DISCRETE MEANFLOW TRAINING CURRICULUM

Chia-Hong Hsu
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
chsu35@student.ubc.ca

Frank Wood
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
fwood@cs.ubc.ca

ABSTRACT

Flow-based image generative models exhibit stable training and produce high quality samples when using multi-step sampling procedures. One-step generative models can produce high quality image samples but can be difficult to optimize as they often exhibit unstable training dynamics. Meanflow models exhibit excellent few-step sampling performance and tantalizing one-step sampling performance. Notably, MeanFlow models that achieve this have required extremely large training budgets. We significantly decrease the amount of computation and data budget it takes to train Meanflow models by noting and exploiting a particular discretization of the Meanflow objective that yields a consistency property which we formulate into a "Discrete Meanflow" (DMF) Training Curriculum. Initialized with a pretrained Flow Model, DMF curriculum reaches one-step FID 3.36 on CIFAR-10 in only 2000 epochs. We anticipate that faster training curriculums of Meanflow models, specifically those fine-tuned from existing Flow Models, drives efficient training methods of future one-step examples.

1 INTRODUCTION

Diffusion models and flow-based generative frameworks have fundamentally redefined the landscape of generative AI, offering a level of training stability and mode coverage that Generative Adversarial Networks (GAN) Goodfellow et al. (2014); Karras et al. (2018) notably lacked Ho et al. (2020); Song et al. (2021); Lipman et al. (2023). By transforming noise priors into complex data distributions through a probability path, these models have demonstrated remarkable generalization across diverse modalities, including high-resolution images, video, and audio Ho et al. (2022); Zhu et al. (2026); Podell et al. (2023). However, unlike one-step GAN's, the inherent requirement of multi-step iterative sampling over the Probability Flow ODE path remains a significant bottleneck for real-time applications. This limitation has sparked intense research interest in one/few-step variants, primarily through the lens of trajectory-aware distillation, reconciling high quality generation baselines with inference efficiency Salimans & Ho (2022); Luhman & Luhman (2021).

The development of one/few-step generative models has increasingly shifted toward objectives derived from flow-matching trajectories. While initially shown to be an effective distillation technique to accelerate pre-trained teachers, researchers are increasingly interested in the potential of these objectives to function as self-contained, "from scratch" models. As a first, the self-supervised Consistency Training framework enabled Consistency Models to learn the solution of the underlying diffusion ODE Song et al. (2023). This line of work demonstrated that one-step generation can be achieved by exploiting a fundamental consistency property between sample pairs along diffusion and flow trajectories Kim et al. (2024); Zheng et al. (2024); Hu et al. (2025a).

At the forefront of this shift is MeanFlow (MF) Geng et al. (2025a), a framework that reformulated training around the average velocity over time intervals. Thanks to its ability to generate samples in one/few-steps, this approach has attracted follow-up works into more stabilized training dynamics, improvements in architecture, and distillation efficiency Geng et al. (2025b); Lee et al. (2025). However, this theoretical elegance comes at a prohibitive cost: the continuous MF identity is notoriously expensive to train, often requiring heavy Jacobian-vector products (JVPs) that increases per-batch costs. Recent literature has sought to refine this; for instance, α-Flow Zhang et al. (2025) explores the unification of moment matching with MF, while CMT Hu et al. (2025b) and iMT Geng

et al. (2025b) explore the potential of integrating self-distillation targets directly into the MeanFlow objective to improve convergence. Much of the field's recent progress has been driven by increasing model scale and computational resources, while comparatively less attention has been devoted to developing methods that make training more efficient and affordable Geng et al. (2025b).

In this work, we propose Discrete MeanFlow (DMF) training curriculum, a budget-friendly framework designed to bridge the gap between standard flow models and the MeanFlow identity for fast convergence. Our approach replaces expensive continuous identities with a staged curriculum that progressively introduces more challenging learning objectives. We demonstrate the efficacy of DMF on pixel-space CIFAR-10 Krizhevsky et al. (2009), achieving competitive FID Heusel et al. (2018) scores at a fraction of the GPU-hour budget. On latent-space ImageNet 256×256 Russakovsky et al. (2015) we show that DMF scales effectively with increased training budget, exhibiting continuous performance improvements when initialized from a pretrained flow model. We further report findings on the stability ceiling observed in latent-space experiments, providing insights into how discretization granularity affects optimization robustness.

