Posterior Augmented Flow Matching

George Stoica[1,2], Sayak Paul[3◇], Matthew Wallingford[4◇]
Vivek Ramanujan[2], Abhay Nori[2], Winson Han[4]
Ali Farhadi[2], Ranjay Krishna[2], and Judy Hoffman[1,5]

[1] Georgia Institute of Technology, USA
[2] University of Washington, USA
[3] Hugging Face, India
[4] Ai2, USA
[5] UC Irvine, USA
gstoica3@gatech.edu ◇Equal Contribution

Abstract. Flow matching (FM) trains a time-dependent vector field that transports samples from a simple prior to a complex data distribution. However, for high-dimensional images, each training sample supervises only a single trajectory and intermediate point, yielding an extremely sparse and high-variance training signal. This under-constrained supervision can cause *flow collapse*, where the learned dynamics memorize specific source–target pairings, mapping diverse inputs to overly similar outputs, failing to generalize. We introduce *Posterior Augmented Flow Matching* (PAFM), a theoretically grounded generalization of FM that replaces single-target supervision with an expectation over an approximate posterior of valid target completions for a given intermediate state and condition. PAFM factorizes this intractable posterior into (i) the likelihood of the intermediate under a hypothesized endpoint and (ii) the prior probability of that endpoint under the condition, and uses an importance sampling scheme to construct a mixture over multiple candidate targets. We prove that PAFM yields an unbiased estimator of the original FM objective while substantially reducing gradient variance during training by aggregating information from many plausible continuation trajectories per intermediate. Finally, we show that PAFM improves over FM by up to 3.4 FID50K across different model scales (SiT-B/2 and SiT-XL/2), different architectures (SiT and MMDiT), and in both class and text conditioned benchmarks (ImageNet and CC12M), with a negligible increase in the compute overhead. Code: `https://github.com/gstoica27/PAFM.git`.

Keywords: Flow Matching · Image Generation · Computer Vision

1 Introduction

Flow matching (FM) [15,17,25,29] trains a vector field that transports probability mass from a simple source distribution (e.g., Gaussian noise) to a complex target

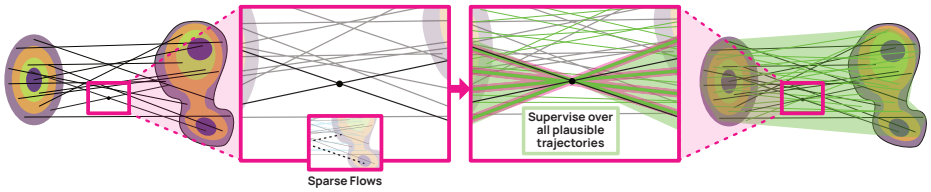

Fig. 1: Left: Standard FM provides a sparse, one-to-one supervision signal, pairing each intermediate point with a single target and yielding a single plausible flow. **Right**: Our PAFM aggregates supervision over the full posterior of compatible targets, producing a denser, more coherent set of plausible flows from each intermediate.

distribution (e.g., natural images), often conditioned on auxiliary inputs that specify *how* to flow to generate samples matching the given condition (e.g., "a photo of a dog") [7, 17]. Concretely, training a FM model proceeds as follows [15]: sample a data example and its condition from the target distribution, sample a point from the source distribution, and define a continuous interpolation (trajectory) between them. A random intermediate point along this trajectory is then sampled, and the model is asked to predict the velocity that moves this intermediate toward the target, conditioned on both the intermediate's location and the condition. Over many such samples, the model learns to approximate a global flow field that, in expectation, carries any source sample to a valid target consistent with the condition.

Under idealized assumptions—unlimited data, infinite model capacity, and perfect optimization—minimizing the FM loss recovers the true underlying vector field, and thus the data distribution, exactly [8, 15]. In practice, however, models are trained on complex, high-dimensional data for a finite number of steps and with bounded capacity. They observe only a vanishingly small subset of all possible trajectories, resulting in *sparse supervision* of the flow field [13].

This sparsity is inherent in the FM loss: each training sample provides feedback on a *single* trajectory (the one connecting the chosen source and target) at a given intermediate point [15]. In high-dimensional spaces, a single intermediate can lie on many plausible trajectories leading to different valid targets under the same condition—for instance, there are countless distinct images consistent with the prompt "a photo of a dog" [1]. Yet the model only ever sees one such outcome. Because the probability of revisiting the exact same intermediate state is effectively zero, the gradient at that state is estimated from a single noisy supervision signal. This under-constrained, high-variance training signal can lead to *flow collapse* [21], where the learned vector field fails to generalize and instead maps diverse inputs to overly similar or muddled outputs, indicating memorization of training pairings rather than learning the true continuous mapping.

We propose **Posterior Augmented Flow Matching (PAFM)**, a theoretically grounded generalization of the standard FM objective. The key insight is that for any intermediate state and condition, there is not a single "correct"

target data point, but rather an entire *posterior distribution* over valid targets in the data distribution. In other words, there are many plausible ways to complete the flow from the intermediate to a final sample that meets the condition. *All such trajectories represent valid training signals for the model.* However, these possible flows are not equally likely: some target completions have higher posterior probability than others. Therefore, PAFM trains the model to predict the expected velocity over all these possible continuation trajectories, *weighted* by their posterior likelihood. In effect, the model learns to follow the average direction toward the full set of outcomes consistent with the condition, instead of being pushed toward only *one* arbitrarily sampled endpoint.

While we cannot directly sample uniformly from the intractable true posterior over continuations, we show that it can be factorized into two simpler distributions that we can approximate: (1) the conditional probability of a given intermediate state if a particular target were the endpoint, and (2) the probability of the conditioning variable given that target. We show that under certain formulations, a sampling algorithm which draws a set of candidate targets for each intermediate and weighs them according to these probabilities forms an approximate posterior mixture. Moreover, by aggregating gradients from multiple possible flows (each weighted by its posterior likelihood) for each intermediate point, PAFM lower-bounds the variance of the flow matching training gradient.

