85 vs 0.69). See legend for median ± 16/84% quantiles.

Figure 3: AstroPT finetuning results: baseline vs finetuned embeddings evaluated via linear probes.

then proceeded to generate new embeddings through the flow-matching model and made predictions using the previously fitted linear regression. The results displayed in figure 4 and table 1 shows the metrics resulting from the generated embeddings. We see that, overall, the metrics that concern the mass seem to fit. However, when it comes to the σ_{DM}/m predictions, the numbers increase when using flow matching generation. This might indicate that our simple MLP does not capture the full extent of AstroPT's manifold and that the generated vectors are too "predictable".

	R2 score	MSE
FM Mass	0.87	0.00567
AstroPT Mass	0.89	0.0991
FM σ_{DM}/m	0.73	0.0393
AstroPT σ_{DM}/m	0.61	0.29

Table 1: Results using a linear regression on the AstroPT embeddings

This underfitting could actually be an explanation to the model's sensitivity to the hyper-parameter $\sigma_{\min}$ and the type of path that is chosen. After an extensive grid-search over the three paths defined in (B), as well as $\sigma_{\min}$ values ranging in $[0.001, 1.5]$, it seems that the best results are obtained with the variance preserving path (11) with a $\sigma_{\min} = 1.0$. A few examples of generation using this hyper-parameter can be seen in Appendix C.

6.3 Final Pipeline

As per figure 1, our final goal is to test how well our downstream DDPM works in conjunction with flow matching (Inference Mode 1) and AstroPT finetuned (Inference Mode 2). We essentially aim to show that Inference Mode 1 is capable of generating realistic images from just mass and cross-section using its learned vector field in flow matching and Inference Mode 2 is capable of reconstructing images from AstroPT using only the mass and cross-section values. Since both approaches have shown strong promise in testing, we will train, in each approach, the DDPM conditioned on the embeddings and plot the results using appropriate metrics.

6.3.1 Inference Mode 1

Inference Mode 1 is a purely generative task, where we intend to turn mass and self-interaction cross-section pairs into realistic images via our flow-model generated vector field. Since flow matching embeddings are stochastic samples from a learned distribution (not deterministic encodings of specific images), we evaluate statistical consistency (asking if generated images have similar distributional properties (power spectrum, flux distribution) to real images) and visual quality (asking if generated images look realistic and correlate appropriately to conditioning inputs).

The generated results (cf Appendix F figure 13) are coherent and not excessively noisy.

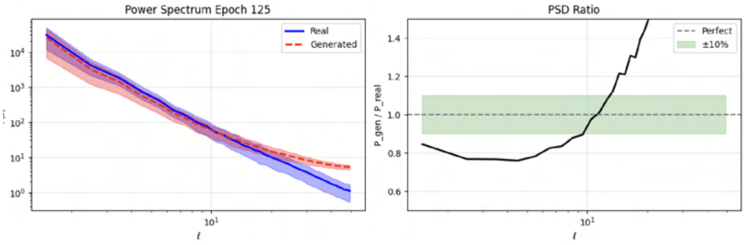
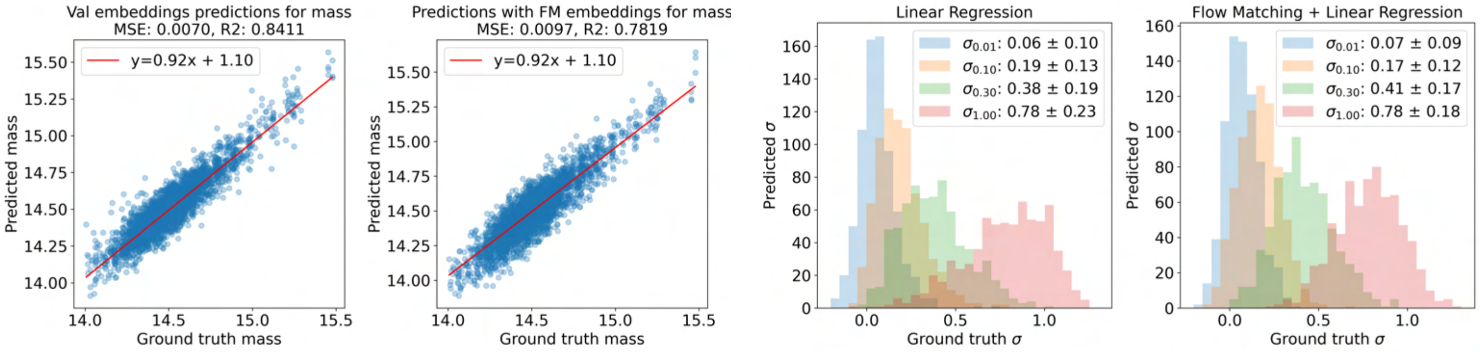