2 Preliminary: The MeanFlow Identity

MeanFlow (MF) proposes a framework for one-step generative modeling by shifting the perspective from instantaneous velocity fields, standard in flow matching, to average velocity formulation over time intervals. Consider a probability flow defined by the ordinary differential equation (ODE), $\mathrm{d}\mathbf{z}_t = \mathbf{v}_t(\mathbf{z}_t)\,\mathrm{d}t$, where $\mathbf{z}_t \in \mathbb{R}^d$ represents the sample state at time $t \in [0, 1]$, $\mathbf{z}_0 \sim p_{\text{data}}$ and $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Under a diffusion/flow-matching setting, the average velocity $\mathbf{u}(\mathbf{z}_t, r, t)$ can be defined as the integrated displacement from time r to t divided by the interval $(t - r)$, with $0 \le r \le t \le 1$, i.e., $\mathbf{u}(\mathbf{z}_t, r, t) := \mathbf{z}_t - \mathbf{z}_r/(t - r)$. The *MeanFlow Identity* yields,

$$\begin{aligned} \mathbf{u}(\mathbf{z}_t, r, t) &= \mathbf{v}_t(\mathbf{z}_t) + (r - t)\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}(\mathbf{z}_t, r, t) \\ &= \mathbf{v}_t(\mathbf{z}_t) + (r - t)\left(\frac{\partial \mathbf{u}(\mathbf{z}_t, r, t)}{\partial \mathbf{z}_t}\mathbf{v}_t(\mathbf{z}_t) + \frac{\partial \mathbf{u}(\mathbf{z}_t, r, t)}{\partial t}\right) \end{aligned} \tag{1}$$

For the complete derivation, we refer the reader to the original work Geng et al. (2025a). In practice, MF models are trained to predict $\mathbf{u}_\theta(\mathbf{z}_t, r, t)$, where the training target is constructed by the conditional velocity field sampled at $\mathbf{z}_t$, and the Jacobian-vector product (JVP, `torch.func.jvp` in PyTorch) of the model with primals $(\mathbf{z}_t, r, t)$ and tangents $(\mathbf{v}_t, 0, 1)$.

Alternatively, if we carry out the partial derivatives above by the definition of limits, we obtain their discrete forms,

$$\begin{aligned} \partial_{\mathbf{z}_t}\mathbf{u}(\mathbf{z}_t, r, t) &= \left(\lim_{\|\boldsymbol{\delta}_i\| \to 0} \frac{\mathbf{u}(\mathbf{z}_t, r, t)_j - \mathbf{u}(\mathbf{z}_t - \boldsymbol{\delta}_i, r, t)_j}{\|\boldsymbol{\delta}_i\|}\right)_{i,j} := \mathcal{J}_{\mathbf{z}_t}, \\ \partial_t \mathbf{u}(\mathbf{z}_t, r, t) &= \lim_{\|\Delta\| \to 0} \frac{\mathbf{u}(\mathbf{z}_t, r, t) - \mathbf{u}(\mathbf{z}_t, r, t - \Delta)}{\|\Delta\|}. \end{aligned} \tag{2}$$

Plugging in the above limit definitions back into equation 1, we can derive the discretization,

$$\lim_{\Delta \to 0} \mathbf{u}(\mathbf{z}_t, r, t) = \lim_{\Delta \to 0}\left\{\frac{\mathbf{v}_t(\mathbf{z}_t) \cdot \Delta + \mathbf{u}(\mathbf{z}_t - \mathbf{v}_t\Delta, r, t - \Delta) \cdot (t - r)}{(\Delta + t - r)}\right\}, \tag{3}$$

which we call *Discrete MeanFlow* (DMF). A detailed derivation is provided in Appendix A.1. DMFs have been studied previously in attempt to unify the framework of Flow Models (FM) to MeanFlows Zhang et al. (2025), as well as approximating the convergence of MFs without computing the JVP Hu et al. (2025b). If r is fixed, the form in equation 3 reveals a consistency property that aligns the average velocity from different samples along the trajectory that is corrected by the instant velocity change. In practice, DMF model predicts $\mathbf{u}_\theta(\mathbf{z}_t, r, t)$, and the target is simply the interpolation between $\mathbf{v}_t$ and $\mathrm{sg}(\mathbf{u}_\theta(\mathbf{z}_t - \mathbf{v}_t\Delta, r, t - \Delta))$, with $\mathrm{sg}(\cdot)$ denoting the stop gradient. In our work, we study the benefits and stability of decreasing the Δ term as a step function. This approach is motivated by the success of training curriculums in Consistency Training Models Song & Dhariwal (2023); Geng et al. (2024); Dao et al. (2025). We provide our detailed methodology in the following section.

