POLYNOMIAL FLOW MATCHING

ABSTRACT. Flow Matching (FM) is a scalable framework for training continuous-time generative models by learning velocity fields that transport a simple base distribution to a complex data distribution. Linear trajectories, however, constrain the transport process and require generating all dimensions simultaneously, which can be inefficient and limiting to structured generation. We introduce Polynomial Flow Matching (PFM), an extension of FM that replaces linear transport paths with higher-order polynomial trajectories passing through learned intermediate anchor states. These curved paths enable sequential generation, with later components of the sample conditioned on earlier ones. Theoretically, we show that PFM reduces sampling complexity by decreasing per-step attention costs. Empirically, we compare the linear optimal transport (OT) path (baseline) to quadratic and quartic PFM variants on MNIST using a UNet-based flow matching architecture. Our results show that quartic PFM model achieves comparable image quality to baseline linear OT FM model (FID 65.84 vs. 65.20) while reducing sampling time compared to quadratic PFM. Although quadratic and quartic PFM does not yield empirical speedups compared to the linear baseline, PFM still has significant potential as a framework for structured and efficient generation.

1. INTRODUCTION

Continuous-time generative models have emerged as a dominant paradigm for high-quality both image (Ho et al., 2020) and text generation (Lou et al., 2024). Prior work positions these models as an alternative to autoregressive models for data without a clear causal structure. Lipman et al. (2023) introduces one such model called Flow Matching (FM). FM learns a velocity field that transports a simple base distribution (e.g. Gaussian noise) to a complex data distribution via an ordinary differential equation (ODE).

While linear transport paths correspond to OT, they also inherently restrict the generative process. Real-world data distributions are quite complex and not necessarily best captured by linear dynamics. Furthermore, linear paths often force the model to generate the entirety of the sample simultaneously which can be harmful for two reasons:

- There is a limited opportunity for structured generation. For example, it could be favorable to generate one part of an image before another, using the former as context.
- As we will show, generating the entirety of the image at once can be an inefficient use of compute during sampling. Sampling efficiency is especially critical, as it is the central bottleneck for enabling the practical deployment of continuous time generative models in a myriad of applications (Ulhaq and Akhtar, 2024).

This motivates approaches that decompose generation into smaller, conditional subproblems, enabling generation that is more structured, controllable, and efficient. Generating partitions of an image sequentially, however, necessitates additional anchoring nodes to provide context to the subsequent generations. If there exceeds three nodes, a straight-line interpolation (i.e. optimal transport (OT)) no longer suffices as nodes are not necessarily co-linear.

In this work, we introduce Polynomial Flow Matching (PFM), a principled extension of Flow Matching that replaces linear transport paths with polynomial trajectories. By fitting polynomial flows through learned intermediate anchor points, PFM introduces curvature into the transport process. Crucially, polynomial trajectories enable sequential subspace generation, conditioning later stages on earlier ones.

Our experiments demonstrate that Polynomial Flow Matching can achieve comparable sample quality relative to linear Flow Matching while offering meaningful reductions in sampling cost for attention-dominated models. Beyond efficiency, PFM offers a geometric view of transport, suggesting that the choice of path—not just the endpoints—plays a critical role in scalable and structured generation.

Our contributions are as follows:

(1) PFM reduces the theoretical sampling complexity for attention dominated architectures. Our approach bridges the strengths of continuous-time flow and autoregressive-style models.
(2) Through experiments on MNIST, we show that using our PFM model, a quartic polynomial yields a speed-up over a quadratic while maintaining image quality. That is, the FID score for the quartic is the approximately equal to the FID score using the OT path.
(3) We were unable to achieve a speedup experimentally for the quadratic model relative to the OT linear model on MNIST and CIFAR-10. This is likely due to other overhead independent of the attention layer—exploring these results further should be a subject of future work.

