This CVPR 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.

MeanFlow Transformers with Representation Autoencoders

Zheyuan Hu[1*] Chieh-Hsin Lai[1] Ge Wu[3] Yuki Mitsufuji[1,2] Stefano Ermon[4]
[1]Sony AI [2]Sony Group Corporation [3]Nankai University [4]Stanford University
zyhu2001@gmail.com chieh-hsin.lai@sony.com

Abstract

MeanFlow (MF) is a diffusion-motivated generative model that enables efficient few-step generation by learning long jumps directly from noise to data. In practice, it is often used as a latent MF by leveraging the pre-trained Stable Diffusion variational autoencoder (SD-VAE) for high-dimensional data modeling. However, MF training remains computationally demanding and is often unstable. During inference, the SD-VAE decoder dominates the generation cost, and MF depends on complex guidance hyperparameters for class-conditional generation. In this work, we develop an efficient training and sampling scheme for MF in the latent space of a Representation Autoencoder (RAE), where a pre-trained vision encoder (e.g., DINO) provides semantically rich latents paired with a lightweight decoder. We observe that naive MF training in the RAE latent space suffers from severe gradient explosion. To stabilize and accelerate training, we adopt Consistency Mid-Training for trajectory-aware initialization and use a two-stage scheme: distillation from a pre-trained flow matching teacher to speed convergence and reduce variance, followed by an optional bootstrapping stage with a one-point velocity estimator to further reduce deviation from the oracle mean flow. This design removes the need for guidance, simplifies training configurations, and reduces computation in both training and sampling. Empirically, our method achieves a 1-step FID of 2.03, outperforming vanilla MF's 3.43, while reducing sampling GFLOPS by 38% and total training cost by 83% on ImageNet 256. We further scale our approach to ImageNet 512, achieving a competitive one-step FID of 3.23 with the lowest GFLOPS among all baselines. Code is available at https://github.com/sony/mf-rae.

1. Introduction

Diffusion models (or flow matching) [15, 34, 36, 37] have been shown to achieve high-fidelity sample generation. Its sampling can be interpreted as solving the associated probability flow ordinary differential equation (PF-ODE) [37]. However, this procedure requires many neural network evaluations to approximate the numerical integration, which makes diffusion model generation notoriously slow.

Recent research [1, 2, 11, 19, 22, 39] has shifted toward *flow map models*, with a promising representative called *MeanFlow* (MF) [11]. These models directly learn the PF-ODE solution map, transporting any initial state at one time to the corresponding state on the same trajectory at another time. As a result, they enable generation in a few steps, mapping pure noise to clean data with only a small number of network evaluations. In practice, MF is often used in a latent space by leveraging a pre-trained Stable Diffusion variational autoencoder (SD-VAE) [30] for high-dimensional image generation. Despite these advances in diffusion-based few-step generative models, MF training and inference remain inefficient for such high-dimensional latent representations.

Training MF, even in a latent space, still requires hundreds of H100 GPU-days [51] on high-dimensional datasets such as ImageNet 256 [7]. It is further complicated by intricate classifier-free guidance (CFG) [14] configurations for class-conditional generation, which involve two CFG scale hyperparameters and two additional hyperparameters controlling the CFG triggering interval. These hyperparameters must be carefully tuned through extensive grid search to maximize MF performance, thereby increasing the overall complexity of training. Moreover, the Jacobian vector product (JVP) required by the MF loss introduces an additional source of computational cost and instability. Even when it is computed using the most efficient forward mode automatic differentiation [33], the JVP remains a significant bottleneck during training. Supporting JVP in modern components such as Flash Attention [6] also requires extra implementation effort [49], making MF cumbersome and time-consuming to adapt to new model architectures.

Regarding the inference, although MF enables few-step generation, the computational cost of the SD-VAE decoder that maps generated latent vectors back to pixel space dominates and substantially slows down the overall generation speed [20]. Specifically, as shown in Figure 1 (a), for

*Work done during an internship at Sony AI.

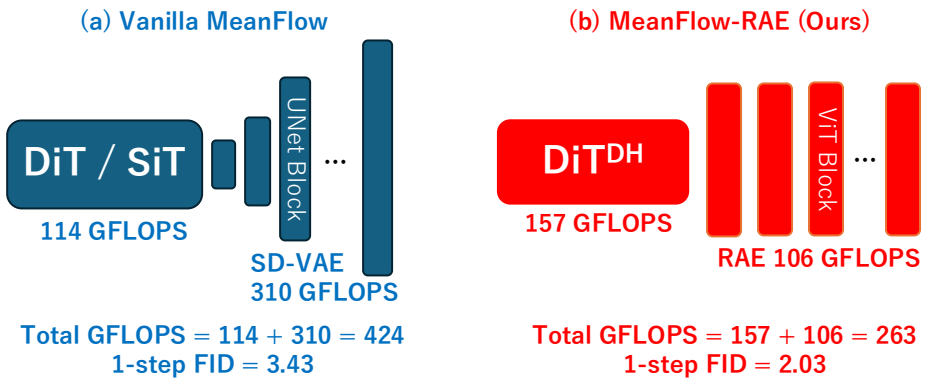

Figure 1. **Overview of our method's advantages.** On ImageNet 256, vanilla MF (a) employs the slow SD-VAE decoder, which accounts for 73% of the total generation cost and thus bottlenecks the few-step generation speed. In contrast, our MF-RAE (b) leverages a higher-dimensional RAE latent space with semantically rich features and an efficient decoder. The DiT$^{\text{DH}}$ architecture is adopted to effectively process the high-dimensional latent space. As a result, while the converged 1-step FID of vanilla MF is 3.43 after more than 600 H100 GPU-days, our MF-RAE achieves a superior FID of 2.03 in only 100 H100 GPU-days. Additionally, our total generation cost is reduced by 38% compared to vanilla MF in terms of GFLOPS, even with the same 1-step generation setting.

vanilla MF with a conventional DiT/SiT architecture combined with SD-VAE on ImageNet 256, the DiT/SiT requires 114 GFLOPS, whereas the SD-VAE decoder consumes 310 GFLOPS, which indicates that approximately 73% of the total computation is spent on the decoder.