Figure 5: Power spectrum comparison. Left: Mean PSD with uncertainty bands. Right: PSD ratio (1.0 = perfect match).



(a) Differences in mass predictions

(b) Differences in σ_{DM}/m predictions

Figure 4: Prediction differences between validation features/embeddings and generated, **for both subplots:** On the left: the prediction quality with the original embeddings. On the right: the predictions made using the generated embeddings

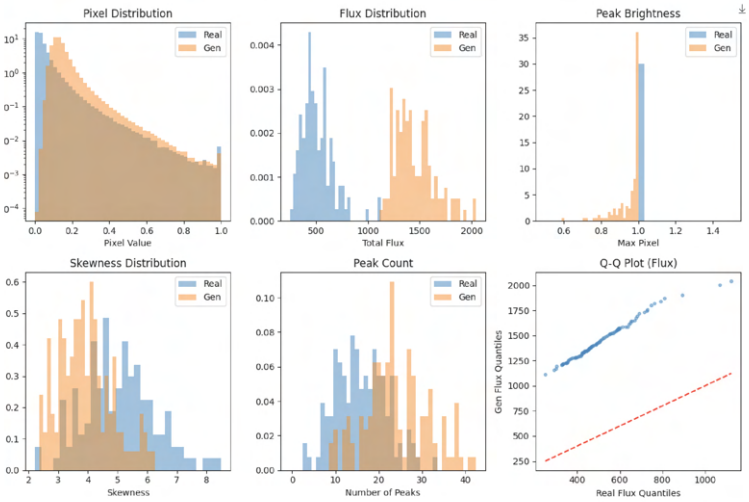


Figure 6: Distribution comparisons between real (blue) and generated (orange) images.

The power spectrum comparison 5 shows a strong link between real and generated images. At middle frequencies, the PSD ratio is within ±10% of being equal. We observe slight discrepancies in power at large scales (ratio ≈ 0.8) and at small scales (ratio > 1.2). The distributional analysis 6 shows strong overlap for pixel values, skewness, and peak counts, confirming the model captures the non-Gaussian structure of galaxy clusters. However, the flux distribution reveals that generated images are systematically brighter than real images, and this is confirmed by the Q-Q plot's deviation above the diagonal.

We summarize that our method shows strong merit but still shows room for improvement (this result was anticipated since training the flow matching is only given mass and σ/m, it can't encode other physical properties like ellipticity accurately leading to a lack of geometric accuracy).

6.3.2 Inference Mode 2

The reconstruction task (encoding an existing image through AstroPT and decoding it via the DDPM) is, as expected, more straightforward than Inference Mode 1's pure generation from physical parameters. Our DDPM does a strong job at reconstruction, where the generated images have reasonable distributions and are close to the original images (cf Appendix F, Figure 14).

We use two key metrics to validate this intuition and assess fidelity in reconstruction (bearing in mind that stochasticity in the diffusion model means the reconstruction cannot be perfect):

- $r(\ell)$: Cross-correlation coefficient measuring paired reconstruction fidelity at each angular scale ℓ. Values near 1 indicate perfect reconstruction: lower ℓ monitors general attributes (overall brightness, etc.) whereas higher ℓ monitors more precise metrics such as noise and sharp edges.
- Power Spectral Density, assessing how much power of the image is contained at different spectral frequencies, telling us which scales dominate the image. More specifically, we compute the log of the generated image over the real one: if they are near 0 everywhere, this is a strong indicator that both images are near-identical at all frequencies.

