This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Contrastive Flow Matching

George Stoica[12†] Vivek Ramanujan[2◇] Xiang Fan[2◇]
Ali Farhadi[2] Ranjay Krishna[2] Judy Hoffman[1]
[1]Georgia Tech [2]University of Washington
[†]Correspondence to: gstoica3@gatech.edu [◇]Equal Contribution

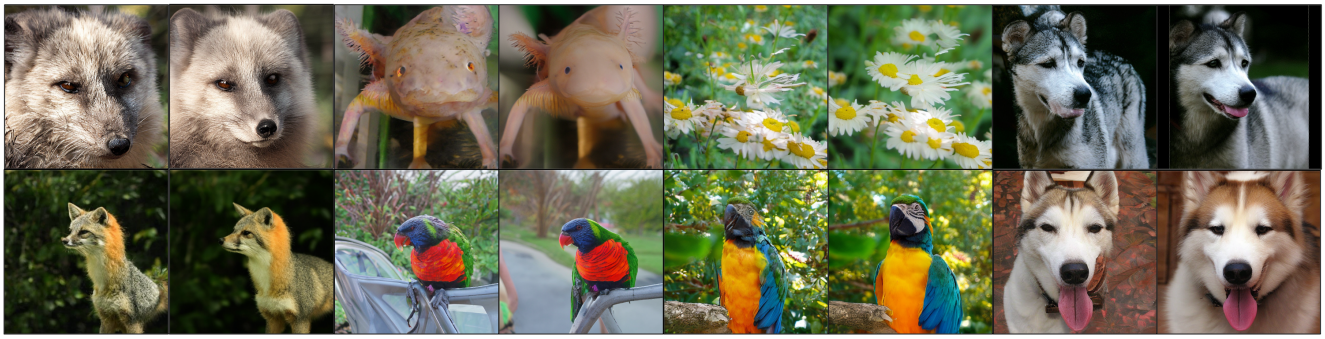

Figure 1. **Training with Contrastive Flow Matching (ΔFM) improves natural image generation.** (**left** is baseline, **right** is with ΔFM) Here we show comparisons between images generated by diffusion models trained on ImageNet-1k (512×512). Each pair of images is generated with the same class and initial noise to ensure similar image structure for comparability. We see that our ΔFM objective encourages significantly more coherent images and improves the consistency of global structure.

Abstract

Unconditional flow matching trains diffusion models to transport samples from a source distribution to a target distribution by enforcing that the flows between sample pairs are unique. However, in conditional settings (e.g., class-conditioned models), this uniqueness is no longer guaranteedflows from different conditions may overlap, leading to more ambiguous generations. We introduce Contrastive Flow Matching, an extension to the flow matching objective that explicitly enforces uniqueness across all conditional flows, enhancing condition separation. Our approach adds a contrastive objective that maximizes dissimilarities between predicted flows from arbitrary sample pairs. We validate Contrastive Flow Matching by conducting extensive experiments across varying model architectures on both class-conditioned (ImageNet-1k) and text-to-image (CC3M) benchmarks. Notably, we find that training models with Contrastive Flow Matching (1) improves training speed by a factor of up to $9\times$, (2) requires up to $5\times$ fewer de-noising steps and (3) lowers FID by up to 8.9 compared to training the same models with flow matching. We release our code at: https://github.com/gstoica27/DeltaFM.git.

1. Introduction

Flow matching for generative modeling trains continuous normalizing flows by regressing ideal probability flow fields between a base (noise) distribution and the data distribution [25]. This approach enables straight-line generative trajectories and has demonstrated competitive image synthesis quality. However, for conditional generation (e.g., class-conditional image generation), vanilla flow matching models often produce outputs that resemble an "average" of the possible images for a given condition, rather than a distinct mode of that condition. In essence, the model may collapse multiple diverse outputs into a single trajectory, yielding samples that lack the expected specificity and diversity for each condition [29, 44]. By contrast, an unconditional flow model—tasked with covering the entire data distribution without any conditioning—implicitly learns more varied flows for different modes of the data. Existing conditional flow matching formulations *do not enforce the flows to differ across conditions*, which can lead to this averaging effect and suboptimal generation fidelity.

To address these limitations and improve generation quality, recent work has explored enhancements to structure the generator's representations and also proposed inference-time guidance strategies. For example, one approach is to incorporate a *REPresentation Alignment* (REPA) objective to

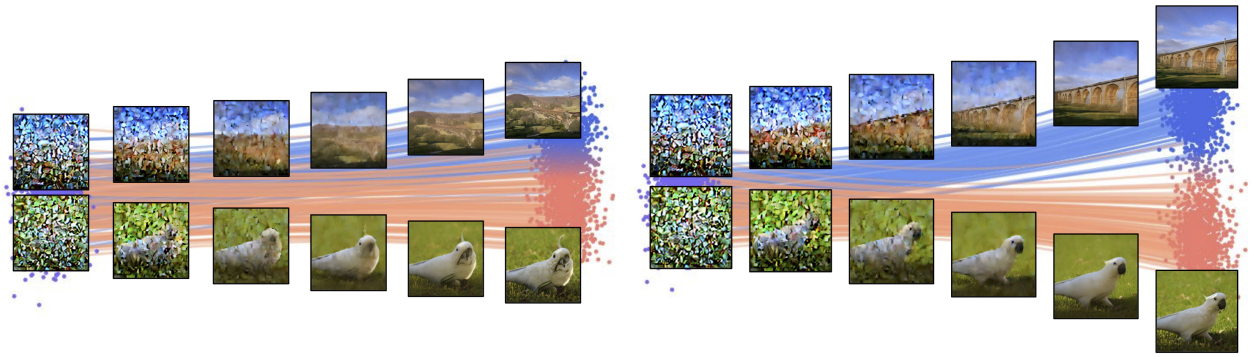

Figure 2. **ΔFM yields more discriminative and higher quality trajectories. (left)** shows the result of standard flow-matching, where flows are straight but end up overlapping for similar class distributions. **(right)** shows how the addition of the ΔFM objective results in more distinct flows, resulting in images which are more representative of their respective classes.