Recent research on the Representation Autoencoder (RAE) [48] replaces the conventional SD-VAE in latent diffusion models with a frozen pre-trained representation encoder (e.g., DINO [4]) and trains only a ViT-based [9] decoder on top of its latent tokens. Unlike the classic SD-VAE that uses a U-Net backbone to compress images into a low-dimensional latent space, RAE modernizes this design by adopting a transformer architecture and using semantically rich, high-dimensional representations as the generative space. To model these higher-dimensional latents, RAE augments a DiT backbone with a wide yet lightweight DDT head [42] tailored for latent diffusion modeling, yielding an expressive and efficient DiT$^{\text{DH}}$ architecture.

For latent diffusion models, RAE brings limited improvement in sampling speed, since the main bottleneck is the large number of network evaluations required to solve the latent PF-ODE. However, we emphasize that RAE's decoder efficiency is particularly crucial for few-step models such as MF: its decoder requires about 106 GFLOPS, nearly a $3\times$ reduction compared to the $\sim$310 GFLOPS SD-VAE decoder used in vanilla MF. This directly alleviates the decoder side bottleneck in current few-step generation and is a key motivation for our approach.

To this end, we improve the efficiency of MF training and sampling by learning MF in the RAE latent space, and show that our approach is stable, fast, and high-quality in practice. We systematically analyze and decompose MF training into the following components.

First, we observe that naively training MF with the DiT$^{\text{DH}}$ architecture in the RAE latent space is unstable: gradients explode early in training, regardless of whether MF is initialized randomly or from a pre-trained flow matching teacher. We attribute this to a mismatch between the training signal of flow matching and the objective of MF: flow matching learns infinitesimal transitions along the PF-ODE trajectory, whereas MF must learn long jumps between distant time steps, and random initialization further aggravates this issue. To address this, we initialize MF with weights obtained from Consistency Mid-Training (CMT) [16], which learns a trajectory-aware initialization by following the numerical PF-ODE trajectory of a pre-trained flow matching model.

Second, after stabilizing MF training with CMT, we adopt the MeanFlow Distillation (MFD) algorithm, which efficiently converts a pre-trained flow matching model (teacher) into a few-step MF model. We then introduce a novel optional bootstrapping stage that replaces the teacher with a one-point velocity estimator and performs a brief, low-cost fine-tuning phase. This bootstrapping stage becomes crucial for decoupling MF performance from the teacher's quality, particularly when the teacher is suboptimal: we show theoretically and verify empirically that this two-stage procedure breaks the performance ceiling that the original teacher model would otherwise impose on MF.

To avoid ad-hoc tuning of guidance hyperparameters (e.g., the CFG scale and effective intervals, or the Auto-Guidance strength plus an additional guidance model [18]) in class-conditional generation, which otherwise makes MF highly sensitive to configuration, we distill a pre-trained class-conditional flow matching model in the RAE latent space into an MF model in the same space. This yields a

class-conditional MF model that operates entirely without guidance parameters. Removing guidance therefore both simplifies the configuration and reduces the computational cost per iteration, since guided MF requires extra model evaluations, leading to slower convergence in practice.

Third, to further accelerate MF training, we replace the Jacobian–vector product (JVP) term in the MF loss with a finite-difference approximation, which achieves similar empirical performance to exact JVP [43].

We refer to the resulting model, equipped with all these components, as *MeanFlow-RAE* (MF–RAE). These changes stabilize optimization and significantly speed up convergence, as we will demonstrate empirically. Moreover, we empirically observe that MF-RAE can largely reuse the hyperparameters from flow matching pre-training, with only minor modifications. We attribute this robustness to the expressive RAE latent space in which MF-RAE operates. In contrast, vanilla MF with SD-VAE latents requires careful retuning. Finally, sampling with MF–RAE is also accelerated thanks to the lightweight RAE decoder and the few-step generation nature of MF.

We validate our approach on ImageNet. At 256×256 resolution (see Figure 1 for an overview), we achieve an FID of 2.03 with a single sampling step using approximately 100 H100 GPU-days of training in total (including flow matching pre-training, CMT mid-training, and MF post-training), compared to vanilla MF, which attains an FID of 3.43 after more than 600 H100 GPU-days. Under the same single-step generation setting, our method also reduces the total GFLOPS by 38%, thanks to the efficient RAE decoder. This delivers higher image quality and faster generation with substantially lower training cost. We further scale up MF-RAE to ImageNet 512 and achieve a competitive 1-step FID score of 3.23 while maintaining the lowest sampling GFLOPS cost.

Overall, the MF-RAE framework advances few-step flow map models along three axes: it reduces configuration complexity by removing guidance hyperparameters, stabilizes optimization via CMT-based initialization and MFD-based training targets, and accelerates both training and sampling while improving sample quality. Because it is built on a generic RAE latent space with a DiT-based backbone, MF-RAE remains compatible with future improvements in training algorithms and transformer-based architectures for flow map models, thereby providing an extensible and general pipeline for efficient few-step generation.

2. Preliminary

Representation Autoencoder (RAE). RAE [48] replaces the conventional SD-VAE of dimensionality compression with a pre-trained semantic representation encoder E, such as DINOv2 [28] or SigLIP2 [41]. The encoder E is kept frozen, while a ViT-based [9] decoder D is trained to achieve high-fidelity reconstruction by leveraging high-dimensional latent representations. Specifically, given an input image $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$, the frozen encoder E extracts a semantic representation $\mathbf{z}_0 := E(\mathbf{x})$, which is subsequently decoded by D to reconstruct the pixel-level image $\hat{\mathbf{x}} := D(\mathbf{z}_0)$. The decoder is optimized using a composite objective $\mathcal{L}_{\text{rec}}$ that combines L_1, learned perceptual (LPIPS) [47], and adversarial losses (GAN) [12]:

$$\mathcal{L}_{\text{rec}}(\mathbf{x}) = \omega_{\text{L}}\,\text{LPIPS}(\hat{\mathbf{x}},\mathbf{x}) + \text{L}_1(\hat{\mathbf{x}},\mathbf{x}) + \omega_{\text{G}}\eta\,\text{GAN}(\hat{\mathbf{x}},\mathbf{x}),$$

which defines a high-quality reconstruction objective, where ω_{L}, ω_{G}, and η denote the weights of the respective loss terms. In addition, RAE extends DiT to DiT$^{\text{DH}}$ by incorporating a wide yet efficient DDT head [42], enabling effective modeling of the high-dimensional latent space.