In addition to our theoretical contributions, we demonstrate the applicability of PAFM in real-world settings and model-architectures. We conduct experiments on the popular class-conditioned ImageNet-1K dataset [5] across different model scales (SiT-B/2 and SiT-XL/2 [17]), and on the large-scale text-to-image CC12M [4] dataset with the MMDiT [7] model architecture. We investigate several practical strategies for constructing the candidate target set, including FAISS [6] k-nearest-neighbor retrieval in the latent space, augmentation of the source image via random crops at varying scales, and resampling from the VAE [22] moment distribution. PAFM improves flow matching across all settings and selection strategies while incurring only marginal computational overhead, highlighting its seamless integration into modern generation workflows.

2 Related Work

Continuous-time generative modeling and flow matching. Continuous-time generative models view sampling as integrating an ODE or SDE between a simple source distribution and a complex target distribution, covering stochastic interpolants, diffusion models, and flow-based methods [3, 11, 24]. Flow matching (FM) and conditional flow matching (CFM) train a vector field by matching it to the ground-truth probability current along prescribed interpolants between source and data samples [15, 26]. Subsequent work has scaled FM and CFM to large image and video models by carefully designing architectures and interpolants, including interpolant transformers such as SiT [17] and large-scale rectified-flow-style models [7, 16]. These methods all share the same supervision pattern: each

intermediate point along an interpolant is paired with a *single* target sample, yielding a one-to-one training signal.

Rectified flows and trajectory shaping. Rectified flows encourage straight trajectories between source and target, enabling near one-step generation and linking flow-based methods to optimal transport formulations [3, 16]. At scale, rectified-flow transformers match or surpass diffusion models on high-resolution text-to-image benchmarks via improved time sampling and architecture design [7], while complementary work studies how controlling trajectory curvature can reduce solver error and accelerate sampling [14]. Our formulation is compatible with these ideas: PAFM leaves the interpolant and geometric regularization unchanged and instead modifies how supervision is aggregated over targets.

Theoretical analyses and improved flow matching objectives. The FM objective enjoys strong asymptotic guarantees: under mild assumptions on vector-field parameterization and sample complexity, FM attains near minimax-optimal convergence rates in Wasserstein distance, comparable to diffusion models [8]. Follow-up work tightens the link between learned and ideal probability paths, motivating divergence-matching augmentations to CFM [13] and relating rectified flows to optimal transport via straightening and gradient constraints [9]. These analyses refine global objectives and path properties but still assume the standard one-target-per-intermediate coupling.

Failure modes and sparse supervision in flow-based models. In realistic regimes, models are finite-capacity, trained for finite steps, and only observe a sparse subset of source–target trajectories [23]. Recent empirical and theoretical work has exposed failure modes of flow-based models under these conditions, including degeneracies induced by straight-path objectives and deterministic couplings [21]. Gradient-variance analyses further show that rectified-flow-style objectives can memorize arbitrary pairings in low-stochasticity regimes, even when interpolant lines intersect, leading to ill-defined vector fields at inference [21]. Our work is complementary: starting from the same observation that supervision is sparse and under-constrained, we replace one-to-one supervision with a posterior expectation over all compatible targets for each intermediate latent.

3 Background: Flow Matching

We provide a formal overview of conditional flow matching and the flow matching (FM) objective as it pertains to our work.

Assumptions. Let $\mathcal{N}(0, I_d) = p_{source}$ be a source distribution characterized by the standard Gaussian. Similarly, let p_{data} be the target data distribution with unknown support. Samples drawn from p_{source} are denoted by ϵ^i, while those from p_{data} are denoted by (z^i, y^i) pairs, where z^i is a data point and y^i is its associated condition. For our purposes, z^i represents an image and y^i a text conditioning. We may equivalently sample $z^i \sim p_{data}(\cdot|y^i)$ or $y^i \sim p_{data}(\cdot|z^i)$.

Flow matching describes the flow between p_{source} and p_{data} over *normalized and reversed* time. Let Unif$[0,1)$ describe the distribution over this time, and denote its samples by t. Training FM models involves sampling from the source and target distribution (e.g., ϵ^i and (z^i, y^i)), defining a trajectory between their respective samples and then further sampling an intermediate point which lies along their flow.

Let $\alpha(t), \beta(t)$ be two time-dependent monotonic functions that define the path between ϵ^i and z^i, such that $\alpha(0) = \beta(1) = 0$ and $\alpha(1) = \beta(0) = 1$. For notational simplicity, we refer to point-evaluations at t by α_t and β_t respectively. Using α_t and β_t, we sample the intermediate point as $z_t^i = \alpha_t \epsilon^i + \beta_t z^i$. These samples are assumed to lie on a conditional probability path, given by $z_t^i \sim p_t(\cdot|z^i) = \mathcal{N}(\beta_t z^i, \alpha_t^2 I_d)$. Observe that $p_0(\cdot|z^i) = p_{source}$ and $p_1(\cdot|z^i) = \delta_{z^i}$ where δ_{z^i} is the "Dirac delta" distribution. We characterize the flow between ϵ^i and z^i by the time-derivative of α_t and β_t respectively: $v(z_t^i|z^i) = \dot{\alpha}_t \epsilon^i + \dot{\beta}_t z^i$. We describe the vector-space over which ϵ^i, z^i, z_t^i are supported, as the "latent-space."