structure the representations at an intermediate layer with those from a high-quality pretrained vision encoder [44]. By using feature embeddings from a DINO self-supervised vision transformer [5, 31], the generative model's hidden states are guided toward semantically meaningful directions. This representational alignment provides an additional learning signal that has been shown to improve both training convergence and final image fidelity, albeit at the cost of requiring an external pretrained encoder and an auxiliary loss term. Another popular technique is *classifier-free guidance* (CFG) for conditional generation [18], which involves jointly training the model in unconditional and conditional modes (often by randomly dropping the condition during training). At inference time, CFG performs two forward passes—one with the conditioning input and one without—and then extrapolates between the two outputs to push the sample closer to the conditional target [18, 29]. While CFG can significantly enhance image detail and adherence to the prompt or class label, it doubles the sampling cost and complicates training by necessitating an implicit unconditional generator alongside the conditional ones [11, 11, 20].

We propose **Contrastive Flow Matching** (ΔFM), a new approach that augments the flow matching objective with an auxiliary contrastive learning objective. ΔFM encourages more diverse and distinct conditional generations. It applies a contrastive loss on the flow vectors (or representations) of samples within each training batch, encouraging the model to produce dissimilar flows for different conditioning inputs. Intuitively, this loss penalizes the model if two samples with different conditions yield similar flow dynamics, thereby explicitly discouraging the collapse of multiple conditions onto a single "average" generative trajectory. As a result, given a particular condition, the model learns to generate a unique flow through latent space that is characteristic of that condition alone, leading to more varied and condition-specific outputs. Importantly, this contrastive augmentation is *complementary* to existing methods. It can be applied along with REPA, further ensuring that flows not only align with pretrained features but also remain distinct across conditions. Likewise, it is compatible with classifier-free guidance at sampling time, allowing one to combine its benefits with CFG for even stronger conditional signal amplification.

Inspired by contrastive training objectives, ΔFM applies a pairwise loss term between samples in a training batch: for each positive sample from the batch, we randomly sample a negative counterpart. We then encourage the model to not only learn the flow towards the positive sample but also to learn the flow away from the negative sample. This is achieved by adding a contrastive loss to the flow matching objective, which promotes class separability throughout the flow. Our method is simple to implement and can be easily integrated into existing diffusion models without any additional data and with minimal computational overhead.

We validate the advantages of ΔFM through (1) extensive experiments on conditional image generation using ImageNet images across multiple SiT [29] model scales and training frameworks [29, 44], and (2) text-to-image experiments on the CC3M [37] with the MMDiT [14] architecture. Thanks to contrastive flows, ΔFM consistently outperforms traditional diffusion flow matching in quality and diversity metrics, achieving up to an 8.9-point reduction in FID-50K on ImageNet, and 5-point reduction in FID on the whole CC3M validation set. It is also compatible with recent significant improvements in the diffusion objective, such as Representation Alignment (REPA) [44]. By encouraging class separability, ΔFM is able to efficiently reach a given image quality with $5\times$ **fewer** sampling steps than a baseline Flow Matching model, translating directly to faster generation. It also enhances training efficiency by up to **9**$\times$. Finally, ΔFM stacks with classifier-free guidance, lowering FID by **5.7%** compared to flow matching models.

2. Related works

Our work lies in the domain of image generative models, primarily diffusion and flow matching models. We augment flow matching with a contrastive learning objective to provide an alternative solution to classifier free guidance.

Generative modeling has rapidly advanced through two primary paradigms: diffusion-based methods [19, 39] and flow matching [25]. **Denoising diffusion models** typically rely on stochastic differential equations (SDEs) and score-based learning to iteratively add and remove noise [19]. Denoising diffusion implicit models (DDIMs) [39] reduce this sampling complexity by removing non-determinism in the reverse process, while progressive distillation [34] further accelerates inference by shortening the denoising chain. Advanced ODE solvers [6] and distillation methods [41] have also enhanced sampling efficiency. Despite their success, diffusion models can be slow at inference due to iterative denoising [19].

Flow matching [6] has been designed to reduce inference steps. It directly parameterizes continuous-time transport dynamics for more efficient sampling. Probability flow ODEs [25, 39] learn an explicit transport map between data and latent distributions. Unlike diffusion models, it bypasses separate score estimation and stochastic noise, which reduces function evaluations and tends to improve training convergence [6]. A common type of flow matching algorithm popularized recently is the rectified flow [26], which refines probability flow ODEs through direct optimal transport learning, improving numerical stability and sampling speed. This approach mitigates the high computational burden of diffusion sampling while maintaining high-fidelity image generation with fewer integration steps.