Flow Matching (FM) and Diffusion Model. FM (or diffusion model) [24, 27] interpolates between clean data (or latent representations $\mathbf{z}_0 = E(\mathbf{x})$) $\mathbf{z}_0 \sim p_{\text{data}}$ with noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ via $\mathbf{z}_t = \alpha_t\mathbf{z}_0 + \sigma_t\boldsymbol{\epsilon}$, where a typical choice is $\alpha_t = 1-t$, $\sigma_t = t$ for $t \in [0, 1]$. Data generation from noise is achieved by learning a vector field $\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}_t, t)$ to match the conditional velocity $\alpha_t'\mathbf{z}_0 + \sigma_t'\boldsymbol{\epsilon}$ on average:

$$\mathcal{L}_{\text{FM}}(\boldsymbol{\theta}) = \mathbb{E}_t\,\mathbb{E}_{\mathbf{z}_0,\boldsymbol{\epsilon}}\Big[w(t)\big\|\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}_t,t) - \big(\alpha_t'\mathbf{z}_0 + \sigma_t'\boldsymbol{\epsilon}\big)\big\|_2^2\Big],$$

where $w(t)$ is a time weighting function. The optimum is achieved as $\mathbf{v}(\mathbf{z}_t, t) = \mathbb{E}[\alpha_t'\mathbf{z}_0 + \sigma_t'\boldsymbol{\epsilon}|\mathbf{z}_t]$.

Given pre-trained $\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}_t, t) \approx \mathbf{v}(\mathbf{z}_t, t)$, generation is achieved via the integration of PF-ODE [37], $\frac{\mathrm{d}\mathbf{z}_t}{\mathrm{d}t} = \mathbf{v}(\mathbf{z}_t, t)$, starting from $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ down to $t = 0$. However, solving the PF-ODE requires tens or even hundreds of model inferences, since $\mathbf{v}$ captures only infinitesimal PF-ODE transitions. This makes FM's generation computationally expensive due to numerical discretization. MF addresses this limitation by directly learning long ODE jumps, enabling few-step generation.

MeanFlow (MF). MF [11] learns the average velocity integration for few-step generation, stemming from the idea of fitting arbitrary long PF-ODE jumps between t, s [19]:

$$\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, s) \approx \mathbf{h}(\mathbf{z}_t, t, s) = \frac{1}{t-s}\int_s^t \mathbf{v}(\mathbf{z}_u, u)\,\mathrm{d}u.$$

Taking the derivative with t to obtain the MF identity provides a tractable optimization target:

$$\mathcal{L}_{\text{MF}}(\boldsymbol{\theta}) := \mathbb{E}_{t>s}\mathbb{E}_{\mathbf{z}_t}\Big[\|\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_t,t,s) - \mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{z}_t,t,s)\|_2^2\Big], \tag{1}$$

where the stop-grad regression target (with stop-gradient parameters $\boldsymbol{\theta}^-$) is defined as

$$\mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{z}_t,t,s) := \mathbf{v}(\mathbf{z}_t,t) - (t-s)\big(\mathbf{v}(\mathbf{z}_t,t)\partial_{\mathbf{x}}\mathbf{h}_{\boldsymbol{\theta}^-} + \partial_t\mathbf{h}_{\boldsymbol{\theta}^-}\big).$$

The ground truth velocity $\mathbf{v}(\mathbf{z}_t, t)$ is either replaced by (1) one-point estimation $\alpha_t' \mathbf{z}_0 + \sigma_t' \boldsymbol{\epsilon}$ where $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \boldsymbol{\epsilon}$ in MF Training (MFT) or (2) a teacher flow matching model in MF Distillation (MFD).

3. Proposed Method: MF-RAE

We revisit the MF training loss defined in Equation (1) in a more general form, which clarifies the MF training design and suggests a principled recipe for designing flow map models more broadly.

Let $\mathbf{w}$ be a vector, and consider the stop-gradient regression target defined as

$$\mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s; \mathbf{w}) := \mathbf{w} - (t-s)\big((\partial_{\mathbf{z}} \mathbf{h}_{\boldsymbol{\theta}^-})\mathbf{w} + \partial_t \mathbf{h}_{\boldsymbol{\theta}^-}\big),$$

where $\mathbf{w}$ is either chosen as a one-point estimate of the conditional velocity $\hat{\mathbf{v}}(\mathbf{z}_t, t) := \alpha_t' \mathbf{z}_0 + \sigma_t' \boldsymbol{\epsilon}$ in MFT, or as the output of a pre-trained flow matching model $\mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}_t, t)$ in MFD. Among $\mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}$, the transport derivative $(\partial_{\mathbf{z}} \mathbf{h}_{\boldsymbol{\theta}^-})\,\mathbf{w} + \partial_t \mathbf{h}_{\boldsymbol{\theta}^-}$ along $\mathbf{w}$ can be computed as a JVP of $\mathbf{h}_{\boldsymbol{\theta}^-}$ with respect to $(\mathbf{z}, t, s)$ in the direction $[\mathbf{w}, 1, 0]$, i.e.

$$[\partial_{\mathbf{z}} \mathbf{h}_{\boldsymbol{\theta}^-},\, \partial_t \mathbf{h}_{\boldsymbol{\theta}^-},\, \partial_s \mathbf{h}_{\boldsymbol{\theta}^-}]^\top [\mathbf{w}, 1, 0].$$

This general target $\mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s; \mathbf{w})$ induces a generalized MF loss, which we denote by

$$\mathcal{L}_{\text{MF}}(\boldsymbol{\theta}; \mathbf{w}) := \mathbb{E}_{t>s}\, \mathbb{E}_{\mathbf{z}_t}\Big[\big\| \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, s) - \mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s; \mathbf{w})\big\|_2^2\Big]. \tag{2}$$

Building on this general formulation, we develop a systematic view of MF training and propose concrete improvements along four key axes:

- Section 3.1: a better latent space $\mathbf{z}_t$ for MF modeling using RAE latents;
- Section 3.2: trajectory-aware initialization for the MF $\mathbf{h}_{\boldsymbol{\theta}}$ via Consistency Mid-Training (CMT);
- Section 3.3: choice of the proxy velocity $\mathbf{w}$ in $\mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\cdot; \mathbf{w})$ through a trade-off between MFD and MFT;
- Section 3.4: efficient computation of the transport derivative $(\partial_{\mathbf{z}} \mathbf{h}_{\boldsymbol{\theta}^-})\mathbf{w} + \partial_t \mathbf{h}_{\boldsymbol{\theta}^-}$ via finite differences.