2. Related Work

2.1. **Continuous Normalizing Flows.** Continuous Normalizing Flows (CNFs) model generative processes by transforming a simple base distribution into a complex target distribution via ordinary differential equations (ODEs). Chen et al. (2018) established the framework of Neural ODEs, learning vector fields to define these continuous transformations. However, training via ODE solvers is computationally expensive. To address this, Lipman et al. (2023) introduces FM as a scalable way to train CNFs. Our work aims to reduce the computational cost of attention mechanisms in FM models during sampling.

2.2. **Diffusion Models and Structured Generation.** Diffusion models generate data by reversing a gradual noising process (Ho et al., 2020). While effective, their iterative nature incurs high sampling costs, motivating work on accelerated solvers (Song et al., 2021; Lu et al., 2022) and noise scheduling (Nichol and Dhariwal, 2021). Beyond speed, recent work has focused on structured generation. Cascaded models decompose generation into resolution stages (Saharia et al., 2022), while recent advances in discrete domains have introduced “anchoring” to guide the generative process. Specifically, Rout et al. (2025) propose first predicting key ”anchor” tokens to condition the reconstruction of the remaining sequence. Our work bridges these works: we apply the intuition of non-linear interpolation from Benamou et al. (2019) and the structured guidance of anchoring from Rout et al. (2025).

3. Background

We follow Lipman et al. (2023) in describing CNFs. Consider a continuous time process over $t \in [0,1]$ that transports samples from an initial distribution p_0 at $t = 0$ to a target distribution p_1 at $t = 1$. The vector field $v_t(x)$ globally characterizes this transport process, providing a trajectory for each sample x at time t. This vector field describes a flow $\phi : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ (i.e. the “position of the particle at time t”) via the ODE:

$$\frac{d}{dt}\phi_t(x) = v_t(\phi_t(x)), \quad \phi_0(x) = x. \tag{1}$$

For a correct choice of v_t the transport procedure is simple: sample $x_0 \sim p_0$; solve the ODE for $t = 1$; output $\phi_1(x_0) \sim p_1$. Thus, the aim is to learn a good velocity field.

In the context of generative modeling, our training data composes of samples from an unknown density q. We must construct a path from a simple distribution $p_0 \sim N(x|0, I)$ to a complicated distribution $p_1 \approx q$. To learn the parameters of v_t (i.e. θ), Lipman et al. (2023) introduces the flow matching objective

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)}\|v_t(x) - u_t(x)\|^2, \tag{2}$$

Since the above objective is intractable, we can condition on the intended destination data sample x_1. Lipman et al. (2023) proves that the Flow Matching objective is equivalent to optimizing the Conditional Flow Matching objective

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1),p_t(x|x_1)}\|v_t(x) - u_t(x|x_1)\|^2. \tag{3}$$

Furthermore, Lipman et al. (2023) defines the conditional probability paths as

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

where we can choose functions μ_t and σ_t.

Consider the linear interpolation between the noise sample $x_0 \sim N(0, I)$ and the data sample x_1 as the flow map

$$\psi_t(x_0) = (1-t)x_0 + tx_1. \tag{4}$$

The above flow creates straight-line trajectories and implicitly defines the OT path described in Lipman et al. (2023) (we assume $\sigma_{min} = 0$ in our implementation).

$$\begin{aligned} \mathbb{E}[\psi_t \mid x_1] &= t\, x_1, \\ \text{Var}[\psi_t \mid x_1] &= (1-t)^2 I. \end{aligned} \tag{5}$$