Since both diffusion and flow matching models are trained to match the target distribution of real images, they often produce 'averaged' samples that lack the sharp details and strong conditional fidelity [17]. Regardless of how much these models speed up, they often need to be invoked multiple times with unique seed noise to find a high-fidelity sample. In response, **guidance techniques** have been introduced to substantially promote high-fidelity synthesis. Classifier guidance [12], classifier-free guidance [17], energy guidance [8, 27, 40, 45], and more advanced methods [9, 20, 21, 23, 38] improve fidelity and controllability, without requiring multiple invocations. Although they achieve remarkable performance, they typically still require additional computational overhead. CFG requires calling sampling from a second 'unconditional' generation and guiding the 'conditional' generation away from the unconditional variant [28, 42, 43, 46]. We adapt the flow matching objective with a contrastive loss between the transport vectors within a batch. By doing so, we achieve the same benefits of CFG, without the additional overhead of needing to train an unconditional generator or using one during inference.

Contrastive learning was originally proposed for face recognition [36], where it was designed to encourage a margin between positive and negative face pairs. In generative adversarial networks (GANs), it has been applied to improve sample quality by structuring latent representations [4]. However, to the best of our knowledge, it has not been explored in the context of visual diffusion or flow matching models. We incorporate this contrastive objective to demonstrate its utility in speeding up training and inference of flow-based generative models.

3. Background and motivation

We focus on flow matching models [25] due to its rising popularity as an effective training paradigm for generative models [1, 2, 24]. In this section, we provide a brief overview of flow matching through the perspective of stochastic interpolants [2, 29], as it pertains to our work.

Preliminaries. Let $p(x)$ be an arbitrary distribution defined on the reals, and let $\mathcal{N}(0, \mathrm{I})$ be a Gaussian noise distribution. The objective of flow matching is to learn a transport between the two distributions. That is, given an arbitrary $\epsilon \sim \mathcal{N}(0, \mathrm{I})$, a flow matching model gradually transforms ϵ over time into an $\hat{x}$ that is part of $p(x)$. Stochastic interpolants [2, 29] define this transformation as a time-dependent stochastic process, where transformation steps are summarized as follows,

$$\hat{x}_t = \alpha_t \hat{x} + \sigma_t \epsilon \tag{1}$$

where α_t and σ_t are decreasing and increasing time-dependent functions respectively defined on $t \in [0, T]$, such that $\alpha_T = \sigma_0 = 1$ and $\alpha_0 = \sigma_T = 0$. While theoretically, α_t, σ_t need not be linear, linear complexity is often sufficient to obtain strong diffusion models [25, 29, 44].

Flow matching. Given such a process, flow matching models learn to transport between noise to $p(x)$ by estimating a velocity field over an probability flow ordinary differential equation (PF ODE), $dx_t = v(x_t, t)dt$, whose distribution at time t is the marginal $p_t(x)$. This velocity is given by the expectations of $\hat{x}$ and ϵ conditioned on x_t,

$$v(x_t, t) = \dot{\alpha}_t \mathbb{E}[\hat{x} | x_t = x] + \dot{\sigma}_t \mathbb{E}[\epsilon | x_t = x], \tag{2}$$

where $\dot{\alpha}_t, \dot{\sigma}_t$ are the time-based derivatives of α_t and σ_t respectively. Since, $\hat{x}$ and ϵ are arbitrary samples from their respective distributions, $v(x_t, t)$ is expected "direction" of all transport paths between noise and $p(x)$ that pass through x_t at t. While the optimal $v(x_t, t)$ is intractable, it can be approximated with a flow-model $v_\theta(x_t, t)$, by minimizing the training objective:

$$\mathcal{L}^{(FM)}(\theta) = \mathbb{E}\left[||v_\theta(x_t, t) - (\dot{\alpha}_t \hat{x} + \dot{\sigma}_t \epsilon)||^2\right] \tag{3}$$

Key to understanding the properties of flow matching is the concept of flow uniqueness [25]. That is, flows following the well-defined ODE cannot intersect at *any* time

$t \in [0, T)$. As such, flow models can iteratively refine unique-discriminative features relevant to any $x \sim p(x)$ in each x_t, leading to more efficient and accurate diffusion paths compared to other training paradigms [25].

Conditional flow matching. Commonly, $p(x)$ may be a marginal distribution over several class-conditional distributions (e.g., the classes of ImageNet [33]). Training models in such cases is nearly identical to standard flow matching, except that flows are further conditioned on the target distribution class:

$$\mathcal{L}_{cond}^{(FM)}(\theta) = \mathbb{E}\left[||v_\theta(x_t, t, y) - (\dot{\alpha}_t \hat{x} + \dot{\sigma}_t \epsilon)||^2\right], \quad (4)$$

where $\hat{x} \sim p(x|y)$. Resultant models have the desirable trait of being more controllable: their generated outputs can be tailored to their respective input conditions. However, this comes at the notable cost of flow-uniqueness. Specifically these models only generate unique flows compared to others *within* the same class-condition, not necessarily *across* classes. This inhibits x_t's from storing important class-specific features and leads to poorer quality generations. Second, the conditional flow matching objective trains models without knowledge of the distributional spread from other class-conditions, leading to flows that may generate ambiguous outputs when conditional distributions overlap. This increases the likelihood of ambiguous generations that form a mixture between different conditions, restricting model capabilities. We study these effects in Section 5.

4. Contrastive Flow Matching

We introduce Contrastive Flow Matching (ΔFM), a novel approach designed to address the challenges of learning efficient class-distinct flow representations in conditional generative models. Standard conditional flow matching (FM) models tend to produce flow trajectories that align across different samples, leading to reduced class separability. ΔFM extends the FM objective by incorporating a contrastive regularization term, which explicitly discourages alignment between the learned flow trajectories of distinct samples.