We refer to the resulting designs of MF with RAE-based latent modeling and our tailored training scheme as *MeanFlow-RAE* (MF-RAE). In the following sections, we present the components of MF-RAE in detail.

3.1. MF with DiT$^{\text{DH}}$ Architecture and RAE Latents

RAE, originally proposed for latent flow matching (diffusion) models, achieves strong generation quality even *without* guidance by leveraging the expressive latent space of a pre-trained visual encoder. However, in standard latent diffusion models, RAE brings only limited gains in wall-clock sampling speed: although its ViT-based decoder is lighter in terms of GFLOPS, the dominant bottleneck remains the iterative PF-ODE solving of the latent diffusion model itself.

In contrast, RAE is particularly well-suited to MF and, more broadly, to few-step latent flow map models. Since MF evaluates the latent model in only one or two steps, the overall generation cost is no longer dominated by iterative dynamics. Thus, RAE's efficient decoder's acceleration is more pronounced. Moreover, the rich RAE latents can accelerate MF convergence. They also enable high-quality generation *without* any guidance such as CFG or Auto-Guidance, thereby eliminating guidance-related hyperparameters and significantly simplifying MF training.

We propose the MF-RAE transformer architecture by extending the DiT$^{\text{DH}}$ backbone used in RAE with an additional time-embedding module for the time difference $t-s$. Specifically, we sum the embeddings of the class label, the current time t, and the time difference $t-s$, whereas the original DiT$^{\text{DH}}$ only sums the label and time-t embeddings. This simple change allows the model to explicitly encode both the absolute time and the time difference, which is important for learning accurate flow maps in MF.

This modification is analogous to the change used in vanilla MF when adapting DiT/SiT backbones of the diffusion model for flow map learning, ensuring a fair comparison with the vanilla MF baseline. More broadly, any architectural changes developed to adapt DiT/SiT to MF can be incorporated in the same way into the DiT$^{\text{DH}}$ backbone. Thus, future advances in architectures for MF can be plugged into our MF-RAE without altering the overall design, highlighting our generality and extensibility.

3.2. Stabilizing MF via Consistency Mid-Training (CMT) Initialization

Although DiT$^{\text{DH}}$ with RAE provides a stronger and more efficient latent backbone than DiT/SiT with SD-VAE, we find that naively training MF $\mathbf{h}_{\boldsymbol{\theta}}$ on RAE latents is highly unstable. In particular, gradients explode whether $\mathbf{h}_{\boldsymbol{\theta}}$ is initialized randomly or from a pre-trained FM teacher. Empirically, with the XL model, training diverges almost immediately; for example, under FM-teacher initialization, both the loss and gradient norm spike to around 10^5 by epoch 2. At smaller scales (S and B), both random and diffusion-based initializations remain stable only in the very early phase; the loss then gradually increases and eventually blows up. The best 1-step FID observed before divergence is still above 20, which is far from convergence.

To mitigate this instability, we initialize $\mathbf{h}_{\boldsymbol{\theta}}$ using weights obtained from Consistency Mid-Training (CMT) [16]. Instead of starting the MF model $\mathbf{h}_{\boldsymbol{\theta}}$ from a pre-trained infinitesimal flow matching model (which learns only local jumps along the PF-ODE trajectory) or from an unstructured random initialization, we first run CMT to learn a trajectory-aware initialization from the numerical

ODE trajectory of the teacher flow matching model:

$$\mathcal{L}_{\text{CMT-MF}}(\boldsymbol{\theta}) = \mathbb{E}_{i>j}\mathbb{E}_{\mathbf{z}_T\sim p_{\text{prior}}}\Big[\big\|\mathbf{h}_{\boldsymbol{\theta}}(\hat{\mathbf{z}}_{t_i}, t_i, t_j) - \tfrac{\hat{\mathbf{z}}_{t_i}-\hat{\mathbf{z}}_{t_j}}{t_i-t_j}\big\|_2^2\Big], \tag{3}$$

where $\{\hat{\mathbf{z}}_{t_i}\}$ is the teacher FM model's trajectory, obtained by integrating from a prior sample $\mathbf{z}_T \sim p_{\text{prior}}$ and evaluating it at the discrete time grid $\{t_i\}$. In other words, CMT warm-up trains $\mathbf{h}_{\boldsymbol{\theta}}$ to reproduce a proxy of the long jumps required by MF by matching the corresponding long transitions along the teacher trajectory.

In the original CMT setup, a high-order multistep ODE solver (e.g., second-order Heun) is used to obtain accurate teacher trajectories within 16 NFEs. In our RAE setting, a simple first-order Euler solver with 16 NFEs already suffices: on ImageNet 256, the RAE diffusion achieves FID 1.51 with 50 steps and 2.32 with 16 steps, which is more than adequate for CMT's teacher.

3.3. Trade-Offs Between MFD and MFT

With the formulation in Equation (2), $\mathbf{w}$ is usually chosen either as a point estimate of the conditional velocity $\hat{\mathbf{v}}$ (MFT), or as the output of a pre-trained diffusion teacher $\mathbf{v}_{\boldsymbol{\phi}}$ (MFD). However, it remains unclear, in a principled sense, which choice of $\mathbf{w}$ is more beneficial for the training.

The following proposition clarifies this by characterizing how replacing the oracle MF $\mathbf{h}(\mathbf{z}_t, t, s) = \frac{1}{t-s}\int_s^t \mathbf{v}(\mathbf{z}_u, u), \mathrm{d}u$ with the proxy $\mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s; \mathbf{w})$, together with the choices $\mathbf{w} = \hat{\mathbf{v}}$ or $\mathbf{w} = \mathbf{v}_{\boldsymbol{\phi}}$, makes the practical objective $\mathcal{L}_{\text{MF}}(\boldsymbol{\theta}; \mathbf{w})$ deviate from the oracle objective loss function.