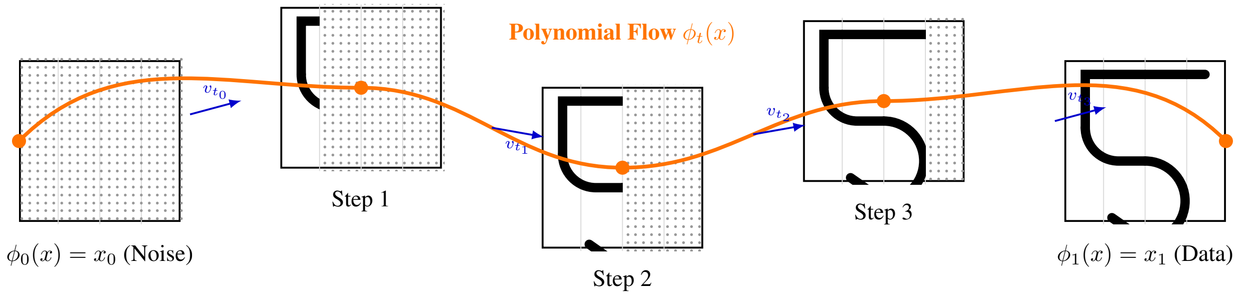


FIGURE 1. **Schematic of Polynomial Flow Matching (PFM).** A polynomial flow $\phi_t(x)$ transports samples from noise $\phi_0(x)$ to data $\phi_1(x)$ through intermediate anchor states that sequentially resolve four $d \times (d/4)$ subspaces.

4. APPROACH

TABLE 1. Sampling-time complexity for polynomial flow matching with attention-dominated models.

Per-part shape	Tokens per part	Attention cost per step	Total steps	Total attention cost	Speedup	Polynomial degree
$d \times d$	d^2	d^4	T	Td^4	$1\times$	1 (2 anchors)
$d \times \frac{d}{2}$	$\frac{d^2}{2}$	$\frac{d^4}{4}$	$2T$	$\frac{Td^4}{2}$	$2\times$	2 (3 anchors)
$d \times \frac{d}{4}$	$\frac{d^2}{4}$	$\frac{d^4}{16}$	$4T$	$\frac{Td^4}{4}$	$4\times$	4 (5 anchors)
$d \times \sqrt{d}$	$d^{3/2}$	d^3	$\sqrt{d}T$	$Td^{7/2}$	$\sqrt{d}\times$	$\sqrt{d}$ ($\sqrt{d} + 1$ anchors)

Table 1 motivates our approach. When attention dominates the computational cost, partitioning the generation space into k equal-sized components and generating each component sequentially reduces the overall sampling complexity by a factor of k (we can even achieve asymptotic speedup by setting the dimension to $\sqrt{d}$, although this is not practical). Importantly, however, when generating components sequentially, the content produced in earlier components must inform and condition the generation of subsequent components. Thus, we must take a page from the autoregressive book. We change the interpolation between x_0 and x_1 such that it must pass through "anchor" points which provide context from the prior generation. In case where the dimensions are $d \times \frac{d}{2}$, there is exactly one anchor point y_1 (i.e. the boundary between the top and bottom half). Since we must fit a polynomial to three points, using a linear model as proposed in Lipman et al. (2023) becomes naive, as doing so would unduly assume that the three points are co-linear. Thus, our polynomial must be of degree at least 2 (i.e. a quadratic) to fit three points.

Specifically,

$$\psi_t(x_0) = \begin{cases} (1+t)x_1 - tx_0, & t \in [-1, 0], \\ x_0 L_{-1}(t) + y_1 L_0(t) + x_2 L_1(t), & t \in [0, 1], \end{cases}$$

where L_{-1}, L_0, L_1 denote the quadratic Lagrange basis polynomials [1] with nodes $\{-1, 0, 1\}$, and $y_1 = f_\theta(x_1)$ is a learned intermediate representation.