3 METHOD: TRAINING CURRICULUM

Notice that $\mathbf{v}_t$ in both DMF and MF attributes to the signal of convergence. Without it, the model would collapse to a simple solution that produces arbitrary constant. Drawing inspiration from training curriculums in Consistency Models, we propose Discrete MeanFlow (DMF) Training Curriculum that aims to improve convergence of the consistency property by adaptively *turning down the knob*, Δ, in equation 3. Following ECT Geng et al. (2024), our training curriculum equally divides the target training budget into different stages, each with different training targets denoted as $\mathbf{u}^i_{\text{target}}, i \in \{0, ..., K-1\}$, where i is the stage of training, and K is the total number of stages.

We start with a large Δ at the beginning of the curriculum. The step size Δ in DMF can be as large as $(t-r)$. In this case, the target for $\mathbf{u}_\theta(\mathbf{z}_t, r, t)$ becomes $\frac{1}{2}(\mathbf{v}_t(\mathbf{z}_t) + \frac{1}{2}\text{sg}(\mathbf{u}_\theta(\mathbf{z}_r, r, r))$, where $\mathbf{z}_r$ is the cleaner sample that lies on the linear trajectory induced by the conditional velocity field $\mathbf{v}_t = \epsilon - \mathbf{z}_0,\ \mathbf{z}_0 \sim p_{\text{data}}$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Observe that the ground truth for $\mathbf{u}_\theta(\mathbf{z}_r, r, r)$ is the instantaneous velocity at r Geng et al. (2025a), which coincidentally, is also trained using the conditional velocity. Therefore, our first stage of the curriculum collapses to the flow matching objective, where the target is simply $\mathbf{u}^0_{\text{target}} := \mathbf{v}_t$.

For the following i-th intermediate stages, $i \in \{1, ..., K-2\}$, we adaptively decrease the step size Δ by defining it as a function of stage, denoted as Δ_i. With a chosen shrinking factor q, a straightforward design choice would suggest $\Delta_i = (t-r)/q^i$. However, we found out that this led to suboptimal FID performance from our CIFAR-10 experiments. We discovered that a training curriculum based on a noise schedule mapped to the *Variance-Exploding* (VE) diffusion framework Song et al. (2021) proved particularly effective. Specifically, let $\Phi(t) = t/(1-t)$ be the transformation that maps time t to the VE scheme, then

$$t' = \Phi^{-1}\left(\Phi(t) - \frac{\Phi(t) - \Phi(r)}{q^i}\right), \quad \Delta_i^\dagger = t - t'. \tag{4}$$

As a result, the target for these intermediate stages follows as substituting the Δ with $\Delta_i^\dagger$ in the DMF equation 3. For the last stage (stage $K-1$), we fallback to train the model with the MF objective, with the assumption this smoothly transitions to the *hardest* objective. In summary, our training curriculum defines a sequence of objectives for different stages as,

$$\mathbf{u}^i_{\text{target}} := \begin{cases} \mathbf{v}_t, & \text{if } i = 0, \\ \dfrac{\mathbf{v}_t \cdot \Delta_i^\dagger + \text{sg}\left(\mathbf{u}_\theta(\mathbf{z}_t - \mathbf{v}_t \cdot \Delta_i^\dagger, r, t - \Delta_i^\dagger)\right) \cdot (t-r)}{\Delta_i^\dagger + t - r}, & \text{if } 1 \le i \le K-2, \\ \mathbf{v}_t + (r-t) \cdot \text{sg}\left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}_\theta(\mathbf{z}_t, r, t)\right), & \text{if } i = K-1. \end{cases} \tag{5}$$

Note that we follow MF models to compute the JVP with PyTorch's forward auto-differentiation operation in the last stage. A detailed full procedure of the DMF curriculum is provided in the Appendix A.2.

4 EXPERIMENTS

DMF training curriculum shares the same target with flow-matching at the first stage of training. This suggests that initializing our model with a pretrained Flow Model (FM) is a better candidate than random initialization. Our baseline for the CIFAR-10 experiments implements the configurations from the official MF paper. We strictly follow Appendix A in Geng et al. (2025a), with the only difference being EMA ratio that is scaled down w.r.t. the limited training budget. To demonstrate the efficacy of our curriculum under a cold-start scenario, we impose a fixed budget of 4000 epochs for models trained from random initialization and report the best FID achieved. For fair comparison, experiments utilizing MF fine-tuning or the DMF curriculum initialized from an FM are restricted to a 2000-epoch budget, as the base FM itself has been pretrained for 2000 epochs. Our pretrained FM follows the standard configurations from the official flow matching repository[1]., which achieves an FID of 3.09 with 50-steps sampling from our reimplementation.