Proposition 3.1. *For any $\lambda \in [0, 1]$, consider the combination of the one-point estimator and the pre-trained velocity*

$$\mathbf{w}_\lambda := (1-\lambda)\hat{\mathbf{v}} + \lambda \mathbf{v}_{\boldsymbol{\phi}}.$$

Plugging $\mathbf{w}_\lambda$ into the target $\mathbf{h}^{\text{tgt}}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s; \mathbf{w})$ (with $\mathbf{w} = \mathbf{w}_\lambda$) yields the corresponding loss $\mathcal{L}_{\text{MF}}(\boldsymbol{\theta}; \mathbf{w}_\lambda)$. Consider the following three residuals: the one-point velocity residual, the teacher–oracle velocity residual, and the oracle bias.

$$\begin{aligned}
\delta\hat{\mathbf{v}}_t &:= \hat{\mathbf{v}}(\mathbf{z}_t, t) - \mathbf{v}(\mathbf{z}_t, t),\\
\delta\mathbf{v}^{\boldsymbol{\phi}}_t &:= \mathbf{v}_{\boldsymbol{\phi}}(\mathbf{z}_t, t) - \mathbf{v}(\mathbf{z}_t, t),\\
\delta\mathbf{h}(\mathbf{z}_t, t, s) &:= \mathbf{h}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s) - \mathbf{h}(\mathbf{z}_t, t, s).
\end{aligned}$$

Then the MF loss admits the following decomposition:

$$\begin{aligned}
\mathcal{L}_{\text{MF}}(\boldsymbol{\theta}; \mathbf{w}_\lambda) = \mathbb{E}_{t,\mathbf{z}_t}\Big\|\mathbf{h}_{\boldsymbol{\theta}} - \big(\mathbf{h} + \mathbf{B} + \lambda\,\mathbf{A}_{\boldsymbol{\theta}^-}\,\delta\mathbf{v}^{\boldsymbol{\phi}}_t\big)\Big\|^2\\
+ (1-\lambda)^2\,\mathbb{E}\big\|\mathbf{A}_{\boldsymbol{\theta}^-}\,\delta\hat{\mathbf{v}}_t\big\|^2,
\end{aligned} \tag{4}$$

where $\mathbf{B}(\mathbf{z}_t, t, s) := (t-s)\Big(\partial_t\delta\mathbf{h} + \big(\nabla_{\mathbf{x}}\delta\mathbf{h}\big)\mathbf{v}\Big)$ and $\mathbf{A}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s) := \mathbf{I} - (t-s)\,\nabla_{\mathbf{x}}\mathbf{h}_{\boldsymbol{\theta}^-}$.

The proof is provided in Section A. When $\lambda = 0$, we have $\mathbf{w}_0 = \hat{\mathbf{v}}$, and Equation (4) reduces to

$$\mathcal{L}_{\text{MF}}(\boldsymbol{\theta}; \mathbf{w}_0) = \mathbb{E}_{t,\mathbf{z}_t}\big\|\mathbf{h}_{\boldsymbol{\theta}} - (\mathbf{h} + \mathbf{B})\big\|^2 + \mathbb{E}\big\|\mathbf{A}_{\boldsymbol{\theta}^-}\,\delta\hat{\mathbf{v}}_t\big\|^2,$$

which corresponds to MF trained purely from the one-point velocity estimator (MFT). In this case, the loss includes the full variance induced by the noisy one point estimate $\hat{\mathbf{v}}$ through the second term. When $\lambda = 1$, we have $\mathbf{w}_1 = \mathbf{v}_{\boldsymbol{\phi}}$, and the variance term disappears:

$$\mathcal{L}_{\text{MF}}(\boldsymbol{\theta}; \mathbf{w}_1) = \mathbb{E}_{t>s,\mathbf{z}_t}\big\|\mathbf{h}_{\boldsymbol{\theta}} - (\mathbf{h} + \mathbf{B} + \mathbf{A}_{\boldsymbol{\theta}^-}\,\delta\mathbf{v}^{\boldsymbol{\phi}}_t)\big\|^2.$$

This corresponds to the pure distillation regime (MFD), where the objective depends only on the teacher velocity residual $\delta\mathbf{v}^{\boldsymbol{\phi}}_t$ and the oracle bias $\mathbf{B}$.

Therefore, when the teacher model is of sufficiently high quality, such as a strong RAE flow matching model (i.e., $\delta\mathbf{v}^{\boldsymbol{\phi}}_t \approx \mathbf{0}$), MFD yields both smaller bias and lower variance, leading to a faster convergence rate. In practice, however, MFD is still limited by the quality of the teacher, since it inevitably inherits some bias whenever $\delta\mathbf{v}^{\boldsymbol{\phi}}_t \neq \mathbf{0}$. To further reduce this residual bias, we may optionally apply MFT after MFD has converged. Although MFT typically has higher variance, this variance is effectively reduced when starting from a well converged MFD model, allowing us to benefit from the smaller bias characteristic of MFT. When the teacher model is sufficiently strong, the MF model obtained by MFD alone already achieves good performance, and the additional MFT stage becomes unnecessary.

To the best of our knowledge, we are the first to provide a clear theoretical analysis of the trade-off between MFD and MFT. In summary, our theoretical results suggest a practical bias–variance control procedure that first applies MFD with a pre-trained flow matching model, followed by an optional MFT stage using a one-point estimate.

3.4. Finite Difference for Generality

The main computational and stability bottleneck of MF is the JVP required in the transport derivative $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{h}_{\boldsymbol{\theta}^-}(\mathbf{z}_t, t, s) = (\partial_{\mathbf{z}}\mathbf{h}_{\boldsymbol{\theta}^-})\mathbf{w} + \partial_t\mathbf{h}_{\boldsymbol{\theta}^-}$ in its regression target. A natural way to avoid explicit JVPs is to approximate the time derivative with a finite difference [43]. Given a step size Δt, we write

$$\tfrac{\mathrm{d}}{\mathrm{d}t}\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_t, t, s) \approx \tfrac{\mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_{t+\Delta t}, t+\Delta t, s) - \mathbf{h}_{\boldsymbol{\theta}}(\mathbf{z}_{t-\Delta t}, t-\Delta t, s)}{2\Delta t},$$

where $\mathbf{z}_{t\pm\Delta t} \approx \mathbf{z}_t \pm \Delta t\mathbf{w}(\mathbf{z}_t, t)$ is obtained by a first-order Euler step along the teacher velocity field $\mathbf{w} = \mathbf{v}_{\boldsymbol{\phi}}$.

Empirically, choosing $\Delta t \in [0.001, 0.01]$ yields stable training, and values in this range lead to similar convergence behavior and performance comparable to using exact JVPs (i.e., the limit $\Delta t \to 0$), indicating that the discretization error is negligible in practice [43]. Hence, we simply fix the middle value $\Delta t = 0.005$ throughout our experiments.