The same logic follows for the $d \times \frac{d}{4}$ case with three anchor points y_1, y_2, y_3. The quartic flow map is

$$\psi_t(x_0) = \begin{cases} (1+t)x_1 - tx_0, & t \in [-1, 0], \\ x_0 L_{-1}(t) + y_1 L_{-\frac{1}{2}}(t) + y_2 L_0(t) + y_3 L_{\frac{1}{2}}(t) + x_4 L_1(t), & t \in [0, 1], \end{cases}$$

where $L_{-1}, L_{-\frac{1}{2}}, L_0, L_{\frac{1}{2}}, L_1$ are the quartic Lagrange basis polynomials with nodes $\{-1,\ -\frac{1}{2},\ 0,\ \frac{1}{2},\ 1\}$ and $y_i = f_{\theta_i}(\cdot)$ denote learned intermediate representations (Figure 1).

The transport paths are curved for the quadratic and quartic cases, so they are no longer OT. Our experimental aim is to minimize the objective

[1] The Lagrange basis is defined generally as $L_{t_i} = \prod_{j \neq i} \frac{t - t_j}{t_i - t_j}$

$$\int_0^1 D_{\mathrm{KL}}\left(p_t^{\mathrm{OT}}(x|x_1) \parallel p_t^{\mathrm{Poly}}(x|x_1)\right) dt, \tag{6}$$

while still using a polynomial of sufficient degree to achieve the aforementioned speedup.

5. Experimental Results

We evaluate and train our polynomial flow matching approach on the MNIST dataset, a widely-used benchmark for generative modeling consisting of 28x28 grayscale handwritten digit images. Images are then preprocessed using standard normalization techniques — input images are normalized from [0, 1] to [-1, 1] range (using the transform $x' = 2x - 1$), resized to their target dimensions via bilinear interpolation, and duplicated across 3 channels (replication) to make use of standard FID computation libraries.

We trained three flow-matching models — one standard flow-matching model (which we determine to be our "linear baseline" based on its linear spatial regime), one quadratic regime (which utilizes two sequential models to predict 16x32 each, fixing an anchor point with a quadratic spatial path), and one quartic regime (which utilizes four sequential models of 8x32 predictions each, with four anchor points to create our quartic polynomial spatial path). All experiments utilize the TorchCFM UNet architecture (used as library), a conditional flow matching model with attention mechanisms that has shown strong performance on image generation tasks.

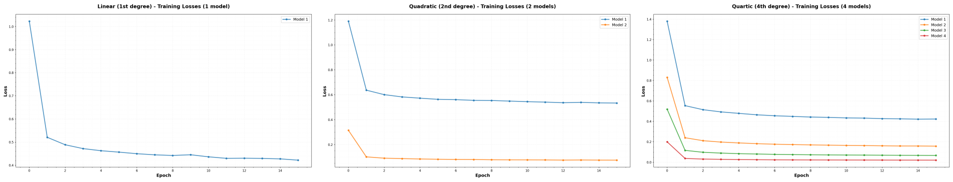


FIGURE 2. Training loss plots

Figure 2 shows the training losses, where we can see reasonably quick convergence among all models. For the two polynomial regimes, we can see multiple losses that depict the loss taken by each successive model (see key), where each successive model has lesser and lesser loss (absorbing in each stage the initial velocity prediction loss and the portion after each anchor point). In each regime, we can see approximate convergence around 8 epochs, with minimal loss reductions afterward.

We report two primary metrics — FID (Frechet Inception Distance) and sampling time per image (measured by timing the time to generate a batch of i.e. 2048 images for each model, and averaging) — to measure the quality of output and the efficiency of sampling output respectively, as we hope to improve the sampling efficiency while maintaining image quality.

Sampling each of the three trained models for 2048 samples, we generate the following sample images at each compression and polynomial training fidelity — linear, quadratic, and quartic.