The flow matching objective. While α, β can take arbitrary forms, $\alpha(t) = t$ and $\beta(t) = 1 - t$ (with $\dot{\alpha}(t) = 1, \dot{\beta}(t) = -1$) is the most popular and widely used formulation [17, 29]. Thus, we define $z_t^i = t\epsilon^i + (1-t)z^i$ and $v(z_t^i|z^i) = \epsilon^i - z^i$ respectively. FM involves training a model to produce flows from arbitrary z_t^i to real targets (e.g., z^i) based on a condition (e.g., y^i). Let $f_\theta(z_t^i|t, y^i)$ describe this model and let θ be its parameterization. The flow matching objective is finally given by,

$$\mathcal{L}^{(FM)}(\theta) = \mathbb{E}_{t\sim Unif[0,1),(z^i,y^i)\sim p_{data}(\cdot),z_t^i\sim p_t(\cdot|z^i)} ||f_\theta(z_t^i|t, y^i) - v(z_t^i|z^i)||^2 \quad (1)$$

Thus, FM models learn the *expected* flows that pass through each intermediate sample towards the entire target distribution, conditioned on each y^i. In practice, the formal objective in Eq. (1) is transformed to,

$$\min \overline{\mathcal{L}^{(FM)}}(\theta) = \min \frac{1}{N} \sum_{(z^i,y^i)\sim D} ||f_\theta(z_t^i|t, y^i) - v(z_t^i|z^i)||_2^2 \quad (2)$$

when training flow models, where D is the target dataset.

Sparse supervision. While Eq. (2) is an unbiased estimator for Eq. (1) and its global minima is the global optimum under flow matching, we immediately observe its inherent risk of sparse supervision. For any given intermediate latent point z_t^i and given condition y^i, a model is supervised with only a single target trajectory $v(z_t^i|z^i)$. However, in the high-dimensional and complex settings of contemporary generation tasks, z_t^i could validly lie on a multitude of trajectories flowing to different targets $\{z^j\}$ that all satisfy the same condition y^i. By only providing one such target, Eq. (2) gives a high-variance and under-constrained signal over each latent. As a result, models often rely on sparse coverage of the true conditional vector fields within the latent space, leading to collapsed flows when transporting from source to target data [2].

4 Posterior Augmented Flow Matching (PAFM)

The focus of this section is two-fold. First, we show how the problem of *sparse supervision* can be addressed by reformulating the flow matching objective to enable simultaneous *multiple* trajectory supervision for training flow models. We coin this new objective Posterior Augmented Flow Matching (PAFM), and prove that it improves gradient stability over flow matching under the *same* theoretical assumptions. Second, we show how PAFM can be seamlessly adapted for training rectified flow models.

4.1 Theoretical Results

Reformulating the flow matching objective. We begin by observing that it is theoretically feasible for multiple samples drawn from p_{source} and p_{data} to contain paths which flow through the same z_t^i. Moreover, the likelihood of sampling these trajectories is given by a posterior-distribution over the *same* distributions supporting the flow matching objective. Coupling these observations together, we introduce Posterior Augmented Flow Matching (PAFM) as,

$$\mathcal{L}^{(PAFM)}(\theta) = \mathbb{E}_{\substack{t\sim Unif[0,1), y^i\sim p_{\text{data}}(y),\\ z_t^i\sim p_t(\cdot), z^j\sim p_t(\cdot|z_t^i,y^i)}} ||f_\theta(z_t^i|t,y^i) - v(z_t^i|z^j)||^2 \tag{3}$$

where $p_{data}(y)$ defines the probability distribution over the conditions, $p_t(\cdot)$ is a (unknown) probability distribution over the latent-space conditioned on t, and $p_t(\cdot|z_t^i, y^i)$ is the *posterior* distribution of the target data given a *fixed* latent-point z_t^i and condition y^i.

Importantly, Eq. (3) is mathematically equivalent to the flow matching objective introduced by Eq. (1). We demonstrate this in Theorem 1.

Theorem 1 *The Posterior Augmented Flow Matching objective $\mathcal{L}^{(PAFM)}$ is an unbiased estimator of the standard Flow Matching objective $\mathcal{L}^{(FM)}$. In expectation, the two objectives are identical.*

$$\mathcal{L}^{(PAFM)}(\theta) = \mathcal{L}^{(FM)}(\theta) \tag{4}$$

Proof. It suffices to show that the joint probability distributions over which the expectations are taken over are identical. FM (Eq. (1)) takes its expectation over the joint probability distribution given by $p(t) \cdot p_{data}(z, y) \cdot p_t(z_t|z)$. The PAFM (Eq. (3)) takes its expectation over the joint distribution given by $p(t) \cdot p_t(z_t) \cdot p_t(z, y|z_t)$. Note that by the flow matching assumptions introduced

in Sec. 3, $z \perp y|z_t$. Starting from the distributions for FM,

$$p(z,y)p_t(z_t|z) = \frac{p(z,y)p(z_t,z,t)}{p(z,t)} = \frac{p(y|z)p(z_t,t|z)p(z)^2}{p(z)p(t)} \tag{5}$$

$$= \frac{p(y|z)p(z_t,t|z)p(z)}{p(t)} = \frac{p(y,z_t,t,z)}{p(t)} \tag{6}$$

$$= \frac{p(z|z_t,y,t)p(z_t,y,t)}{p(t)} = p_t(z|z_t,y)p_t(z_t)p(y) \tag{7}$$

As the two distributions are identical, so is their expectation over the same loss function $||f_\theta(z_t|t,y) - v(z_t|z)||^2$. □

PAFM theoretically reduces gradient variance. Notably, and in contrast to flow matching, Eq. (3) enables us to optimize f_θ by sampling multiple target points $\{z^j\}$ for every z_t^i, yielding a *denser* supervision signal during training. We now demonstrate that this fact alone provides gradient estimates for the underlying optimal f_θ that *lower-bound* the variance of the gradient estimates from the FM objective.