3.5. Summary of MF-RAE Pipeline

In summary, we decompose the challenging MF-RAE training into three more manageable stages, adopting a divide-and-conquer approach. Each stage plays a crucial role in facilitating and stabilizing the subsequent stage.

(1) **Pre-training:** Train a high-quality flow matching teacher in the RAE latent space.
(2) **Mid-training:** Apply CMT to learn a trajectory-aware MF initialization (using the proposed DiT^{DH} architecture), where the pre-trained teacher generates the reference trajectories and serves as CMT's initialization.
(3) **Post-training:** Starting from the CMT weights, train the MF model in the RAE latents using the MFD with finite differences; optionally, apply MFT afterward to further reduce loss bias and improve model quality.

4. Related Work

4.1. Diffusion Models and Flow Matching

Diffusion models/flow matching aims to learn a time-dependent velocity field that gradually transforms a simple Gaussian prior distribution into the target data distribution. In diffusion models, this transformation can be described by the PF-ODE, allowing for flexible generative sampling via the numerical simulation of ODEs [24, 25, 37].

In high-dimensional image synthesis, Diffusion Transformers (DiT) in the SD-VAE latent space pioneered scaling diffusion models to large, high-fidelity tasks [29, 30], enabling efficient training and inference while preserving semantic and visual quality. SiT [27] extends DiT with flow matching interpolants for more flexible distribution transport. Subsequent work leverages semantic representations to improve reconstruction and generation [5, 17, 21], including REPA [45], which aligns SiT features with pre-trained encoders to speed up training, and REG [44], which further injects a pre-trained encoder's class token in denoising to capture image-label pair information better. RAE [48] further treats a discriminative encoder as a tokenizer to enable semantic-space reconstruction and generation while alleviating issues of high-dimensional latent spaces. However, these models still struggle to achieve high fidelity under few-step sampling.

4.2. Few-Step Flow Map Models

Diffusion models and flow matching suffer from slow sampling due to the recursive model inferences required during ODE solving after discretizing numerous time steps. Flow map models, such as the consistency model (CM) [38], consistency trajectory model (CTM) [19], and MF [11], directly learn the solution map of a deterministic PF-ODE, thereby enabling fast few-step sampling. Specifically, CM learns to map any noisy point along the trajectory directly to its corresponding clean point on the same ODE trajectory. CTM extends CM by learning mappings between arbitrary points along the ODE trajectory. MF shares an mathematically equivalent parameterization with CTM but instead learns the average ODE integration between two points.

5. Experiments

Dataset and Setup. We evaluate sample quality by FID [13] on class-conditional ImageNet 256 and 512, training cost by total H100 GPU time, and generation efficiency by GFLOPs per sample. Given a pre-trained RAE encoder–decoder, the MF-RAE pipeline comprises three stages (Section 3.5). In the CMT mid-training stage, CMT generates two trajectories (no CFG) per iteration from the teacher DiT^{DH} using a 16-step Euler ODE solver, achieving FID 2.3/1.66 on ImageNet 256/512. Details of the flow-matching pre-training and CMT mid-training stages are deferred to Appendix B. Below, we describe the simplified hyperparameter setting used to train the MF model in MF-RAE.

Hyperparameter Simplicity of MF-RAE. We train MF-RAE with (almost) the same hyperparameters as the DiT^{DH} flow matching stage, changing only a few scalars: we reduce the batch size from 1024 to 256/128 for ImageNet 256/512, lower the learning rate from 2×10^{-4} to 1×10^{-4} (and keep it fixed for both resolutions), and adjust the EMA rate from 0.9995 to $0.9999/0.9995$ for ImageNet 256/512. The smaller batch sizes are purely for efficiency, enabled by the stable CMT initialization, while the learning rate and EMA are aligned with the original MF settings [11] to isolate architectural and algorithmic effects. In practice, this means one can almost directly reuse the flow matching configuration and obtain a few-step MF-RAE generator with only minimal tweaks.

By contrast, vanilla MF requires substantial hyperparameter redesign relative to its flow matching teacher. The corresponding SiT + SD-VAE model [27] is trained with uniform time sampling, whereas vanilla MF switches to a carefully tuned log-normal time distribution and further depends on delicate choices of CFG weights, CFG time intervals, and an additional CFG mixing scale κ (in their notation) to make the method work well. MF-RAE needs none of these bespoke techniques: we keep the teacher's uniform time sampling and use no guidance for class-conditional generation, yet still obtain fast, stable convergence. This highlights that MF-RAE is significantly more hyperparameter-robust and easier to deploy than vanilla MF.

5.1. ImageNet 256 Main Results (Table 1)

The sample quality results of MF-RAE compared with various baseline models are presented in Table 1. Our MF-RAE achieves state-of-the-art (SOTA) 1-step and 2-step genera-

tion quality among all few-step flow map models. Furthermore, these strong results are obtained with both lower generation and training costs, as explained below.

Table 1. Sample quality on class-conditional ImageNet 256.

METHOD	NFE (↓)	FID (↓)	#Params
Diffusion Models & Flow Matching (*no guidance)			
ADM-G [8]	250×2	3.94	554M
DiT-XL/2 [29]	250×2	2.27	675M
SiT-XL/2 [27]	250×2	2.06	675M
REPA [45]	250×2	1.29	675M
REG [44]	250×2	1.36	675M
RAE* [48]	50	1.51	839M
RAE [48]	50×2	**1.13**	839M
GANs & Masked Models			
BigGAN [3]	1	6.95	112M
StyleGAN [32]	1	2.30	166M
MAR [23]	256×2	**1.55**	943M
VAR-*d*30 [40]	10×2	1.92	2B
Flow Map Models			
iCT [35]	1 / 2	34.24 / 20.30	675M
IMM [50]	2	7.77	675M
Shortcut [10]	1	10.60	675M
MeanFlow [11]	1 / 2	3.43 / 2.20	676M
CMT w/ MF [16]	1	3.34	676M
AlphaFlow [46]	1 / 2	2.58 / 1.95	675M
MF-RAE (Ours)	1 / 2	**2.03 / 1.89**	841M

Faster Generation of MF-RAE over Baselines. The total generation cost, measured in GFLOPS, is the decoder cost plus (NFE) times the diffusion transformer cost. In the ImageNet 256 experiments, our method achieves lower GFLOPS and thus faster generation speed for both 1-step and 2-step generation. For the 1-step case, vanilla DiT-based MF requires $310 + 114 = 424$ GFLOPS, whereas our approach uses only $106 + 157 = 263$ GFLOPS. For the 2-step case, vanilla DiT-based MF costs $310 + 114 \times 2 = 538$ GFLOPS, while ours requires just $106 + 157 \times 2 = 420$ GFLOPS. Despite having the same NFE, MF-RAE enables faster generation, as illustrated in Figure 1.