[1] `github.com/facebookresearch/flow_matching`

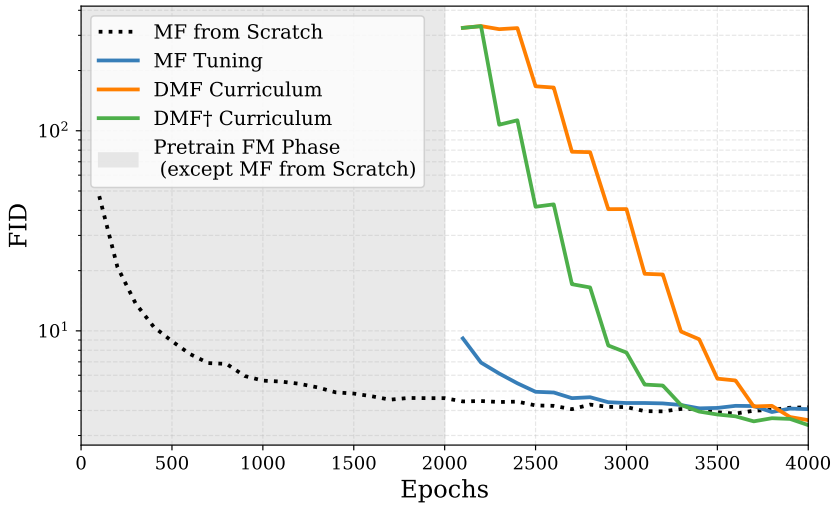

Figure 1: Training convergence on unconditional CIFAR-10. DMF curriculums achieve better 1-step FID compared to the MF baseline with equal training data budget, despite starting from a pretrained flow model at 2000 epochs.

Following the latent diffusion paradigm, ImageNet 256×256 samples are compressed via the SD-VAE Rombach et al. (2022) into a 4×32×32 latent space prior to training. We initialize the model from the weights of a pretrained SiT-XL/2 Ma et al. (2024). Due to computational constraints, we evaluate the DMF curriculum on image generation without Classifier-Free Guidance (CFG) Ho & Salimans (2022). We omit CFG for two primary reasons: first, it significantly increases training overhead by requiring two additional model passes to compute the interpolated target (Tab. 2); second, determining the optimal guidance scale w is computationally expensive, requiring extensive hyperparameter sweeps. Consequently, we compare our results against the SiT-XL/2 baseline sampled without CFG to ensure a fair, though unorthodox, comparison within a fixed training budget.

4.1 Unconditional CIFAR-10

Training Configuration. The baselines include MeanFlow (MF) trained from scratch and MF fine-tuned from a pretrained Flow Model (FM). We set the number of stages for the DMF curriculum to $K = 10$, with a decay factor $q = 2$. We distinguish DMF and DMF† by their curriculum schedules $\{\Delta_i\}$ and $\{\Delta_i^\dagger\}$, respectively, as defined in Section 3. Besides, both DMF's are trained with approximately 100% MF objective. Only regions where timestep difference is infinitesimally small, $t - r < \epsilon_t$, $\epsilon_t = 10^{-6}$, the target is set to the velocity field $\mathbf{v}_t$ for numerical stability. All MF and DMF models are trained using the Adam optimizer with a learning rate of 6×10^{-4}, batch size 1024, adaptive loss Geng et al. (2024), and are evaluated using the same EMA=0.999. A detailed configuration is included in Appendix A.3.

CIFAR-10 Convergence Analysis. Consistent with observations from curriculum training in Consistency Models Geng et al. (2024), MF training and fine-tuning exhibit faster initial convergence, but their improvements diminish at later stages. As shown in Fig. 1, MF achieves a final FID of 3.85 when trained from scratch and 3.93 when fine-tuned from an FM initialization. In contrast, DMF curriculum training progresses more slowly and improves in a stage-wise manner. The discontinuity in FID improvement reflects convergence within each curriculum stage. Despite this slower early progress, DMF achieves superior final performance, with DMF† using the VE-transformed scheduler reaching a comparable FID of 3.36. Notably, DMF curriculum training attains competitive or better FID compared to prior methods trained with substantially larger computational budgets (Tab. 1).

Table 1: **CIFAR-10 comprehensive 1-step FID comparison**. Methods are categorized by initialization strategy. Budgets represent total training epochs, for models initialized with pretrained models, the budget are formatted as "Cost of Pretraining" + "Cost of Tuning". DMF curriculum training attains competitive or better FID compared to prior methods.