Theorem 2 *Let $\phi(z^j|z_t^i,y^i) = ||f_\theta(z_t^i|y^i,t) - v(z_t^i|z^j)||^2$ be the per-sample flow matching loss. Define its gradient with respect to f_θ as $g(z^j|z_t^i,y^i) = \nabla_f \phi(z^j|z_t^i,y^i) = 2\left(f_\theta(z_t^i|y^i,t) - v(z_t^i|z^j)\right)$. For a fixed z_t^i, y^i, let $Var_{p_t(\cdot|z_t^i,y^i)}\left[g(z^j|z_t^i,y^i)\right] = \Sigma_g(z_t^i)$. The PAFM objective, when optimized using $K > 1$ samples per z_t^i via Self-Normalized Importance Sampling (SNIS), yields a gradient estimator with variance $\Sigma_g / ESS(z_t^i)$, where $ESS(z_t^i) \geq 1$ is the Kish Effective Sample Size [28].*

Proof. First, notice that,

$$p_t(z^j|z_t^i,y^i) = \frac{p_t(z^j,z_t^i,y^i)}{p_t(z_t^i,y^i)} = \frac{p_t(z_t^i,y^i|z^j)p(z^j)}{p_t(z_t^i,y^i)} = \frac{p_t(z_t^i|z^j)p_t(y^i|z^j)p(z^j)}{p_t(z_t^i,y^i)} \tag{8}$$

$$\propto p_t(z_t^i|z^j)p_t(y^i|z^j)p(z^j) \tag{9}$$

Concretely, the posterior $p_t(z^j|z_t^i,y^i)$ is directly proportional to the individual probabilities $p_t(z_t^i|z^j)$ $p_t(y^i|z^j)p(z^j)$ up to a fixed normalizing factor in z_t^i, y^i. Let $\hat{p}_t(z^j|z_t^i,y^i) = p_t(z_t^i|z^j)p_t(y^i|z^j)p(z^j)$ be the un-normalized approximation of $p_t(z^j|z_t^i,y^i)$. Now, define a (un-normalized) distribution $q_t(z^j|z_t^i,y^i)$, and sample $K \geq 1$ elements from the distribution to obtain $\{z^j\}_{j=1}^K \sim q_t(z^j|z_t^i,y^i)$. Leveraging self-normalizing importance sampling (SNIS), we define weights α_j,

$$\alpha_j = \frac{\hat{p}_t(z^j|z_t^i,y^i)}{q_t(z^j|z_t^i,y^i)} = \frac{p_t(z_t^i|z^j)p_t(y^i|z^j)p(z^j)}{q_t(z^j|z_t^i,y^i)} \tag{10}$$

Defining $q_t(z^j|z_t^i,y^i) = p(z^j)$, α_j reduces to $p_t(z_t^i|z^j)p_t(y^i|z^j)$. Of these, $p_t(z_t^i|z^j)$ is simply the conditional probability path, and $p_t(y^i|z^j)$ is the posterior-distribution

of the condition given the target data point. We then obtain the SNIS normalized weights by computing $w_j = {}^{\alpha_j}/_{\sum_{k=1}^{K} \alpha_k}$. By definition, the SNIS estimator of $\mathbb{E}_{\{z^j \sim p_t(\cdot|z_t^i, y^i)\}} \left[g(z^j|z_t^i, y^i) \right]$ is $\mu_g^{(SNIS)}(z_t^i|K) = \sum_{j=1}^{K} w_j g(z^j|z_t^i, y^i)$.

We further define the Kish effective sample size (ESS) as, $ESS(z_t^i) = \left[\sum_{j=1}^{K} w_j\right]^2 / \sum_{j=1}^{K} w_j^2 = {}^1/_{\sum_{j=1}^{K} w_j^2}$ Thus, $1 \leq ESS(z_t^i) \leq K$. Putting all the pieces together, we obtain,

$$\mathrm{Var}\left(\mu_g^{(SNIS)}(z_t^i|K)\right) = \sum_{j=1}^{K} w_j^2 \mathrm{Var}\left(g(z^j|z_t^i, y^i)\right) = \sum_{j=1}^{K} w_j^2 \Sigma_g = \frac{\Sigma_g}{\mathrm{ESS}(z_t^i)} \quad (11)$$

Now, observe that when $K = 1$, the SNIS estimator *reduces* to the gradient variance of the flow matching objective from Eq. (1):

$$\mu_g^{(SNIS)}(z_t^i|1) = \sum_{j=1}^{1} w_j g(z^j|z_t^i, y^i) = g(z^j|z_t^i, y^i) \quad (12)$$

$$= \nabla_f ||f_\theta(z_t^i|t, y^i) - v(z_t^i|z^j)||^2 \quad (13)$$

$$\rightarrow \mathrm{Var}\left(\mu_g^{(SNIS)}(z_t^i|1)\right) = \Sigma_g \quad (14)$$

Thus, by choosing $K \geq 1$, PAFM reduces the variance of each gradient estimate over z_t^i by a factor of $\mathrm{ESS}(z_t^i) \geq 1$ compared to flow matching. □

This further highlights an important result: PAFM is a *generalization* of the flow matching objective, and reduces to it when $K = 1$.

4.2 Model Training

We now translate the formal definition of posterior augmented flow matching (Eq. (3)) to a practical training objective for rectified flow models. Posterior-augmented flow matching (PAFM) extends the standard flow matching (FM) objective by introducing three components: (i) a target set $\{z^j\}_{j=1}^{K}$ providing additional supervision trajectories, (ii) the conditional probability path $p_t(z_t^i|z^j)$ capturing how plausible reaching each target is from the current latent and (iii) the condition likelihood $p(y^i|z^j)$ measuring the compatibility of targets with the input condition.