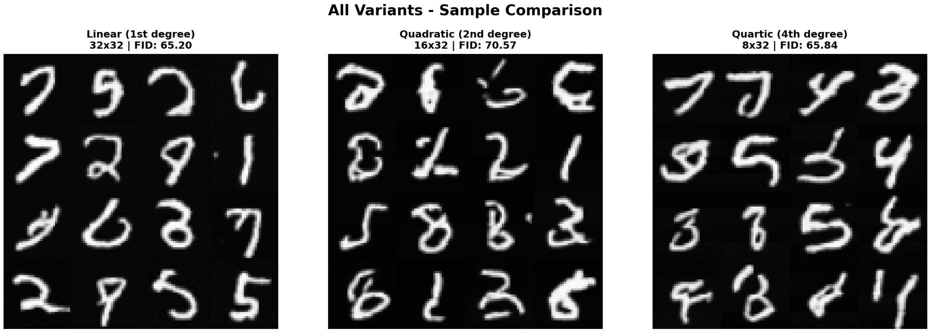


FIGURE 3. Sampling Results

TABLE 2. FID and sampling speed comparison across polynomial flow variants.

Variant	Size	Compression	FID	Time / img (ms)
Linear (1st degree)	32×32	$1/1$	65.1961	31.32
Quadratic (2nd degree)	16×32	$1/2$	70.5654	35.04
Quartic (4th degree)	8×32	$1/4$	**65.8411**	**31.80**

In the linear (baseline FM) sampling method, we can see the MNIST image shapes clearly in the majority of the images, with relatively smooth lines and low noise — although with some confusing/nonsensical shapes, like the top row's third and fourth images. In contrast, although we can see the general associated shapes in the quadratic polynomial method (sampled with two models at 16x32 each, flowing sequentially), the lines are much more choppy and noisy. In the quartic method (with a quartic polynomial and four sequential samplings of four models at 8x32 each, we can see yet again smoother and more defined shapes clearly resembling numerical outputs.

The FID scores of each method align with the qualitative results above, with the linear FM baseline achieving an FID of approximately 65.19, the quadratic method yielding a slightly worse 70.57, and we can see that the FID decreases again for the quartic method (at 65.84). Similarly with sampling time, we see that the linear FM model takes approximately 31.32 ms of sampling time per image, and though this increases for the quadratic model (to 35.84 ms per image), we can see it decrease again for higher-order splits in the quartic model (back down to 31.80ms).

This shows that the polynomial method is able to maintain FID and image quality but that sampling efficiency stays roughly constant, though we believe that the decrease in runtime between the quadratic and quartic methods shows this is could be largely due to overhead — and suggesting that the pattern could continue with future splits and higher-order polynomial fitting. We repeat the above procedure for CIFAR-10 with the results included in the Appendix.

6. DISCUSSION AND FUTURE WORK

Although we were unable to match the theoretical sampling speedup, we are able to maintain FID and image quality (as seen in FID evaluation and qualitative results) over the various polynomials. Furthermore, the increase in time from linear to quadratic then the decrease to quartic suggests that the polynomial method produces high overhead, that is able to decrease over further splits. Over larger images with even further splits, we might be able to achieve our anticipated speedup while maintaining FID and image quality. We also performed an ablation examining more esoteric polynomial paths over the $\frac{d}{2}$ split case (Figure 4). We can see that the quadratic case fits the anchor points much better than other methods, which suffer in image quality.

In the future, we will improve the baseline FID score for OT to gain a more meaningful comparison. We also hope to perform a full grid search over all possible sequential divisions (i.e. more divisions and in more dimensions) that may help us yield better results as illustrated in Figure 8. Lastly, we hope to test our method on other U-Net architectures. Additionally, while our analysis assumes attention-dominated sampling complexity, the lack of empirical speedup suggests that attention may not be the dominant computational bottleneck for the architecture we considered. In practice, fixed overheads such as convolutional operations can interfere with theoretical reductions in attention complexity. This is consistent with our observation that quadratic and linear variants exhibit similar sampling times despite differing theoretical attention costs. It is possible, however, that the theoretical benefit becomes visible experimentally when evaluated at a larger scale.