Method	Initialization	Budget (Epochs)	FID (↓)
From Scratch			
iCT	Random	8k	**2.83**
MF (ours imple.)	Random	4k	3.85
MF Geng et al. (2025a)	Random	16k	<u>2.90</u>
With DM / FM Initialization			
sCT Lu & Song (2025)	Pretrained DM	4k + 4k	**2.85**
ECT Geng et al. (2024)	Pretrained DM	4k + 1k	3.60
MF (ours imple.)	Pretrained FM	2k + 2k	3.93
DMF Curriculum	Pretrained FM	2k + 2k	3.58
DMF† Curriculum	Pretrained FM	2k + 2k	<u>3.36</u>

† Denotes DMF Curriculum using the VE-transformed scheduler.

Table 2: **Per-batch training cost**. Batch size 1024, 4 H100's. The DMF loss is approximately **1.2**×∼**1.8**× faster than MF as it does not require heavy JVP. The cost of MF is computed without doing classifier-free guidance, so the scale ratio has further been reduced from MF in practice, i.e., excluding 2 extra forward passes.

Dataset	Method	Sec./Batch
CIFAR-10	MF	0.38
	DMF	**0.32**
ImageNet 256×256	MF	3.08
	DMF	**1.71**

Table 3: **CIFAR-10 end-to-end training cost for same data budget and final FID.** DMF Curriculum consists of 2000 epochs of Flow Model training followed by 2000 epochs of DMF tuning, and is compared against MeanFlow trained from scratch for 4000 epochs. In terms of GPU hours measured in H100's under the same data budget, DMF Curriculum **1.3**× faster.

Method	GPU Hours	FID
MF	85.33	3.85
DMF Curriculum	**66.6**	**3.36**

4.2 IMAGENET 256X256, SD-VAE LATENTS

Training Configuration. To evaluate the scalability of our approach, we apply the DMF curriculum to a SiT-XL/2 baseline pretrained for 1400 epochs on the SD-VAE latents, which achieves an FID of 11.52 with 50-step sampling without Classifier-Free Guidance (CFG). Latent spaces encoded via SD-VAE are known to be susceptible to extreme outliers at Consistency Training Dao et al. (2025); Hu et al. (2025b). To mitigate this, we employ a robust Cauchy loss with a high robust value of $c = 0.3$. Plus, instead of using the conditional velocity as $\mathbf{v}_t$, we compute the velocity field using softmax as a kernel Xu et al. (2023) by sampling an additional sub-batch of 127 data from the dataset for each sample, reducing the variance at training. A detail note on this will be provided in Appendix A.2. We utilize a 6-stage $\Delta_i^\dagger$ curriculum ($K = 6$) with a factor decay of $q=4$.

Direct Fine-tuning from Pure MeanFlow. As shown in Tab. 4, DMF† yields rapid convergence in the low-epoch regime, reaching FID of 21.18 with a training budget of 6 epochs, and to 14.53 with increased training budget of 48 epochs. A significant departure from standard practice is our use of a "pure" MeanFlow objective. While existing methods typically rely on a hybrid training mixture of flow-matching and MF (MF) objectives in ratios ranging from 1:1 to 3:1 Geng et al. (2025b;a); Zhang et al. (2025); Hu et al. (2025b), we perform direct fine-tuning using a nearly 100% MF objective (Tab. 5). Similar to the CIFAR-10 experiments, the only regions where we do flow-matching is where the timestep difference is infinitesimal, $t-r < \epsilon_t$.

Stability Analysis and Optimization Limits. Despite the aforementioned robustness measures, we observed an empirical stability ceiling when scaling the DMF† training budget to 96 epochs. Specifically, optimization tends to diverge during the fifth curriculum stage, where the discretization ratio

Table 4: **ImageNet 256×256 curriculum training budget w.r.t FID.** We report the FID of our 1-step DMF curriculum relative to the 1400-epoch pretrained SiT-XL/2 baseline. The FID is computed on samples generated w/o CFG sampling or tuning.

Method	**Training Epochs**	**Rel. Budget**	**Steps**	**FID** $\downarrow$
SiT-XL/2 (Baseline)	1400 (Pretrain)	100.0%	50	11.52
DMF†	1400 + 6	+0.42%	1	21.18
DMF†	1400 + 12	+0.85%	1	18.03
DMF†	1400 + 24	+1.71%	1	16.95
DMF†	1400 + 48	+3.42%	1	14.53
DMF†	1400 + 96	+3.42%	1	294.13

Table 5: **Comparison of Training Paradigms on ImageNet-256.** Unlike majority MF that are trained on hybrid objectives or distillation teachers, our DMF curriculum enables "pure" MF fine-tuning.