Ingredients. Let $\tilde{x} \sim p(x|\tilde{y})$ denote a sample drawn from the data distribution conditioned on an arbitrary class $\tilde{y}$, and let $\tilde{\epsilon} \sim \mathcal{N}(0, I)$ represent an independent noise sample. To ensure that the contrastive objective captures distinct flow trajectories, we impose the conditions $\tilde{x} \neq \hat{x}$ and $\tilde{\epsilon} \neq \epsilon$, where $\tilde{y}$ may or may not be equal to y. Importantly, we do not assume the existence of a time step $t \in [0, T]$ such that $x_t = \alpha_t \tilde{x} + \sigma_t \tilde{\epsilon}$. Consequently, $\tilde{x}$ and $\tilde{\epsilon}$ represent truly independent flow trajectories in comparison to $\hat{x}$ and ϵ.

The contrastive regularization. Given $v_\theta(x_t, t, y)$ and an arbitrary $\tilde{x}, \tilde{\epsilon}$ sample pair, the contrastive objective aims to *maximize* the dissimilarity between the estimated flow of $v_\theta(x_t, t, y)$ from ϵ to $\hat{x}$, and the independent flow produced by $\tilde{x}, \tilde{\epsilon}$. We achieve this by maximizing the quantity,

$$E\left[||v_\theta(x_t, t, y) - (\dot{\alpha}_t \tilde{x} + \dot{\sigma}_t \tilde{\epsilon})||^2\right]. \quad (5)$$

Since $\tilde{x}$ is drawn from the marginal $p(x)$ rather than $p(x|y)$, Equation 5 trains flow matching models to produce flows that are *unconditionally* unique.

Putting it all together. We now define contrastive flow matching as follows,

$$\mathcal{L}^{(\Delta\text{FM})}(\theta) = \mathrm{E}\begin{bmatrix} ||v_\theta(x_t, t, y) - (\dot{\alpha}_t \hat{x} + \dot{\sigma}_t \epsilon)||^2 \\ -\lambda ||v_\theta(x_t, t, y) - (\dot{\alpha}_t \tilde{x} + \dot{\sigma}_t \tilde{\epsilon})||^2 \end{bmatrix} \quad (6)$$

where $\lambda \in [0, 1)$ is a fixed hyperparameter that controls the strength of the contrastive regularization. Thus, ΔFM simultaneously encourages flow matching models to estimate effective transports from noise to corresponding class-conditional distributions (the flow matching objective), while enforcing each to be discriminative *across* classes (contrastive regularization). Note that ΔFM can be thought of as a generalization of flow matching, as ΔFM reduces to FM when $\lambda = 0$. We study the effects of varying λ in Section 5.5.

Implementation. Contrastive flow matching (ΔFM) is easily integrated into any flow matching training loop, with minimal overhead. Algorithm 1 illustrates the implementation of an arbitrary batch step, where navy text marks additions to the standard flow matching objective. Thus, ΔFM solely depends on the information already available to the flow matching objective at each batch step, without computing any additional forward steps. Furthermore, ΔFM seamlessly folds into flow matching training regimes, making it a "plug-and-play" objective for existing setups.

Algorithm 1 Contrastive Flow Matching Batch Step

Input: A model v_θ, batch of N flow examples $F = \{(x_1, y_1, \epsilon_1), \ldots, (x_N, y_N, \epsilon_N)\}$ where $(x_i, y_i) \sim p(x, y)$ and $\epsilon_i \sim \mathcal{N}(0, \mathrm{I})$, β learning rate, $\lambda = 0.05$.
Output: Updated model parameters θ
$L(\theta) = 0$
for i in range(N) **do**
  $t \sim U(0, 1), x_t = \alpha_t x_i + \sigma_t \epsilon_i$
  sample $(\tilde{x}, \tilde{y}, \tilde{\epsilon}) \sim F$, s.t. $(\tilde{x}, \tilde{y}, \tilde{\epsilon}) \neq (x_i, y_i, \epsilon_i)$
  $\hat{v} = v(x_t, t, y_i), v = \dot{\alpha}_t x_i + \dot{\sigma}_t \epsilon, \tilde{v} = \dot{\alpha}_t \tilde{x} + \dot{\sigma}_t \tilde{\epsilon}$
  $L(\theta) += ||\hat{v} - v||^2 - \lambda ||\hat{v} - \tilde{v}||^2$
end for
$\theta \leftarrow \theta - \frac{\beta}{N} \nabla_\theta L(\theta)$

Discussion. Figure 3 illustrates the effects of contrastive flow matching compared to flow matching. The figure shows the resultant flows after training a small diffusion model in a simple toy-setting. Specifically, we create a two-dimensional

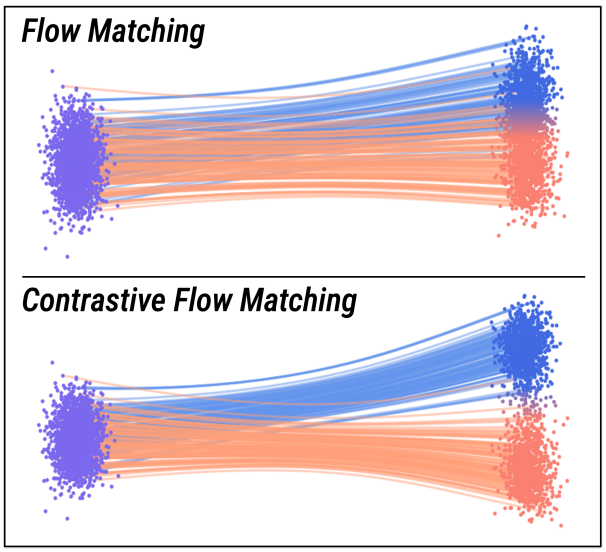

Figure 3. **Contrastive Flow Matching intrinsically separates flows between classes.** We train a small three layer MLP flow matching model to transport between a two dimensional multivariate noise distribution (violet) and two independent blue and orange class distributions respectively. The class distributions are designed to have $\sim 50\%$ overlap, and we plot the learned class-conditioned flows between noise samples and each respective class distribution using class colors. Top: Flow matching models learn overlapping transports between distributions, generating outputs that lie in ambiguous regions between the two classes. Bottom: Contrastive flow matching models have significantly more discriminative flows, generating class-coherent samples while reducing ambiguity.