Faster Convergence of MF-RAE over Baselines. We compare the total convergence time of vanilla MF and our MF-RAE on ImageNet 256. Vanilla MF is trained from scratch using the MFT (i.e., without a pre-trained teacher), requiring 1400 epochs, corresponding to about 7M iterations, with a total training cost of over 600 H100 GPU-days.

By contrast, MF-RAE proceeds in three stages: flow matching pre-training for 800 epochs / 1M iterations (78 H100 GPU-days), CMT mid-training for 27K iterations (2.1 H100 GPU-days), and MFD post-training for 36 epochs / 180K iterations (21 H100 GPU-days). In total, MF-RAE requires only about 100 H100 GPU-days, representing more than a $6\times$ reduction in training cost compared to vanilla MF, while achieving faster convergence.

Moreover, once a flow matching teacher has been pre-trained or is available off the shelf, the additional cost of converting it into a few-step MF model via CMT mid-training and MF distillation is only 23 H100 GPU-days. This shows that MF-RAE provides an efficient and practical route to distill a strong flow matching model into a fast few-step generator.

Table 2. Ablation on latent representation and training scheme for MF on ImageNet 256. Compared to MF trained on SD-VAE latents with DiT/SiT, our MF-RAE configuration (RAE with DiT$^{\text{DH}}$ via MFD; last row) converges faster and achieves the best 1- and 2-step FIDs while requiring no guidance. In contrast, MF trained on SD-VAE latents exhibits severe performance degradation when guidance is removed. On a fixed latent space, MFD also outperforms MFT, showing that both the RAE representation and the MFD objective are key to fast and stable MF convergence.

Algorithm	Guided?	Architecture	NFE (↓)	FID (↓)
MFT	✓	SD-VAE with DiT/SiT	1 / 2	3.38 / 2.20
MFD	✓	SD-VAE with DiT/SiT	1 / 2	3.15 / 1.95
MFD	×	SD-VAE with DiT/SiT	1 / 2	5.94 / 4.01
MFT	×	RAE with DiT$^{\text{DH}}$	1 / 2	2.81 / 2.56
MFD	×	RAE with DiT$^{\text{DH}}$	1 / 2	**2.03 / 1.89**

Ablation with Latent Representations and MFT versus MFD. We empirically analyze how the choice of latent representation (SD-VAE paired with DiT/SiT vs. RAE paired with DiT$^{\text{DH}}$) and the training scheme (MFT vs. MFD) affect performance on ImageNet 256, with results summarized in Table 2. The first row in Table 2 indicates the vanilla MF, while the last row is our MF-RAE.

For the SD-VAE setting, we use the vanilla MF-XL/2 configuration and a recent REG-based SiT teacher [44]. This teacher attains FID 1.36 with CFG and 1.80 unguided, comparable to the RAE-space teacher (FID 1.51 unguided). To ensure a fair comparison, we keep batch size, learning rate, EMA, CMT mid-training iterations, and optimizer identical across settings.

Comparing MF trained on SD-VAE latents (via MFD or MFT) with our MF-RAE configuration (DiT$^{\text{DH}}$ + RAE via MFD), MF-RAE converges faster and achieves the best FID among methods with similar teacher quality. Moreover, we observe that MF on SD-VAE latents performs poorly without guidance, even with distillation, whereas MF-RAE attains high-quality unguided class-conditional generation. This shows that the RAE latents is crucial for simplifying MF training and eliminating any guidance hyperparameters.

Interestingly, vanilla MF (trained on SD-VAE latents) can be trained from scratch with random initialization, but requires about 1400 epochs (600+ H100 GPU-days) to converge. In contrast, MF on the semantic RAE latent space

cannot be trained from scratch, either with random initialization or with a pre-trained diffusion teacher as initialization, unless we use CMT initialization. This indicates that the effectiveness of MF is strongly dependent on the choice of representation and architecture, and that our MF-RAE with CMT initialization provides a general stability mechanism that enables future model extensions.

Finally, we directly compare MFD and MFT on the same RAE latent representation (the last two rows in Table 2). In this setting, MFD with a pre-trained teacher achieves 1- and 2-step FIDs of 2.03 and 1.89, while MFT with a one-point velocity reaches only 2.81 and 2.56, showing that MFD is substantially more effective. Given a well-trained flow matching teacher, distillation supplies low-variance training and high-quality velocity targets (see Proposition 3.1), so MF-RAE converges faster and to better performance. Since the teacher-student FID gap is already small (the teacher with 50 NFEs has FID 1.51), an additional bootstrapping stage is unnecessary in this regime.

Ablation of JVP vs. Finite Difference. Table 3 compares MF-RAE FIDs using JVP vs. a finite-difference approximation across Δt (default 5×10^{-3}). Finite difference matches JVP for $\Delta t \in [10^{-3}, 10^{-2}]$ with marginal FID changes, while too small Δt (e.g., 10^{-4}) becomes numerically unstable and degrades FID. Overall, finite-difference proxy is robust over a broad Δt range and yields $\sim 1.5\times$ faster wall-clock training than JVP.

Table 3. JVP vs. Δt.