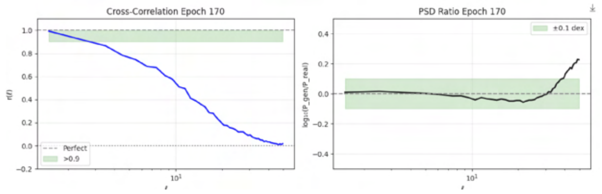


Figure 7: Cross-correlation and PSD spectra

Figure 7 demonstrates that at low ℓ (large scales), $r(\ell)$ spans between 0.8 and 1.0, showing almost perfect reconstruction of the image's large-scale structure. The model also shows promising metrics at mid-level ℓ ($r(\ell)$ at approximately 0.3 to 0.6), demonstrating capability to reconstruct finer details. As expected, the model struggles with extremely fine-grained details at high ℓ.The PSD spectrum is near 0 everywhere, which confirms our visual intuition that the model is capable of retranscribing the image near-perfectly as shown by their very similar spectra.

For reproducibility, find hyperparameters we used (found via grid search over a range of reasonable/admitted values for algorithms, where best metrics were achieved in similar setups):

	AstroPT	**FM**	**DDPM-FM**	**DDPM-Emb**
LR / Batch / Epochs	10^{-5}/64/60	10^{-4}/full/3k	3×10^{-4}/32/250	3×10^{-4}/32/300
Scheduler	Cosine	–	Cosine	Cosine
Contrastive (λ, τ, warmup)	0.1, 0.1, 5	–	–	–
FM (σ, OT)	–	1, Variance Preserving 11	–	–
DDPM (T, CFG, EMA)	–	–	1k, 0.1, .9999	1k, 0.1, .9999

Note that the training framework uses AdamW optimizer.

7 Conclusion and Future Work

For future work, while we believe our general approach and architecture is a good fit for the problem, all individual components (finetune, flow and diffusion models) can be improved upon. Due to the lack of time we were not able to test all combinations possible to find the best results. Furthermore, while it wasn't in the scope of the project, this report can be further extended and serve as support for anyone who wants to pursue generation of both X-ray measurements and distribution of stellar mass.

8 Ethical Risks

Using the Digital Ethics Canvas, we evaluated risks across the five categories: Welfare, Fairness, Autonomy, Privacy, and Sustainability. Privacy is not applicable (our simulation data contains no personal information). Sustainability is a net positive: our work aims to reduce expensive computational costs of simulations (at the slight cost of using SCITAS resources to train). We identified physical implausibility (Welfare) as the primary ethical risk.

Risk Description. Generative models can produce outputs that appear visually realistic but violate physical laws or contain structures absent in real simulations.

- Stakeholders impacted: Astrophysics researchers using generated samples for downstream analysis.
- Negative impact: Unlike hallucinations in NLP, physically implausible results in galaxy cluster maps are far harder to detect without domain expertise. Such mistakes can potentially lead to incorrect scientific conclusions about dark matter properties.
- Significance: incorrect scientific conclusions could negatively impact research efforts and likelihood is medium if Inference Mode 2 is used while it is high if Inference Mode 1 (less accurate) is used. Especially on Inference Mode 2, our validation shows reasonable fidelity, but edge cases can still happen.

Risk Evaluation. We evaluated this risk quantitatively by comparing generated images against real simulations using power spectral density (PSD) analysis and distributional metrics (pixel distributions, flux, skewness, peak counts). Figures 5-7 demonstrate that, especially for the Inference Mode 2 scenario, generated images match real images within ±10% at mid-frequencies, though slight discrepancies remain at extreme scales. Inference mode 1 is more at risk of this, and caution should be taken if stakeholders use Inference Mode 1 as-is (it still needs to be improved).

Mitigation. We addressed this risk by: providing comprehensive validation metrics alongside all results, enabling users to assess physical plausibility, documenting the training distribution bounds ($\sigma_{DM}/m \in \{0.01, 0.1, 0.3, 1.0\}$ cm^2/g) and recommending users remain within them, and comparing algorithm implementations against original research papers to ensure our implementations were accurate and correct. Ultimately some risk of implausible outputs remains unavoidable as our models are stochastic (as is the cas with all ML applications).

We believe we have dealt with these risks to the best of our ability to ensure our work is scientifically accurate, usable, and ethical.

References

[ABVE25] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2025.

[AT25] K. Fatras A. Tong. Torch cfm. `https://github.com/atong01/conditional-flow-matching`, 2025.

[Har24] David Harvey. A deep-learning algorithm to disentangle self-interacting dark matter and agn feedback models, 2024.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[KTW+20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2020.