violet gaussian noise distribution and two independent two-dimensional class distributions (in blue and orange respectively) such that the latter distributions have $\approx 50\%$ overlap. Samples from each distribution are represented as "dots", with those in the target distributions colored according to the gaussian *kernel-density estimate* between samples from each class in their respective region. We observe that training the model with flow matching (top) create flows with large degrees of overlap between classes, generating samples with lower class-distinction. In contrast, training the same model with contrastive flow matching (bottom) yields trajectories that are significantly more diverse across classes, while also generating samples which capture distinct features of each respective class.

5. Experiments

We validate contrastive flow matching (ΔFM) through extensive experiments across various model, training and benchmark configurations. Overall, models trained with ΔFM consistently outperform flow matching (FM) models across *all* settings.

Datasets. We conduct both class-conditioned and text-to-image experiments. We use ImageNet-1k [10] processed at both (256×256) and (512×512) resolutions for our class-conditioned experiments, and follow the data preprocessing procedure of ADM [12] We then follow [44] and encode each image using the Stable Diffusion VAE [32] into a tensor $z \in \mathbb{R}^{32\times32\times4}$. For text-to-image (t2i), we use the Conceptual Captions 3M (CC3M) dataset [37] processed at (256×256) resolution and follow the data processing procedure of [3]. We train all models by strictly following the setup in [44], and use a batch size of 256 unless otherwise specified. We do not alter the training conditions to be favorable to ΔFM, and we always set $\lambda = 0.05$ when applicable.

Measurements. We report five quantitative metrics throughout our experiments. We report Fréchet inception distance (FID) [16], inception score (IS) [35], sFID [30], precision (Prec.) and recall (Rec.) [22] using 50,000 samples for our class-conditioned experiments. Similarly, we report FID over the whole validation set in the text-to-image setting. We use the SDE Euler-Maruyama sampler with $w_t = \sigma_t$ for all experiments, and set the number of function evaluations (NFE) to 50 unless otherwise specified.

5.1. Contrastive Flow Matching Improves SiT

Implementation details. We train on the state-of-the-art SiT [29] model architecture, at both B/2 and XL/2 scales.

	Metrics				
Model	FID ↓	IS ↑	sFID ↓	Prec. ↑	Rec. ↑
SiT-B/2	42.28	38.04	11.35	0.5	0.62
+ Using ΔFM	**33.39**	**43.44**	**5.67**	**0.53**	**0.63**
SiT-XL/2	20.01	74.15	8.45	0.63	**0.63**
+ Using ΔFM	**16.32**	**78.07**	**5.08**	**0.66**	**0.63**

(a) **ImageNet-1k (256x256) Results.** ΔFM significantly outperforms flow-matching models across nearly all metrics, and matches Recall on SiT-XL/2.

	Metrics				
Model	FID↓	IS↑	sFID↓	Prec.↑	Rec.↑
SiT-B/2	50.26	33.58	14.88	0.57	0.61
+ Using ΔFM	**41.59**	**38.20**	**6.13**	**0.62**	**0.63**
SiT-XL/2	22.98	70.14	10.71	0.73	**0.60**
+ Using ΔFM	**19.67**	**72.58**	**4.98**	**0.76**	**0.60**

(b) **ImageNet-1k (512x512) Results.** Models trained with ΔFM either substantially outperform or match their flow-matching counterparts.

Table 1. SiT [29] results on ImageNet-1k (256×256; a) and (512×512; b). We train all models for 400K iterations following [44]. All metrics are measured with the SDE Euler-Maruyama sampler with NFE=50 and without classifier guidance. We use $\lambda = 0.05$ for all models trained with ΔFM and do not change any other hyperparameters. ↑ indicates that higher values are better, with ↓ denoting the opposite.

Results. Table 1 summarizes our results. Overall, ΔFM dramatically improves over flow matching in nearly all met-

rics (only matching the flow matching SiT-XL/2 model in recall). Notably, employing ΔFM with SiT-B/2 lowers FID by over 8 compared to flow matching at both ImageNet resolutions, highlighting the strength of ΔFM in smaller model scales. Similarly, ΔFM is robust to larger model scales and outperforms FM by over 3.2 FID when using SiT-XL/2.

5.2. REPA is complementary

REPresentation Alignment (REPA) [44] is a recently introduced training framework that rapidly improves diffusion model performance by strengthening its intermediate representations. Specifically, REPA distills the encodings of foundation vision encoders (e.g., DiNOv2 [5]) into the hidden states of diffusion models through the use of an auxilliary objective. Notably, REPA can improve the training speed of vanilla SiT models by over $17.5\times$, while further improving their performances [44]. ΔFM is easily integrated into REPA and only requires replacing the flow matching objective.

Implementation details. We apply REPA on the same SiT models in Section 5.1, and use the distillation process defined by [44] exactly. Specifically, we distill DiNOv2 [5] ViT-B [13] features into the 4th layer of the SiT-B/2, and the 8th layer of the SiT-XL/2, and use their hyperparameter setup.