		Δt			
NFE	**JVP**	10^{-4}	10^{-3}	5×10^{-3}	10^{-2}
1	1.96	5.63	2.06	2.03	2.17
2	1.87	4.22	1.87	1.89	1.94

5.2. Scale up to ImageNet 512 (Table 4)

We scale up our MF-RAE to ImageNet 512; to the best of our knowledge, this is among the first extensions of MF to this resolution. The sample quality (FID) and generation cost (GFLOPS) are summarized in Table 4. Our method attains a competitive 1-step FID while achieving the lowest generation cost, owing to the efficient decoder. These near-SOTA results are obtained without any guidance: CMT and AYF distills from Auto-Guidance teachers, sCD distills from a CFG teacher, while sCT (which relies solely on one-point velocity estimation without a guided teacher) performs worse than ours. This highlights the simplicity and effectiveness of our approach and suggests that MF-RAE could be further improved by incorporating guidance techniques. Moreover, our results are obtained with substantially shorter training time than SOTA methods such as sCD and CMT. More specifically, we perform MFD for 20K iterations, followed by MFT for an additional 10K iterations using the MFD checkpoint as initialization. The CMT stage requires approximately 8 H100 GPU days, while the combined MFD+MFT stage takes about 9 H100 GPU days. The total cost of 17 H100 GPU days is comparable to that of CMT with ECD (17 days) and significantly lower than sCD's 233 days and sCT's 98 days.

Table 4. Sample quality on class-conditional ImageNet 512. We report the sampling GFLOPS containing decoder and diffusion transformer costs. Methods with * require additional complicated guidance hyperparameters obtained from extensive grid searches.

METHOD	**NFE**	**FID (↓)**	**GFLOPS (↓)**	**#Params**
Flow Map Models				
CMT w/ ECD* [16]	1	3.38	2344	1.5B
sCD* [26]	1	2.28	2344	1.5B
sCT [26]	1	4.29	2344	1.5B
AYF* [31]	1	3.32	1342	280M
MF-RAE (Ours)	1	3.23	1051	841M

Ablation with Initializations Scheme and Bootstrapping Strategy. We empirically validate the proposed bootstrapping strategy, namely the combined MFD+MFT training scheme that replaces the pre-trained teacher with a one-point velocity objective for further fine-tuning. We consider three initialization methods (random, flow matching, CMT) and three training algorithms (MFT only, MFD only, and the bootstrapped MFD+MFT combination), and train each configuration for 30K optimization iterations.

For initialization, both random and flow matching initializations lead to gradient explosions at the beginning of training, whereas only CMT yields stable optimization. This confirms the effectiveness of CMT as a trajectory-aware initialization for MF.

Among training schemes, MFT alone performs worst (one-step FID 5.82), consistent with its high gradient variance. MFD alone converges faster due to lower variance, reaching near-convergence by 20K iterations (one-step FID 3.95). Starting from that point, switching to the one-point MFT objective for an additional 10K iterations further reduces bias and attains one-step FID 3.23. Thus, MFD is well-suited for the early stage to accelerate convergence, while a brief MFT phase refines the model, aligning with Proposition 3.1.

6. Conclusion

Training MF in the RAE latent space is challenging. By combining CMT for stable initialization, MFD for accelerated convergence, and an optional lightweight bootstrapping stage for further refinement, our approach substantially reduces training cost, simplifies configuration by removing guidance, and enables faster few-step generation while preserving state-of-the-art performance. Our pipeline provides a general recipe for training flow-map models in the RAE latent space and can readily incorporate future advances, demonstrating both flexibility and extensibility.

References

[1] Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. *arXiv preprint arXiv:2505.18825*, 2025. 1

[2] Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025. 1

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 7

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2

[5] Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. 6

[6] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 1

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[10] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025. 7

[11] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. 1, 3, 6, 7

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

[13] 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. 6

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

[16] Zheyuan Hu, Chieh-Hsin Lai, Yuki Mitsufuji, and Stefano Ermon. CMT: Mid-Training for Efficient Learning of Consistency, Mean Flow, and Flow Map Models. *arXiv preprint arXiv:2509.24526*, 2025. 2, 4, 7, 8

[17] Ziyuan Huang, DanDan Zheng, Cheng Zou, Rui Liu, Xiaolong Wang, Kaixiang Ji, Weilong Chai, Jianxin Sun, Libin Wang, Yongjie Lv, et al. Ming-univision: Joint image understanding and generation with a unified continuous tokenizer. *arXiv preprint arXiv:2510.06590*, 2025. 6

[18] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024. 2

[19] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024. 1, 3, 6

[20] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Pagoda: Progressive growing of a one-step generator from a low-resolution diffusion teacher. *arXiv preprint arXiv:2405.14822*, 2024. 1

[21] Theodoros Kouzelis, Efstathios Karypidis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Boosting generative image modeling via joint image-feature synthesis. *arXiv preprint arXiv:2504.16064*, 2025. 6

[22] Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025. 1

[23] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024. 7

[24] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. 3, 6

[25] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 6

[26] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *International Conference on Learning Representations*, 2025. 8

[27] 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. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024. 3, 6, 7

[28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 6, 7

[30] 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. 1, 6

[31] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. *arXiv preprint arXiv:2506.14603*, 2025. 8

[32] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 7

[33] Zekun Shi, Zheyuan Hu, Min Lin, and Kenji Kawaguchi. Stochastic Taylor derivative estimator: Efficient amortization for arbitrary differential operators. *Advances in Neural Information Processing Systems*, 37:122316–122353, 2024. 1

[34] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1

[35] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. 7

[36] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 1

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1, 3, 6

[38] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 6

[39] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023. 1

[40] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in Neural Information Processing Systems*, 37:84839–84865, 2024. 7

[41] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 3

[42] Shuai Wang, Zhi Tian, Weilin Huang, and Limin Wang. DDT: Decoupled Diffusion Transformer, 2025. 2, 3

[43] Zidong Wang, Yiyuan Zhang, Xiaoyu Yue, Xiangyu Yue, Yangguang Li, Wanli Ouyang, and Lei Bai. Transition models: Rethinking the generative learning objective. *arXiv preprint arXiv:2509.04394*, 2025. 3, 5

[44] Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, et al. Representation entanglement for generation: Training diffusion transformers is much easier than you think. *arXiv preprint arXiv:2507.01467*, 2025. 6, 7

[45] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. 6, 7

[46] Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. AlphaFlow: Understanding and Improving MeanFlow Models. *arXiv preprint arXiv:2510.20771*, 2025. 7

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 3

[48] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. 2, 3, 6, 7, 13

[49] Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large Scale Diffusion Distillation via Score-Regularized Continuous-Time Consistency. *arXiv preprint arXiv:2510.08431*, 2025. 1

[50] Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *International Conference on Machine Learning*, 2025. 7

[51] Yu Zhu. MeanFlow: PyTorch Implementation. https://github.com/zhuyu-cs/MeanFlow, 2025. PyTorch implementation of Mean Flows for One-step Generative Modeling. 1