[LCBH+23] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[MKR10] Richard Massey, Thomas Kitching, and Johan Richard. The dark matter of gravitational lensing. *Reports on Progress in Physics*, 73(8):086901, July 2010.

[MSBLB16] Ian G. McCarthy, Joop Schaye, Simeon Bird, and Amandine M. C. Le Brun. The bahamas project: calibrated hydrodynamical simulations for large-scale structure cosmology. *Monthly Notices of the Royal Astronomical Society*, 465(3):2936–2965, October 2016.

[PX23] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023.

[SRAHC24] Michael J. Smith, Ryan J. Roberts, Eirini Angeloudi, and Marc Huertas-Company. Astropt: Scaling large observation models for astronomy, 2024.

[TFM+24] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024.

A Definitions

- Mass: The total mass of the cluster, typically dominated by dark matter ($\sim$ 85%), with the remainder in hot intracluster gas and stellar mass.
- σ_{DM}/m: The dark matter self-interaction cross-section per unit mass, typically expressed in cm^2/g. This parametrizes how strongly dark matter particles scatter off each other.
  - σ_{DM} is the scattering cross-section
  - m is the dark matter particle mass
- BCG ellipticity components (e_1, e_2) — The shape of the Brightest Cluster Galaxy, decomposed into two components that encode both the elongation and orientation:
  - $e_1 = \frac{a^2-b^2}{a^2+b^2}\cos(2\theta)$
  - $e_2 = \frac{a^2-b^2}{a^2+b^2}\sin(2\theta)$

B Flow Matching paths

Training of the flow matching model was tested with 3 different analytical paths that are defined in the following:

The optimal transport conditional vector fields (cf Equation 20 from [LCBH+23]):

$$\begin{cases} \mu_t(x_1) &= t \cdot x_1 \\ \sigma_t(x_1) &= 1-(1-\sigma_{\min})t \\ u_t(x|x_1) &= \frac{x_1-(1-\sigma_{\min})x}{1-(1-\sigma_{\min})t} \end{cases} \tag{9}$$

The independent coupling path (Equ 14, 15 of [TFM+24]):

$$\begin{cases} \mu_t(x_1) &= (1-t)\cdot x_0 + t \cdot x_1 \\ \sigma_t(x_1) &= \sigma_{\min} \\ u_t(x|x_1) &= x_1 - x_0 \end{cases} \tag{10}$$

As well as the variance preserving path (Equ 4.10 of [ABVE25]):

$$\begin{cases} \mu_t(x_1) & = \cos(\frac{\pi}{2}t) \cdot x_0 + \sin(\frac{\pi}{2}t) \cdot x_1 \\ \sigma_t(x_1) & = \sigma_{\min} \\ u_t(x|x_1) & = \frac{\pi}{2}\left(\cos\left(\frac{\pi}{2}t\right) \cdot x_1 - \sin\left(\frac{\pi}{2}t\right) \cdot x_0\right) \end{cases} \tag{11}$$

C Example of embedding generation

Here we give a few examples of embeddings generation using the Flow Matching model, in Figure 8, we generate 8000 embeddings for each of the four random condition pairs and make a histogram of their predictions, we can see that the margin of error is quite small and fits within the margin of error of the linear regression.

Furthermore, we also show the UMAP projection of the embeddings for these predictions in Figure 9 and we can clearly see how each side of the "U" corresponds to a different σ_{DM}/m.

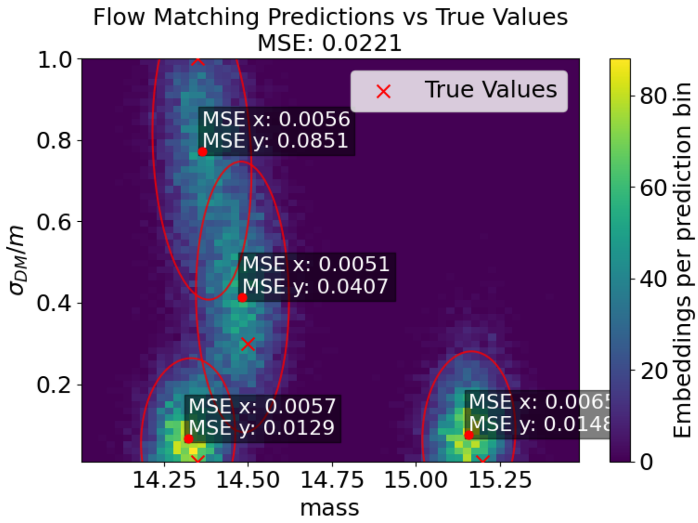


Figure 8: Predicted features histogram of generated embeddings for four different conditions