Selecting target points $\{z^j\}_{j=1}^{K}$. The choice of candidate target points is where PAFM offers its greatest flexibility, and the optimal strategy may vary across generative settings, data modalities and training regimes. While the theoretical objective (Eq. (3)) is defined over the true posterior $p_t(z^j|z_t^i, y^i)$, it is intractable and any practical approximation must be defined over a finite support set. PAFM is agnostic to how these candidates are sourced: they may be sampled from the target distribution, augmentations of data from the distribution, perturbations of data in the latent space, etc... The importance weights w_j then re-weight

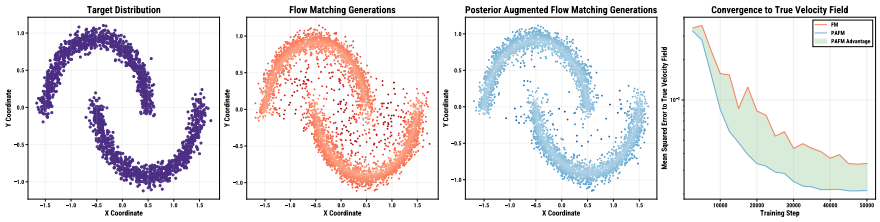

Fig. 2: Posterior augmented flow matching is more robust than flow matching. We train two rectified flow models to generate two crescent moon distributions (left). Second-left: a model trained with FM generates many points in between the two moons. Second-right: the same model trained with PAFM has far less of an issue. Right: PAFM estimates the true velocity field across t significantly better.

these candidates according to their posterior likelihoods, shaping the supervision signal toward the posterior distribution. We investigate three proposal methods in Sec. 5, yet PAFM is technically compatible with nearly any strategy.
Note: we always control $\{z^j\}_{j=1}^K$ to include z^i (i.e., the target data point from which z_t^i is sampled under traditional FM).

The conditional probability path, $p_t(z_t^i|z^j)$. The conditional probability path describes how plausible each target z^j is for a given latent-point z_t^i. PAFM uses the exact conditional probability path from the FM assumptions in Sec. 3. Recall that this is a Gaussian distribution parameterized by $\mathcal{N}(\beta_t z^j, \alpha_t^2 I_d)$ where $\alpha_t = t$ and $\beta_t = 1 - t$. The log-likelihood is thus given by,

$$\log p_t(z_t^i|z^j) = -\frac{||z_t^i - (1-t)z^j||^2}{2t^2} + C \tag{15}$$

where C is the normalizing constant. In practice, we compute the un-normalized likelihood $\exp\left(-||z_t^i - (1-t)z^j||^2/2t^2\right)$.
Note: Substituting $z_t^i = t\epsilon^i + (1-t)z^i$ into Eq. (15) and simplifying yields the equivalent formulation: $\exp\left(-||t\epsilon^i + (1-t)(z^i - z^j)||^2/2t^2\right)$. Because $\{t, \epsilon^i, z^i\}$ are constant for all $z^j \in \{z^j\}_{j=1}^K$, the sharpness of $p_t(z_t^i|z^j)$ largely depends on the distance between each z^j and z^i.

The condition likelihood, $p_t(y^i|z^j)$. The condition likelihood depicts how suitable each target z^j is for the condition y^i. In the case of class-conditioned flow matching, this is a deterministic function $\forall t$: $p_t(y^i|z^j) = \begin{cases} 1 \text{ if } y^i = y^j \\ 0 \text{ otherwise} \end{cases}$. In the case of continuous-conditioned generation (e.g., text-to-image), this distribution is unknown apriori and must be approximated—forming the second design choice in PAFM. While many viable choices exist, one intuitive strategy for modeling the distribution is employing a vision-language model such as CLIP [20] to compute the alignment of y^i and z^j.

Note: We use the naive class-conditioned variant for our text-to-image experiments owing to its simplicity, and leave more refined condition likelihood approximations to future work.

The PAFM training objective. With a defined candidate selection strategy for $\{z^j\}_{j=1}^K$ and chosen $p_t(y^i|z^j)$ distribution, we can train with PAFM using the following general objective,

$$\min \overline{\mathcal{L}^{(PAFM)}}(\theta) = \min \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} w_j ||f_\theta(z_t^i|t, y^i) - v(z_t^i|z^j)||_2^2 \tag{16}$$

where $w_j = p_t(z_t^i|z^j)p_t(y^i|z^j)$, and $\sum_{j=1}^K w_j = 1$.

The PAFM batch step. Algorithm 1 illustrates a batch step with PAFM, where navy text denotes additions to the standard FM objective.

Algorithm 1 Posterior Augmented Flow Matching Batch Step

Input: Model f_θ, batch $B = \{(z^i, y^i, \epsilon^i)\}_{i=1}^n$ where $(z^i, y^i) \sim p_{\text{data}}(\cdot)$, $\epsilon^i \sim \mathcal{N}(0, \mathrm{I}_d)$, learning rate γ. Candidate pool $\{(z^j, y^j)\}_{j=1}^K$ for each i s.t. $(z^i, y^i) \in \{(z^j, y^j)\}_{j=1}^K$.
Output: Updated model parameters θ
$\mathcal{L}(\theta) = 0$
for i in range(n) **do**
  $t \sim U[0, 1)$, $z_t^i = t\epsilon^i + (1-t)z^i$
  for j in range(K) **do**
    $v(z_t^i|z^j) = (z_t^i - z^j)/t$, $\hat{w}_j = p_t(z_t^i|z^j)\, p(y^i|z^j)$
  end for
  $\mathcal{L}(\theta)$ += $\sum_{j=1}^K \left(\hat{w}_j / \sum_{k=1}^K \hat{w}_k\right) \|f_\theta(z_t^i|t, y^i) - v(z_t^i|z^j)\|^2$
end for
$\theta \leftarrow \theta - \frac{\gamma}{N} \nabla_\theta \mathcal{L}(\theta)$

Understanding PAFM with an example. To build intuition for PAFM's effect on training rectified flow models, we train two four-layer MLP models to generate a simple two-dimensional crescent moon distribution in Fig. 2 (left). Under standard FM, the model generates a substantial number of spurious points in the region between the two crescents (second-left), despite training for an extensive period of time (50K steps). This is a consequence of sparse supervision: the model observes few trajectories passing through these intermediate regions, thereby limiting its capability at estimating the true velocity field. However, training the same model with PAFM largely resolves this artifact (second-right), as each intermediate point receives supervision from multiple weighted targets rather than a single trajectory—enabling the model better estimate the true velocity field. The rightmost panel quantifies this gap directly: we plot the mean squared error between the learned velocity and the analytically computed true velocity field across all denoising steps (t) over the course of training. Notably, PAFM steadily lowers its velocity field error throughout training, whereas FM exhibits substantial variance in its field estimates across training steps. Moreover, PAFM ultimately converges to a substantially more accurate velocity field than FM. We provide additional implementation details in App. A.