	Metrics				
Model	FID ↓	IS ↑	sFID ↓	Prec. ↑	Rec. ↑
REPA SiT-B/2	27.33	61.60	11.70	0.57	**0.64**
+ Using ΔFM	**20.52**	**69.71**	**5.47**	**0.61**	0.63
REPA SiT-XL/2	11.14	115.83	8.25	0.67	**0.65**
+ Using ΔFM	**7.29**	**129.89**	**4.93**	**0.71**	0.64

(a) **ImageNet-1k (256x256) Results with REPA.** Adding ΔFM to REPA further improves SiT models across nearly all metrics.

	Metrics				
Model	FID↓	IS↑	sFID↓	Prec.↑	Rec.↑
REPA SiT-B/2	31.90	56.96	13.78	0.67	0.62
+ Using ΔFM	**24.48**	**64.74**	**5.89**	**0.71**	0.61
REPA SiT-XL/2	11.32	119.72	10.21	0.76	**0.63**
+ Using ΔFM	**7.64**	**131.50**	**4.72**	**0.79**	0.62

(b) **ImageNet-1k (512x512) Results with REPA.** ΔFM is robust with REPA at large image resolutions, further improving performance across established metrics.

Table 2. REPA SiT [29] results on ImageNet-1k (256×256; a) and (512×512; b). All models are trained for 400K iterations strictly following the procedure in [44], and set $\lambda = 0.05$. We use the SDE Euler-Maruyama sampler with NFE=50 without classifier guidance for all our metrics.

Results. We report results in Table 2. Similar to Section 5.1, ΔFM substantially improves REPA models by as much as 6.81 FID, and consistently improves flow matching with model scale. This highlights the versatility of the contrastive flow matching objective as a broadly applicable criterion for diffusion model.

5.3. Extending to text-to-image generation

Implementation Details. We train models with the popular MMDiT [14] architecture from scratch on the CC3M dataset [37] for 400K iterations. For faster training, we pair each model with REPA, and follow the recommended training protocol of [44].

Metric	REPA-MMDiT	
	Flow-Matching	ΔFM
FID↓	24	**19**

Table 3. **ΔFM improves on CC3M 256×256.** We use the SDE Euler-Maruyama sampler and NFE=50 without classifier-free guidance.

Results. Table 3 shows our results. ΔFM improves over the flow matching baseline by 5 FID, highlighting its seamless transferability to the broader text-to-image setting. We show qualitative results in Appendix A.

5.4. CFG stacks with contrastive flow matching

Contrastive flow matching offers advantages of Classifier-Free Guidance (CFG), without incurring additional computational costs during inference. In this section, we demonstrate that when computational resources permit, combining ΔFM with CFG can yield further performance enhancements.

Accounting for conflicts. CFG and ΔFM encourage flow matching model generations to be unique and identifiable, in different ways. Specifically, ΔFM trains models whose conditional flows are steered away from other arbitrary flows in the training data, regardless of generation state (x_t). In contrast, CFG steers generations away from the unconditional flow estimates based on x_t. Thus, the signals from each may not always be aligned and naively coupling them may lead to conflicts and suboptimal generations. Fortunately, we can quantify the amount of steerage ΔFM applies on flow matching models by deriving the closed-form solution to Eq. 4: $\min_\theta \mathcal{L}^{(\Delta\text{FM})}(\theta) = \left[(\min_\theta \mathcal{L}^{(\text{FM})}(\theta)) - \lambda\hat{T}\right]/[1-\lambda]$, where $\hat{T}$ is simply the *mean* of all sample trajectories from the training set (please see Appendix B.1 for the full derivation). Thus, ΔFM yields models which estimate flows away from the *data-driven* unconditional trajectory, weighted by λ. While optimizer and training dynamics cannot guarantee that all models trained with ΔFM exactly decompose into these terms, $\hat{T}$ nevertheless approximates its effect on these models. With $\hat{T}$, we can account for conflicts between ΔFM and CFG by modifying the CFG equation to: $\hat{\text{CFG}} = (1-\lambda)\left[wv(x_t|y) + (1-w)v(x_t|\emptyset)\right] + \lambda\tau$, where w is the guidance scale, $\emptyset$ is the unconditional term and λ is the same parameter used during ΔFM training (Appendix B.2 contains the full derivation). Note that, we only apply