D An interesting hyper-parameter for Flow Matching

During hyper-parameter optimization of the Flow-Matching model, some interesting results occurred for high valued $\sigma_{\min} > 2$. In essence, one can interpret $\sigma_{\min}$ as the width of the probability path as the samples moves from x_0 to x_1, thus, increasing it's value leads to more overlap in the paths and thus more noise and unpredictability.

However, as can be seen in Figures 11 10, it seems like the generated embeddings collapse to the core of manifold and the prediction capabilities on the σ_{DM}/m are vastly improved. We did not, however, count these results as valid since it is apparent that the generated embeddings only span a subset of the original AstroPT embeddings and are therefore not complete.

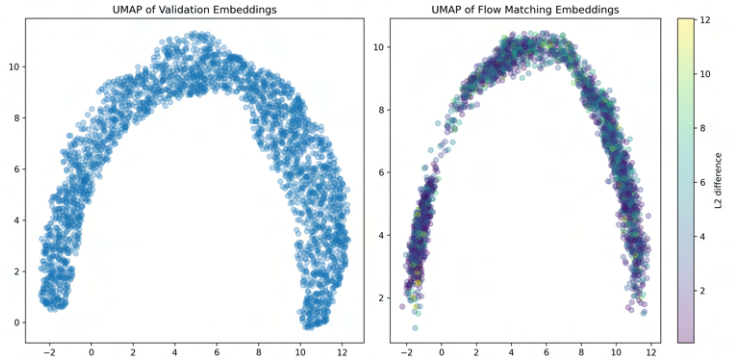
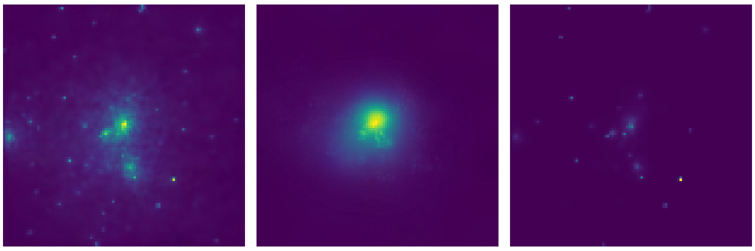


Figure 10: Generated embeddings with $\sigma_{\min} = 2.5$ and the Variance Preserving path

E Samples Examples



(a) Distribution of total mass

(b) X-ray measurements

(c) Distribution of stellar mass

Figure 12: Separate channels from a sample image of the dataset

F Generation for Inference modes 1/2

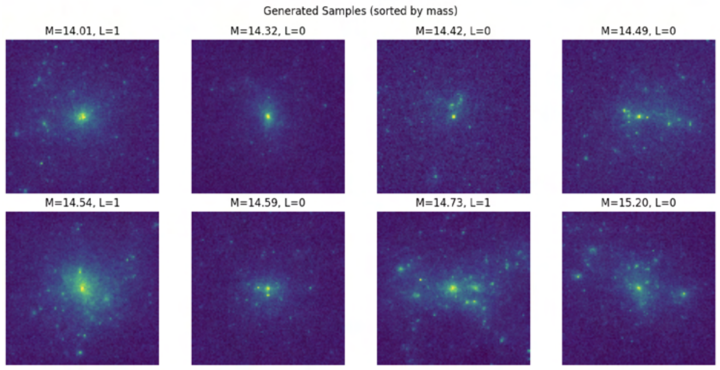


Figure 13: Images generated for (mass M in g, σ_{DM}/m in cm^2/g) pairs)