5 Experiments

Posterior-augmented flow matching (PAFM) is general and can be applied anywhere flow matching (FM) is used. As described in Sec. 4.2, a primary design choice when training with PAFM is selecting the candidate target set $\{z^j\}_{j=1}^K$ for each z^i. While many proposal mechanisms are viable, we find that a simple nearest-neighbor retrieval in latent-space consistently improves over FM across class-conditioned (ImageNet-1K [5]) and text-to-image (CC12M [4]) generation, across architectures (SiT [17] and MMDiT [7]), and across model scales (SiT-B/2 and SiT-XL/2). We further show that PAFM can empirically reduce the gradient variance during training compared with FM, show that PAFM improves performance throughout training, analyze the computational overhead of training with PAFM and ablate alternative target selection strategies—demonstrating the inherent tailorability of PAFM. Additional experiments applying PAFM without REPA, on more advanced objectives (e.g., ΔFM [25]), with uniform weighting, and comparing its training to FM are in App. B.

Setup. We train all our models for 400K iterations with a batch size of 256 on images with 256x256 resolution, and with the REPA loss [29]. All experiments are conducted on single nodes with 8 H100 GPUs. We evaluate all our models without classifier-free guidance (CFG) [12] using the ODE sampler with 50 denoising steps, and report Fréchet Inception Distance (FID) [10], sFID [18], precision and recall over 50K samples, as is standard.

5.1 Nearest Neighbor PAFM Results

Our primary implementation of PAFM forms the target set $\{z^j\}_{j=1}^K$ by retrieving the K nearest-neighbors (KNN) of each training example z^i in latent-space. Neighbors are extracted *once* for each z^i using FAISS indices [6] computed over the train dataset during the data-preprocessing, and training simply loads the precomputed neighbors alongside each z^i—introducing no additional online compute. Nearest neighbors allow a healthy effective sample size throughout training, as candidates are geometrically close to z^i.

Class-conditioned generation on ImageNet-1K. We restrict the neighbor search to images sharing the same class label, computing an independent FAISS index per class over the image latent encodings. Since all candidates share the same class-label, the condition likelihood $p_t(y^i|z^j) = 1, \forall z^j \in \{z^j\}_{j=1}^K$. Tab. 1 shows the results when training SiT-B/2 and XL/2 models with PAFM compared to FM. Notably, PAFM improves over FM-trained models in *nearly all* metrics across *both* model scales and

Table 1: PAFM improves FM on ImageNet-1K. Results on FID50K without CFG and NFE=50 after 400K training iterations. Models are trained with REPA.

Model	Obj.	K	FID↓	sFID↓	Prec.↑	Rec.↑
SiT-B/2	FM	–	27.57	11.44	<u>0.57</u>	0.63
	PAFM	64	25.45	6.91	**0.58**	<u>0.64</u>
	PAFM	16	**24.88**	**6.81**	**0.58**	<u>0.64</u>
	PAFM	8	<u>25.25</u>	<u>6.90</u>	**0.58**	**0.65**
	PAFM	4	25.47	6.97	**0.58**	<u>0.64</u>
SiT-XL/2	FM	–	<u>11.14</u>	<u>8.25</u>	<u>0.67</u>	**0.66**
	PAFM	16	**9.85**	**5.24**	**0.68**	<u>0.65</u>

all K, highlighting its efficacy. Interestingly, PAFM peaks at $K = 16$ for SiT-B/2 reflecting a bias-variance tradeoff: too few neighbors limit variance reduction while too many include distant candidates that add noise rather than signal.

Text-to-image generation on CC12M. Without discrete class labels, we approximate class-conditioning in two stages. First, we encode all images and captions using the SigLIP2 [27] So400M/14 encoder[§] and retrieve the M images whose encodings have highest cosine similarity to y^i. Second, from this shortlist we select the K nearest neighbors to z^i in latent space, as in the class-conditioned setting. This ensures candidates are both semantically compatible with the condition and geometrically close in latent space. We use $M = 128$ and $K = 32$ in our experiments. Since the candidate set has already been filtered for semantic compatibility with y^i, we treat the condition likelihood $p_t(y^i|z^j)$ as approximately uniform across the set and omit it from the weight computation. Thus, only the conditional path likelihood $p_t(z_t^i|z^j)$ determines the relative importance weights—w_j. We train MMDiT models on CC12M with PAFM and FM, and evaluate the resultant checkpoints after 400K training iterations on a held-out validation set of 50K examples. Overall, we observe that FM achieves an FID50K of 10.37, compared to **9.45** when using PAFM, illustrating how PAFM can be utilized to improve flow matching even in more challenging text-to-image settings.

5.2 Alternative Target Selection Strategies

While selecting $\{z^j\}_{j=1}^K$ via nearest neighbor search is sufficient to consistently improve over flow matching, we emphasize that there may be many viable alternative strategies, depending on the desired application setting. In this section, we ablate two additional selection strategies, finding promising results with each. All models trained in this section are REPA-SiT-B/2 with 256 batch size on ImageNet-1K 256×256 resolution for 400K iterations.

Table 2: PAFM with random augmentations improves FM. Results on FID50K without CFG and NFE=50 after 400K training iterations. The target set for PAFM is obtained by center cropping each training image to an "augmented resolution" ("Aug. Res.") and then randomly resize cropping down to 256x256 resolution $K = 5$ times.