Strategy	**Objective Ratio (FM:MF)**	**Training Type**	CFG
Standard MF Geng et al. (2024)	1:1 to 3:1	Direct Training	Yes
RAE-MF Hu et al. (2025b)	3:1	Mid-training Teacher	None
α-Flow Zhang et al. (2025)	1:1	Direct Curriculum	Yes
DMF Curriculum	0:1 (Pure MF)	Direct Curriculum	None

reaches approximately $\Delta_4^\dagger \approx 0.0039 \cdot (t - r)$. We hypothesize that for latent-space datasets, excessively fine discretization introduces a critical trade-off between approximation accuracy and training stability. This suggests that intermediate stages with high discretizations may act as a source of variance that destabilizes the objective. Potential remedies for this instability include an early transition to the MF regime during intermediate stages, or a more granular analysis of model architecture and normalization layers to improve robustness under small step-size regimes.

5 Conclusion and Limitations

In this paper, we show that Discrete MeanFlow (DMF) training curriculum enables high quality one step generation by replacing continuous MeanFlow identities with a staged curriculum of discrete approximations. Experiments on CIFAR-10 and ImageNet 256×256 show that DMF achieves comparable FID with chosen baselines under fixed data budgets, while achieving accelerated convergence as up to $1.8\times$ per batch speedup by avoiding the expensive JVP computations. We further analyze training stability and identify an empirical discretization ceiling in the latent space. When the curriculum becomes too fine, optimization can diverge, revealing a trade off between progressive discretization and training robustness. Future work should focus on enhancing the scalability and robustness of the training curriculum framework through targeted architectural and procedural improvements. Specifically, evaluating it on Representation-learning Autoencoder (RAE) latents Zheng et al. (2025) could determine curriculum's effectiveness and stability on more structured manifolds. Furthermore, introducing a lightweight secondary guidance tuning stage that isolates classifier free guidance allows the primary training phase to remain budget friendly. Finally, investigating architecture-specific robustness, such as the role of normalization layers and weight initialization, remains critical for mitigating the optimization divergence observed in the final stages of latent-space training.

Acknowledgments

We thank Yingchen He for running the experiments, Matthew Niedoba for suggestions on stabilizing the velocity field, and Saeid Naderiparizi for helpful discussions and feedback on the paper.

REFERENCES

Quan Dao, Khanh Doan, Di Liu, Trung Le, and Dimitris Metaxas. Improved training technique for latent consistency models, 2025. URL https://arxiv.org/abs/2502.01441.

Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J. Zico Kolter. Consistency models made easy, 2024. URL https://arxiv.org/abs/2406.14548.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling, 2025a. URL https://arxiv.org/abs/2505.13447.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models, 2025b. URL https://arxiv.org/abs/2512.02012.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

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, 2018. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. URL https://arxiv.org/abs/2204.03458.

Zheyuan Hu, Chieh-Hsin Lai, Yuki Mitsufuji, and Stefano Ermon. Cmt: Mid-training for efficient learning of consistency, mean flow, and flow map models, 2025a. URL https://arxiv.org/abs/2509.24526.

Zheyuan Hu, Chieh-Hsin Lai, Ge Wu, Yuki Mitsufuji, and Stefano Ermon. Meanflow transformers with representation autoencoders, 2025b. URL https://arxiv.org/abs/2511.13019.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018. URL https://arxiv.org/abs/1710.10196.

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, 2024. URL https://arxiv.org/abs/2310.02279.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Kyungmin Lee, Sihyun Yu, and Jinwoo Shin. Decoupled meanflow: Turning flow models into flow maps for accelerated sampling, 2025. URL https://arxiv.org/abs/2510.24474.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models, 2025. URL https://arxiv.org/abs/2410.11081.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed, 2021. URL https://arxiv.org/abs/2101.02388.

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. URL https://arxiv.org/abs/2401.08740.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. URL https://arxiv.org/abs/2307.01952.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

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. URL https://arxiv.org/abs/1409.0575.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models, 2022. URL https://arxiv.org/abs/2202.00512.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models, 2023. URL https://arxiv.org/abs/2310.14189.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models, 2023. URL https://arxiv.org/abs/2303.01469.

Yilun Xu, Shangyuan Tong, and Tommi Jaakkola. Stable target field for reduced variance score estimation in diffusion models, 2023. URL https://arxiv.org/abs/2302.00670.

Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. Alphaflow: Understanding and improving meanflow models, 2025. URL https://arxiv.org/abs/2510.20771.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders, 2025. URL https://arxiv.org/abs/2510.11690.

Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping, 2024. URL https://arxiv.org/abs/2402.19159.

Ge Zhu, Yutong Wen, and Zhiyao Duan. Audio generation through score-based generative modeling: Design principles and implementation, 2026. URL https://arxiv.org/abs/2506.08457.