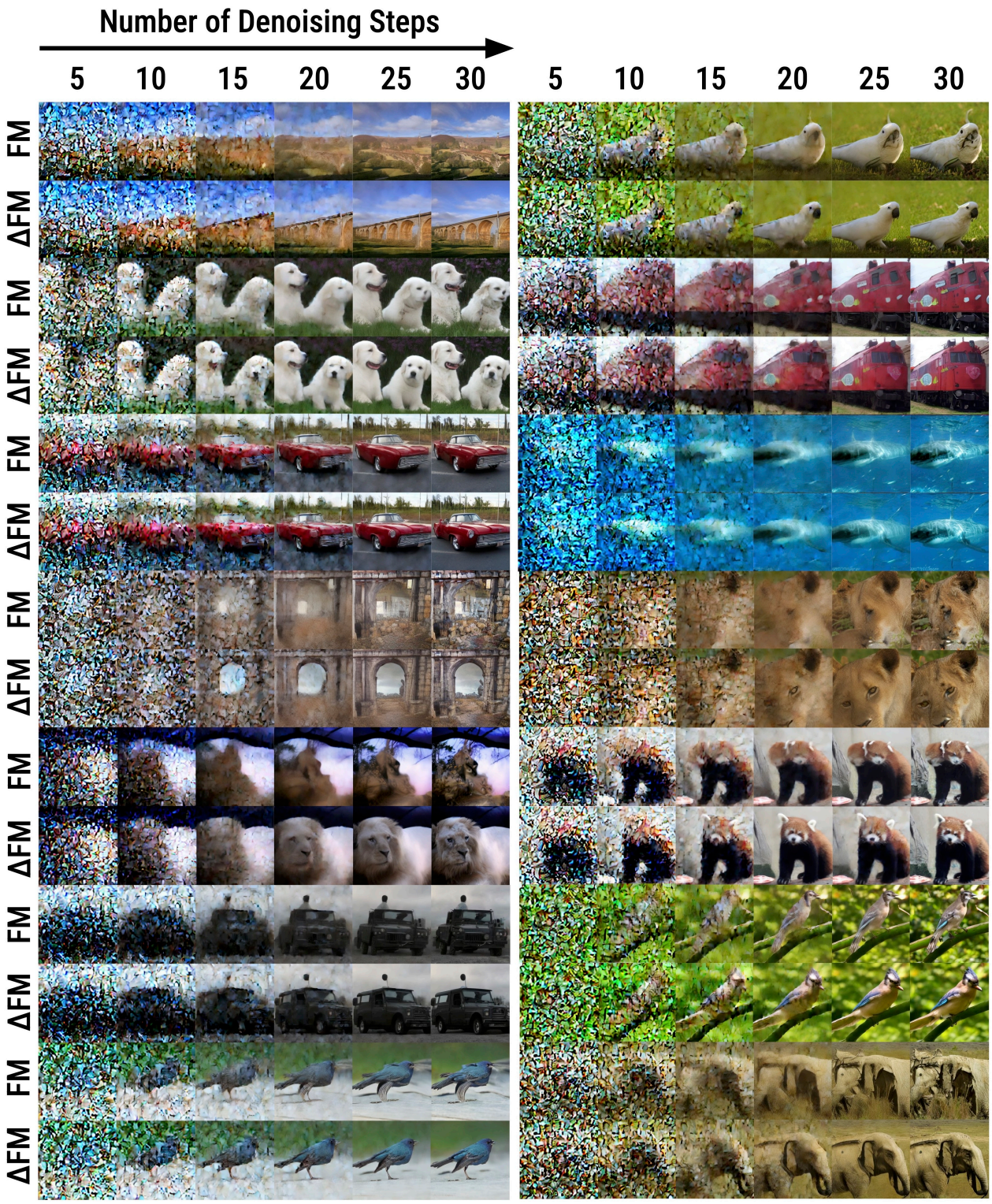

Figure 4. **Contrastive flow matching (ΔFM) denoises significantly more efficiently than flow matching.** We visualize the expected final image estimated by a flow-model when denoised every 5 steps for trajectories of length 30 steps using the SDE Euler-Maruyama sampler and do not use classifier guidance. We compare the trajectories of a REPA SiT-XL/2 [44] trained on ImageNet-256 [10] for 400K steps with flow matching (FM), and the same model trained with the contrastive flow matching (ΔFM) objective. We show these trajectories in sets of pairs generated from the same noise sample during inference, with the flow matching model above our ΔFM version.

Model	CFG Terms			Metric		
	w	σ_{low}	σ_{high}	IS↑	FID↓	sFID↓
REPA SiT-XL/2	1.75	0.0	0.75	280.33	2.09	5.55
+ Using ΔFM	1.85	0.0	0.65	**281.95**	**1.97**	**4.49**

Table 4. **ImageNet 256×256 Results with CFG and NFE=50.** "w" denotes the classifier-free guidance (CFG) weight, and $[\sigma_{\text{low}}, \sigma_{\text{high}}]$ is the time interval under which CFG is applied. We report the best results for each model after conducting a grid search over $w \in \{1.25, 1.75, 1.8, 1.85, 2.25\}$, $\sigma_{\text{low}} = 0$ and $\sigma_{\text{high}} \in \{0.50, 0.65, 0.75, 1.0\}$. ΔFM outperforms FM on all metrics.

Metric	ΔFM λ Values					
	0.0	0.001	0.01	0.05	0.1	0.15
IS↑	115.83	115.70	119.41	**129.89**	116.27	82.20
FID↓	11.14	10.93	9.93	**7.29**	9.86	19.21

Table 5. $\lambda = 0.05$ **is ideal.** We show an ablation of the ΔFM weight parameter λ. A too large λ produces degenerate distributions that do not model class structure well. Too low λ is essentially identical to flow-matching, with very little effect on training. $\lambda = 0.05$ is best and we use this for all our experiments.

CFG within the specified guidance interval $[\sigma_{\text{low}}, \sigma_{\text{high}}]$, and use our *unchanged* ΔFM model outside this interval. Appendix B.3 describes other ways to combine CFG with ΔFM, which we leave to future work.

Results. Table 4 summarizes the results. When paired with CFG, ΔFM improves flow matching models across all metrics, demonstrating its efficacy in settings where computational costs are not a constraint.

5.5. Analyzing Contrastive Flow Matching

Understanding the ΔFM weight (λ). λ directly controls how unique flows are across classes. Increasing λ encourages every diffusion step to be fully discriminative, enabling models to encode distinct representations that integral to generating strong visual outputs at each trajectory step. However, setting it too high can lead to overly-separated flow trajectories, making it difficult to capture the class structure (Table 5.5). However, λ values that are too low mirror the flow matching objective. Notably, we find that $\lambda = 0.05$ is stable across all model and dataset settings, consistently achieving strong performance.