Model	Obj.	Aug. Res.	FID↓	sFID↓	Prec.↑	Rec.↑
SiT-B/2	FM	256 × 256	27.57	11.44	0.57	0.63
	PAFM	281 × 281	<u>25.03</u>	6.84	<u>0.58</u>	**0.65**
	PAFM	320 × 320	**24.15**	**6.70**	**0.59**	**0.65**
	PAFM	384 × 384	25.43	7.13	<u>0.58</u>	**0.65**
	PAFM	512 × 512	25.44	<u>6.72</u>	<u>0.58</u>	<u>0.64</u>
SiT-XL/2	FM	256 × 256	11.14	8.25	0.67	<u>0.66</u>
	PAFM	281 × 281	**9.77**	<u>5.47</u>	<u>0.68</u>	**0.67**
	PAFM	320 × 320	<u>9.96</u>	**5.46**	**0.69**	<u>0.66</u>

Augmenting the source image. A natural alternative to creating the target set for each training image z^i by retrieving its nearest neighbors, is by augmenting the training image itself. Depending on the augmentation, resultant images may provide different views (e.g., from random crops), styles (e.g., jitter or applying style transfer methods), or compositions (e.g., horizontal flips, rotations) of the original image—increasing the diversity of the underlying supervision signal. In this first foray, we focus on

[§] `https://huggingface.co/google/siglip2-so400m-patch14-384`

the effect of sampling different spatial views of z^i. Specifically, we first center-crop each image to $256R \times 256R$ resolution, where $R \geq 1$ is referred to as the "augmentation scale" (aug. scale). We then take $K-1$ random resize crops down to 256×256 resolution, resulting in a target set of $K-1$ different "views", and finish by adding z^i to the set. Notably, the augmentation scale controls the spatial diversity of the resulting targets: small factors yield images that are nearly identical to z^i, while larger factors produce more varied views. While this strategy rests on the assumption that random crops are plausible samples from the target image distribution, we posit that it is reasonable given its widespread use for training foundation vision models [19,20,27]. Tab. 2 shows results on SiT-B/2 and SiT-XL/2 with $K = 5$. PAFM improves over FM at every augmentation scale, substantially lowering FID by up to 3.42. We compute these augmentations online in our experiments; however they could equally be precomputed and instead directly loaded during training to eliminate this overhead.

Sampling from the VAE moment distribution. All our rectified flow matching models are trained in the latent space of a VAE [22], where targets are created by first encoding each training image into a multivariate Gaussian distribution (termed "moment") and then sampling from it—amounting to small random perturbations around the mean encoding. Flow matching (and all our previously described PAFM approaches) sample once from the moment at each step. Thus, a very simple target selection strategy for PAFM is instead sampling from this moment K times. While this is the cheapest selection strategy proposed (sampling from the moment requires minimal overhead), the VAE posterior is tightly concentrated. This yields candidates that explore only a small neighborhood in latent space, and can limit their diversity. Tab. 3 shows that this simple strategy can improve over flow matching when $K = 10$. This shows how PAFM can extract gains from even very modest transformations of the underlying target point z^i, underscoring its flexibility as a training framework.

Table 3: Sampling K times from the VAE improves FM. Results on FID50K without CFG and NFE=50 after 400K training iterations. The target set for PAFM is obtained by randomly sampling from the training image's VAE moment K times (FM always samples once).

Model	Obj.	K	FID↓	sFID↓	Prec.↑	Rec.↑
SiT-B/2	FM	-	27.57	11.44	0.57	0.63
	PAFM	10	**25.15**	**6.95**	**0.58**	**0.65**
SiT-XL/2	FM	–	<u>11.14</u>	<u>8.25</u>	<u>0.67</u>	**0.66**
	PAFM	10	**9.70**	**5.36**	**0.68**	**0.66**

5.3 Analysis

PAFM induces negligible computational overhead. We benchmark the computational overhead of training with the nearest neighbor version of PAFM compared to FM using the MMDiT architecture on CC12M over 500 consecutive training iterations. PAFM marginally decreases throughput

Table 4: PAFM marginally increases overhead. Computational overhead comparison between FM and PAFM with K=32 neighbors on CC12M 256x256 with MMDiT. Benchmarked on 8×H100 GPUs with total batch size of 256.

Metric	FM	PAFM
Throughput (samples/sec)	1151	1074 (–6.6%)
Peak Memory (GB)	19.11	19.18 (+0.4%)
FLOPs/step (GFLOPs)	10773	10773 (+0.0%)

by 6.6%, only increases peak memory consumption by 0.4% and has no perceived change in GFLOPs. We attribute this to the fact that PAFM nearest neighbors can be computed *once* during data-preprocessing and simply loaded alongside each training example during training. Thus, PAFM can be nearly as efficient as its flow matching counterpart.

PAFM reduces gradient variance in practice. Theorem 2 establishes that under—certain sampling conditions—posterior augmented flow matching (PAFM) lowers the gradient variance at each intermediate point in the latent space, by a factor dependent on the number of additional samples used for supervision compared to flow matching (FM). By lower bounding the FM gradient variance at arbitrary fixed intermediate points, PAFM should further lower bound the variance of the gradients across batches during training, leading to more stable gradient updates. We test this empirically by training two REPA-SiT-B/2 models on ImageNet-1K 256x256 resolution with the FM and nearest neighbor PAFM (K=16) objectives respectively. We measure gradient variance as $\mathrm{Tr}(\Sigma_g) = \mathbb{E}\big[\|g - \hat{g}\|^2\big]$, where Σ_g is the covariance of the stochastic gradient defined in Theorem 2 and $\hat{g}$ is the full-batch gradient. At each training iteration, we randomly sample 500 batches (each of size 256), and compute the variance across these mini-batch gradients. Fig. 3 compares the gradient variance midway through training at 50,000 steps over the same 500 iterations. PAFM achieves lower batch-gradient variance compared to FM *at every iteration*, and its mean variance is nearly 4× less than that of FM: 0.22, versus 0.80. This illustrates how PAFM can reduce gradient variance beyond theoretical settings; it is observed empirically in real-world settings, yielding more stable gradient steps.