A APPENDIX

A.1 PROOFS

Proof. To keep the proof of equation 3 elegant, we simplify the notation by setting $r = 0$ starting from the MeanFlow Identity (equation 1), and omit it during our derivations. We will add r back to match the generalized discretized form. Starting from the MF Identity, we have:

$$\begin{aligned}
\mathbf{u}(\mathbf{z}_t, t) &= \mathbf{v}_t(\mathbf{z}_t) - t\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}(\mathbf{z}_t, t) \\
&= \mathbf{v}_t(\mathbf{z}_t) - t\left(\frac{\partial \mathbf{u}(\mathbf{z}_t, t)}{\partial \mathbf{z}_t}\mathbf{v}_t(\mathbf{z}_t) + \frac{\partial \mathbf{u}(\mathbf{z}_t, t)}{\partial t}\right) \\
&= \mathbf{v}_t(\mathbf{z}_t) - t\ \left(\ \mathcal{J}_{\mathbf{z}_t}\mathbf{v}_t(\mathbf{z}_t, t) + \partial_t \mathbf{u}(\mathbf{z}_t, t)\ \right) \\
&= \mathbf{v}_t(\mathbf{z}_t) - t\left(\lim_{\Delta\to 0}\frac{\mathbf{u}(\mathbf{z}_t, t) - \mathbf{u}(\mathbf{z}_t - \mathbf{v}_t\Delta, t)}{\Delta} + \lim_{\Delta\to 0}\frac{\mathbf{u}(\mathbf{z}_t, t) - \mathbf{u}(\mathbf{z}_t, t-\Delta)}{\Delta}\right) \\
&\qquad \text{(merge partial limits into the total derivative along the trajectory } \mathbf{z}_t(t)\text{)} \\
&= \mathbf{v}_t(\mathbf{z}_t) - t\left(\lim_{\Delta\to 0}\frac{\mathbf{u}(\mathbf{z}_t, t) - \mathbf{u}(\mathbf{z}_t - \mathbf{v}_t\Delta, t-\Delta)}{\Delta}\right)
\end{aligned} \tag{6}$$

Multiply by Δ to clear the denominator, both sides.

$$\lim_{\Delta\to 0}\left[\mathbf{u}(\mathbf{z}_t, t)\cdot\Delta\right] = \lim_{\Delta\to 0}\left[\mathbf{v}_t(\mathbf{z}_t)\cdot\Delta - t\cdot(\mathbf{u}(\mathbf{z}_t, t) - \mathbf{u}(\mathbf{z}_t - \mathbf{v}_t\Delta, t-\Delta))\right]$$

Move $\mathbf{u}(\mathbf{z}_t, t)$ to the L.H.S.

$$\lim_{\Delta\to 0}\left[\mathbf{u}(\mathbf{z}_t, t)\cdot(\Delta + t)\right] = \lim_{\Delta\to 0}\left[\mathbf{v}_t(\mathbf{z}_t)\cdot\Delta + t\cdot\mathbf{u}(\mathbf{z}_t - \mathbf{v}_t\Delta, t-\Delta)\right]$$

Divide both sides $(\Delta + t)$.

$$\mathbf{u}(\mathbf{z}_t, t) = \lim_{\Delta\to 0}\frac{\mathbf{v}_t(\mathbf{z}_t)\cdot\Delta + t\cdot\mathbf{u}(\mathbf{z}_t - \mathbf{v}_t\Delta, t-\Delta)}{\Delta + t}$$

Bring back r, we get the final discretized form.

$$\mathbf{u}(\mathbf{z}_t, r, t) = \lim_{\Delta\to 0}\frac{\mathbf{v}_t(\mathbf{z}_t)\cdot\Delta + (t-r)\cdot\mathbf{u}(\mathbf{z}_t - \mathbf{v}_t\Delta, r, t-\Delta)}{\Delta + t - r}$$

□

A.2 ALGORITHM OVERVIEW

Algorithm 1 Discrete MeanFlow (DMF) Curriculum Training

Input: Pretrained flow model $\mathbf{v}_\phi$, dataset $\mathcal{D}$, total stages K, decay factor q, robust value c, sub-batch size B_{sub}, `LogitNormal` hyperparams P_{mean}, P_{std}, training epochs per stage N_{epochs}.

Initialize: Model $\mathbf{u}_\theta \leftarrow \mathbf{v}_\phi$.
for $i = 0$ **to** $K-1$ **do**
  for $n = 0$ **to** $N_{\text{epochs}} - 1$ **do**
    Sample $\mathbf{z}_0 \sim \mathcal{D}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t, r \sim \texttt{LogitNormal}(P_{\text{mean}}, P_{\text{std}})$.
    Compute $\mathbf{z}_t \leftarrow (1-t)\mathbf{z}_0 + t\epsilon$, $\mathbf{z}_r \leftarrow (1-r)\mathbf{z}_0 + r\epsilon$.