Earlier class differentiation during denoising. In Figure 4, we study flow trajectories of standard flow matching (FM) and flow matching with ΔFM. To do this, we take partially denoised latents at various intermediate time steps along a trajectory with total length 30. While initially both follow similar trajectories, they quickly diverge within the first several steps of the denoising process. For instance, the model trained with ΔFM produces more structurally coherent images earlier (around 15 to 20 steps in) than with FM. The iconic features of each class, such as slanted bridge surfaces (Figure 4 (top-left)), animal eyes (Figure 4 (upper-left and top-right), and train windows (Figure 4 (upper-right)), are more clearly visible early on during the diffusion process of the ΔFM model. This enables ΔFM to ultimately generate higher quality images at the final timestep.

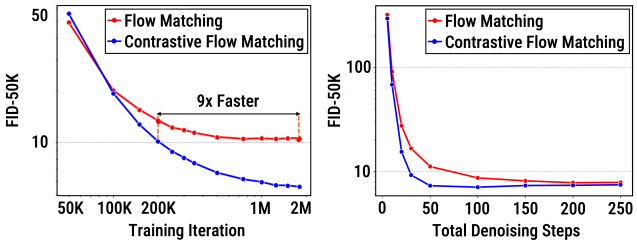

Figure 5. **ΔFM requires significantly fewer training iterations and inference-time denoising steps.** We plot FID-50k on ImageNet 256x256 with different numbers of training iterations and denoising steps. We see that ΔFM outperforms the baseline with **9× fewer** training iterations and **5× reduction** in the number of inference-time denoising steps, indicating that ΔFM is more efficient in both training and inference.

Improved training and inference speed. In Figure 5 (left), we see the significant improvements in training speed from the ΔFM objective. We reach the same performance (measured by FID-50k) as baseline with 9× fewer training iterations. In Figure 5 (right), we also demonstrate significant improvements at inference time. With our objective, we reach superior performance with only 50 denoising steps compared to the baseline with 250 denoising steps. This is a linear 5× improvement in training efficiency. Taken together, these results emphasize the important gains in computational efficiency achieved by our method.

Effects of batch size on ΔFM. We study the effects of batch size on our loss in Appendix C. We find that ΔFM improvements over the REPA baseline across all batch sizes.

6. Conclusion

We introduced Contrastive Flow Matching (ΔFM), a simple addition to the diffusion objective that enforces distinct, diverse flows during image generation. Quantitatively, ΔFM results in improved image quality with far fewer denoising steps (5× faster) and significantly improved training speed (9× faster). Qualitatively, ΔFM improves the structural coherence and global semantics for image generation. All of this is achieved with negligible extra compute per training iteration. Finally, we show that our improvements stack with the recently proposed Representation Alignment (REPA) loss, allowing for strong gains in image generation performance. Looking forward, ΔFM shows the possibility that deviating from perfect distribution modeling in the diffusion objective might result in better image generation.

7. Acknowledgments

This work was supported in part by NSF #2144194 and #2403297, as well as a NSF-GRFP. It was also supported in part by NSF IIS 1652052, IIS 1703166, DARPA N66001-19-2-4031, DARPA W911NF-15-1-0543 and gifts from Allen Institute for Artificial Intelligence, Google and Apple. All views and conclusions expressed in this work are those of the authors and not a reflection of these sources.

References

[1] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. 3

[2] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 3

[3] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023. 5

[4] Gongze Cao, Yezhou Yang, Jie Lei, Cheng Jin, Yang Liu, and Mingli Song. Tripletgan: Training generative model with triplet loss, 2017. 3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 6, 11

[6] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018. 3

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICLR*, 2020. 11

[8] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 3

[9] Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. *arXiv preprint arXiv:2406.08070*, 2024. 3

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5, 7

[11] Alakh Desai and Nuno Vasconcelos. Improving image synthesis with diffusion-negative sampling, 2024. 2

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3, 5

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 6

[14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *ICML*, 2024. 2, 6

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *CVPR*, 2020. 11

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[17] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 2

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 3

[20] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *arXiv preprint arXiv:2406.02507*, 2024. 2, 3

[21] Felix Koulischer, Johannes Deleu, Gabriel Raya, Thomas Demeester, and Luca Ambrogioni. Dynamic negative guidance of diffusion models: Towards immediate content removal. In *Neurips Safe Generative AI Workshop 2024*. 3

[22] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 2019. 5

[23] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024. 3

[24] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 3

[25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023. 1, 3, 4

[26] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. 3

[27] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning*, pages 22825–22855. PMLR, 2023. 3

[28] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 3

[29] Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. 2024. 1, 2, 3, 5, 6

[30] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 5

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 2

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 5

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. 4

[34] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 3

[35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 5

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 815823. IEEE, 2015. 3

[37] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 2, 5, 6

[38] Rahul Shenoy, Zhihong Pan, Kaushik Balakrishnan, Qisen Cheng, Yongmoon Jeon, Heejune Yang, and Jaewon Kim. Gradient-free classifier guidance for diffusion model sampling. *arXiv preprint arXiv:2411.15393*, 2024. 3

[39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3

[40] Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pages 32483–32498. PMLR, 2023. 3

[41] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space, 2021. 3

[42] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024. 3

[43] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024. 3

[44] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think, 2024. 1, 2, 3, 5, 6, 7, 11

[45] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. Egsde: Unpaired image-to-image translation via energy-guided stochastic differential equations. *Advances in Neural Information Processing Systems*, 35:3609–3623, 2022. 3

[46] Mingyuan Zhou, Zhendong Wang, Huangjie Zheng, and Hai Huang. Long and short guidance in score identity distillation for one-step text-to-image generation. *arXiv preprint arXiv:2406.01561*, 2024. 3