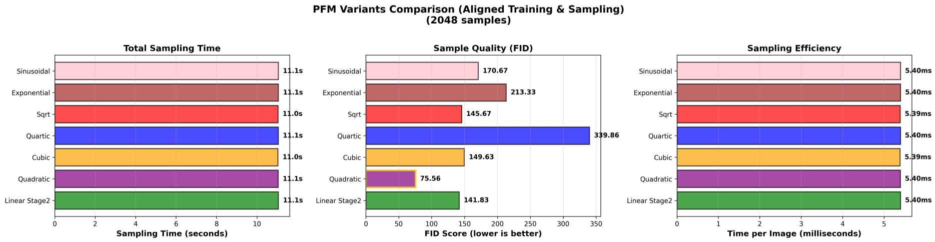


FIGURE 4. Other polynomial paths for $\frac{d}{2}$ splits

References

Jean-David Benamou, Thomas Gallouët, and François-Xavier Vialard. Spline interpolation in the wasserstein space. *SIAM Journal on Imaging Sciences*, 2019.

Ricky T.Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.

Yaron Lipman, Ricky T.Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2023.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution, 2024. URL `https://arxiv.org/abs/2310.16834`.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling. *Advances in Neural Information Processing Systems*, 2022.

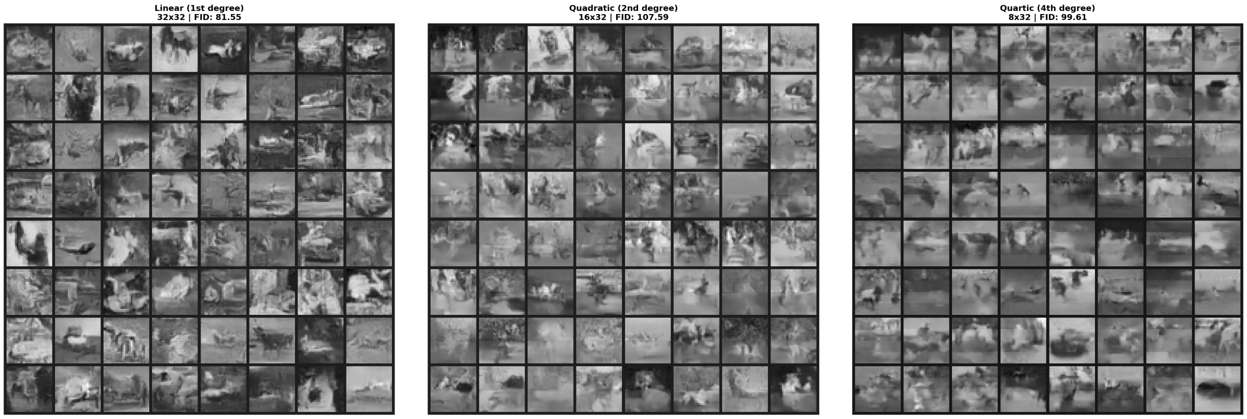
Alexander Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *International Conference on Machine Learning*, 2021.

Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Anchored diffusion language model. *arXiv preprint arXiv:2505.18456*, 2025.

Chitwan Saharia, William Chan, Saurabh Saxena, et al. Imagen: Photorealistic text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021.

Anwaar Ulhaq and Naveed Akhtar. Efficient diffusion models for vision: A survey, 2024. URL `https://arxiv.org/abs/2210.09292`.



(A) Linear (B) Quadratic (C) Quartic

FIGURE 5. CIFAR-10: Generated samples under different polynomial flow-matching parameterizations.