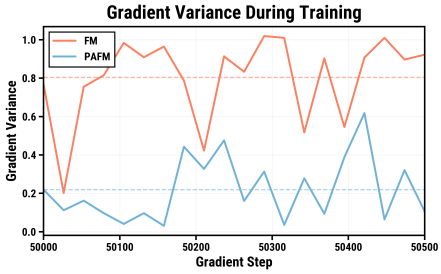

Fig. 3: PAFM reduces mini-batch gradient variance over FM. Optimization steps over the same 500 iterations for REPA-SiT-B/2 [17] models trained with nearest neighbor PAFM with K=16 (blue) and FM (red) on ImageNet-1K [5]. Full lines show gradient variance at each iteration, while the correspondingly colored dashed lines indicate the mean variance across all iterations. PAFM reduces gradient variance by a factor of ∼4.

PAFM improves over FM qualitatively. Fig. 4 shows side-by-side generations using NFE=50 without CFG from REPA-SiT-XL/2 models trained on ImageNet-1K for 400K steps with FM (left) and the VAE sampling variant of PAFM (K=10, right). Across a diverse range of classes, PAFM generations are sharper and more structurally coherent than their FM counterparts. Moreover, while FM generations appear to show difficulty resolving class-discriminative structures, PAFM consistently produces recognizable outputs. These qualitative improvements are consistent with the FID50K gains reported in Tab. 3.

Fig. 4: PAFM improves FM qualitatively. Side-by-side generations with NFE=50 from REPA-SiT-XL/2 models trained for 400K steps on ImageNet-1K using FM (left) and the VAE sampling PAFM (K=10, right) Each pair shows the same class. PAFM consistently produces sharper and more class discriminative outputs.

6 Conclusion

We introduce Posterior Augmented Flow Matching (PAFM), a principled extension of flow matching (FM) that replaces the sparse one-to-one supervision of FM with a posterior-weighted expectation over all plausible trajectories. We show that PAFM is an unbiased estimator of the FM objective that provably lower bounds its gradient variance under certain sampling conditions. PAFM can be flexibly adapted for efficiently training rectified flow models in real-world settings, improving FM performance by up to 3.4 FID50K across class-conditioned and text-to-image benchmarks (ImageNet-1K and CC12M respectively), popular model architectures (SiT and MMDiT), and model scales (SiT-B/2 and SiT-XL/2)—whilst increasing computational overhead by just 6.6%.

Acknowledgments

This work was supported in part by funding from NSF awards #2622839 and #2403297, an NSF GRFP and a Google PhD Fellowship. All views and conclusions expressed in this work are those of the authors and not a reflection of these sources.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: International conference on database theory. Springer (2001)
2. Agrawalla, B., Nauman, M., Agrawal, K., Kumar, A.: floq: Training critics via flow-matching for scaling compute in value-based rl. arXiv (2025)
3. Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E.: Stochastic interpolants: A unifying framework for flows and diffusions. arXiv (2023)
4. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR (2021)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.E., Lomeli, M., Hosseini, L., Jégou, H.: The faiss library. arXiv (2024)
7. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. ICML (2024)
8. Fukumizu, K., Suzuki, T., Isobe, N., Oko, K., Koyama, M.: Flow matching achieves almost minimax optimal convergence. arXiv (2024)
9. Hertrich, J., Chambolle, A., Delon, J.: On the relation between rectified flows and optimal transport. arXiv (2025)
10. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (2020)
12. Ho, J., Salimans, T.: Classifier-free diffusion guidance (2022), `https://arxiv.org/abs/2207.12598`
13. Huang, Y., Transue, T., Wang, S.H., Feldman, W.M., Zhang, H., Wang, B.: Improving flow matching by aligning flow divergence. In: International Conference on Machine Learning (2025)
14. Lee, S., Kim, B., Ye, J.C.: Minimizing trajectory curvature of ODE-based generative models. In: International Conference on Machine Learning (2023)
15. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023), `https://openreview.net/forum?id=PqvMRDCJT9t`
16. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: Advances in Neural Information Processing Systems (2022)
17. Ma, N., Goldstein, M., Albergo, M.S., Boffi, N.M., Vanden-Eijnden, E., Xie, S.: Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In: Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXVII (2024), `https://doi.org/10.1007/978-3-031-72980-5_2`
18. Nash, C., Menick, J., Dieleman, S., Battaglia, P.W.: Generating images with sparse representations. arXiv (2021)

19. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., HAZIZA, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. TMLR (2024)
20. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
21. Reu, T., Dromigny, S., Bronstein, M., Vargas, F.: Gradient variance reveals failure modes in flow-based generative models. arXiv (2025)
22. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
23. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv (2020)
24. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2021)
25. Stoica, G., Ramanujan, V., Fan, X., Farhadi, A., Krishna, R., Hoffman, J.: Contrastive flow matching. ICCV (2025)
26. Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., et al.: Improving and generalizing flow-based generative models with minibatch optimal transport. Transactions on Machine Learning Research (2024)
27. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., Hénaff, O., Harmsen, J., Steiner, A., Zhai, X.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786 (2025)
28. Wiegand, H.: Kish, l.: Survey sampling. john wiley & sons, inc., new york, london 1965, ix + 643 s., 31 abb., 56 tab., preis 83 s. Biometrische Zeitschrift (1968)
29. Yu, S., Kwak, S., Jang, H., Jeong, J., Huang, J., Shin, J., Xie, S.: Representation alignment for generation: Training diffusion transformers is easier than you think (2024), https://arxiv.org/abs/2410.06940