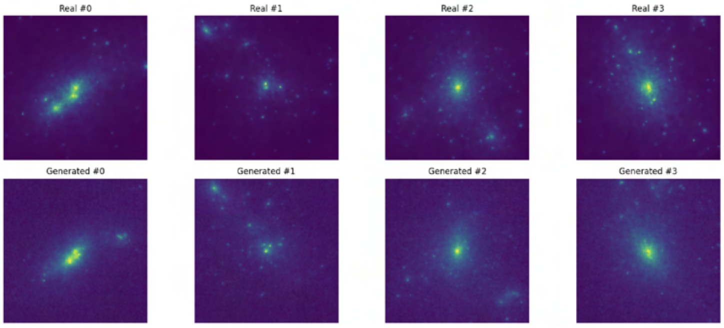


Figure 14: Real vs Generated Images from validation set

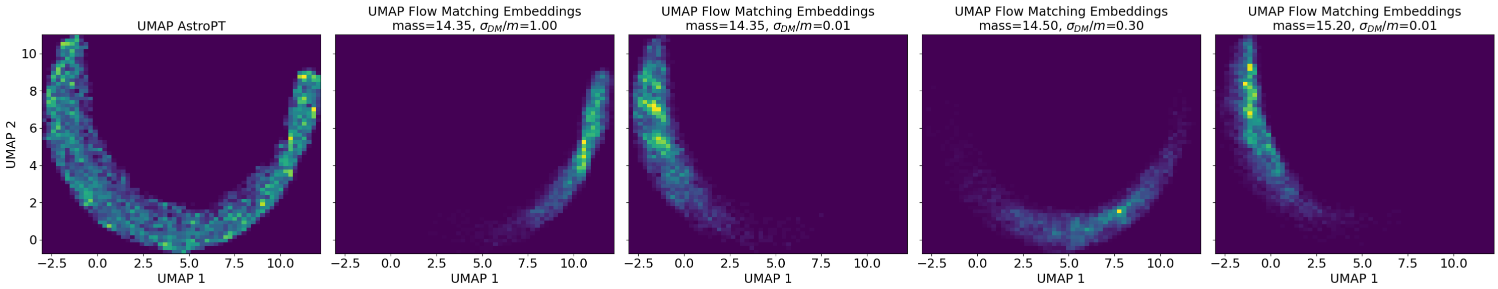
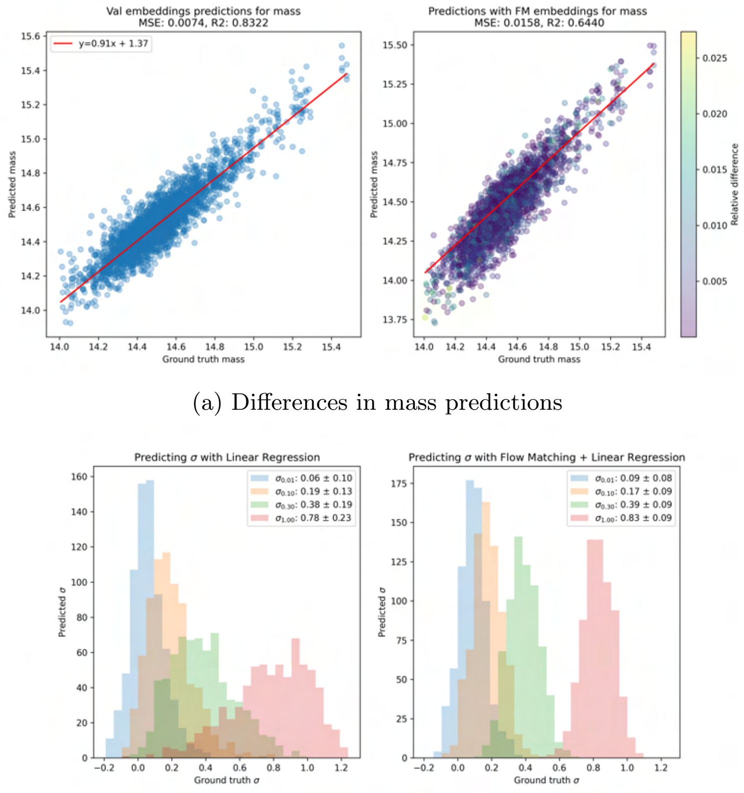


Figure 9: UMAP projections of generated embeddings with conditions, full original space on the 1st plot



(a) Differences in mass predictions

(b) Differences in σ_{DM}/m predictions

Figure 11: Predictions on generated embeddings with $\sigma_{\min} = 2.5$ and Variance Preserving path