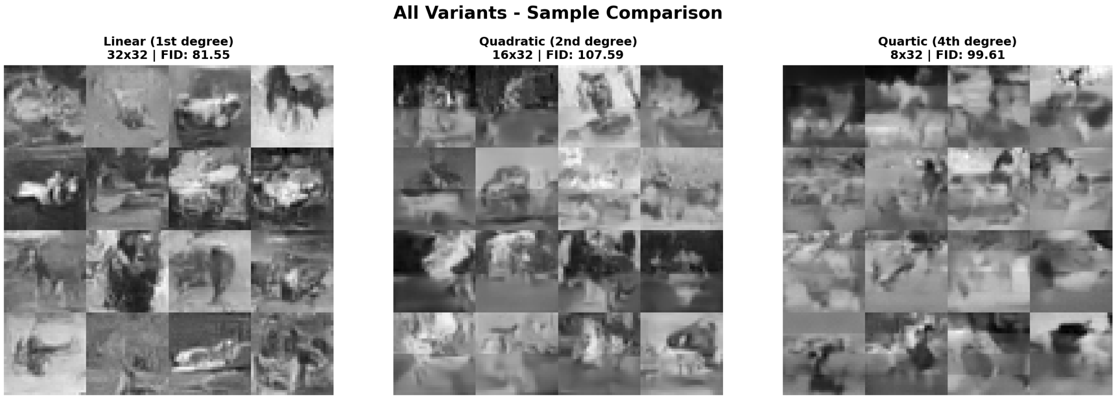
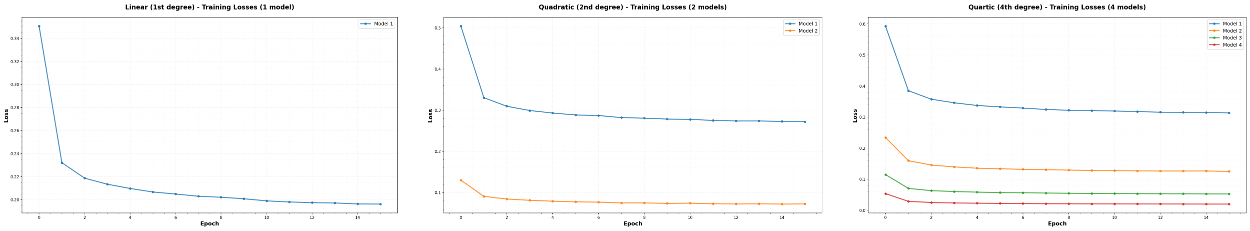


FIGURE 6. CIFAR-10: Composite grid of samples comparing all polynomial flow-matching variants.



(A) Linear (B) Quadratic (C) Quartic

FIGURE 7. CIFAR-10: Training loss curves for linear and higher-order polynomial flow-matching models.

7. APPENDIX

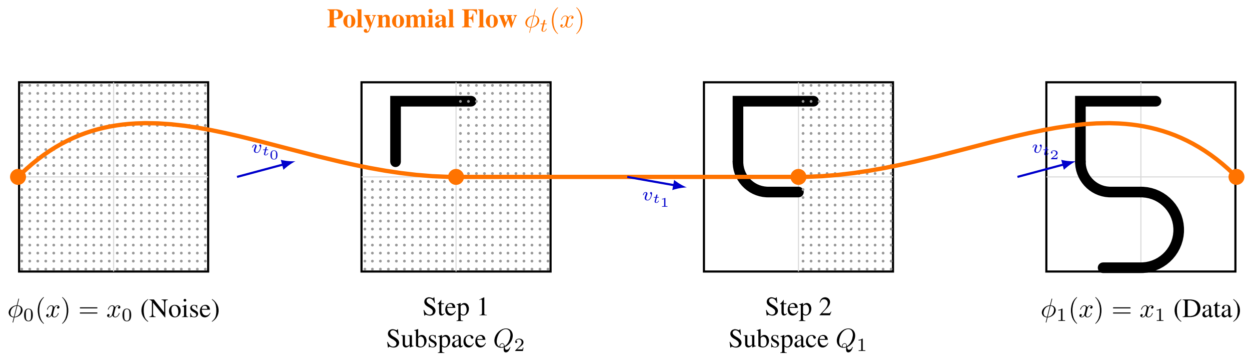


FIGURE 8. Schematic of PFM using quadrants