    Compute velocity field.
    if $\mathcal{D}$ is ImageNet 256×256 **then**
      Sample subset $\mathcal{X}_{sub} \leftarrow \{\mathbf{x}_0^{(k)}\}_{k=1}^{B_{\text{sub}}}$ from the same class as $\mathbf{x}_0$.
      Include current sample in the reference set $\mathcal{X}_{sub} \leftarrow \mathcal{X}'_{sub} \cup \{\mathbf{x}_0\}$.
      Compute normalized weights via softmax over $B_{\text{sub}}+1$ samples.
      for $k = 0$ **to** B_{sub} **do**
        $w_k \leftarrow \frac{\exp\left(-\|\mathbf{z}_t-(1-t)\mathbf{x}_0^{(k)}\|^2/(2t^2)\right)}{\sum_{j=0}^{B_{\text{sub}}}\exp\left(-\|\mathbf{z}_t-(1-t)\mathbf{x}_0^{(j)}\|^2/(2t^2)\right)}$.
      end for
      Compute stable reference: $\bar{\mathbf{x}}_0 \leftarrow \sum_{k=0}^{B_{\text{sub}}} w_k \mathbf{x}_0^{(k)}$.
      $\mathbf{v}_t \leftarrow \frac{\mathbf{z}_t - \bar{\mathbf{x}}_0}{t}$ {Stable target field Xu et al. (2023)}.
    else
      $\mathbf{v}_t \leftarrow \epsilon - \mathbf{z}_0$.
    end if

    Compute $\mathbf{u}_{\text{target}}$.
    if $i = 0$ **then**
      $\mathbf{u}_{\text{target}} \leftarrow \mathbf{v}_t$.
    else if $i = K-1$ **then**
      Compute _, dudt $\leftarrow \texttt{jvp}((\mathbf{z}_t, r, t), (\mathbf{v}_t, 0, 1))$.
      $\mathbf{u}_{\text{target}} \leftarrow \text{sg}\{\mathbf{v}_t - (t-r)\cdot\text{dudt}\}$.
    else
      Compute $\Phi(t) \leftarrow t/(1-t), \Phi(r) \leftarrow r/(1-r)$.
      Compute $\Delta_i^\dagger \leftarrow t - \Phi^{-1}(\Phi(t) - 1/q^i \cdot (\Phi(t) - \Phi(r)))$.
      $\mathbf{u}_{\text{target}} \leftarrow \text{sg}\left\{\frac{1}{(\Delta_i^\dagger + t - r)} \cdot \left[\mathbf{v}_t\Delta_i^\dagger + \mathbf{u}_\theta(\mathbf{z}_t - \mathbf{v}_t\Delta_i^\dagger, t-\Delta_i^\dagger)(t-r)\right]\right\}$
    end if

    if $\mathcal{D}$ is ImageNet 256×256 **then**
      Loss $\leftarrow$ Cauchy$(\cdot, c)$.
    else
      Loss $\leftarrow$ Adaptive$(\cdot, c)$ Geng et al. (2025a).
    end if

    $\mathcal{L}(\theta) \leftarrow \text{Loss}(\mathbf{u}_\theta(\mathbf{z}_t, r, t), \mathbf{u}_{\text{target}})$
    Update θ via gradient descent
  end for
end for
Return: Optimized 1-step model $\mathbf{u}_\theta$

A.3 CONFIGURATIONS

Table 6: CIFAR-10, hyperparameter configurations across different experimental setups.

Hyperparameter	**FM Pretrain**	**MF (Scratch)**	**MF (Fine-tune)**	**DMF† curriculum**
Batch Size	256	1024	1024	1024
Training Epochs	2000	4000	2000	2000
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	2×10^{-4}	6×10^{-4}	6×10^{-4}	6×10^{-4}
EMA Rate	0.999	0.999	0.999	0.999
Network Dropout	0.2	0.2	0.2	0.35
Stages (K)	N/A	N/A	N/A	10
Decay Factor (q)	N/A	N/A	N/A	2
ϵ_t	N/A	N/A	N/A	10^{-6}
Loss Function	Mean Squared Error	Adaptive L_p	Adaptive L_p	Adaptive L_p
Adaptive norm_p	N/A	0.75	0.75	0.75
Adaptive c	N/A	0.001	0.001	0.001
Logitnormal P_{mean}	-1.2	-2.0	-2.0	-2.0
Logitnormal P_{std}	1.2	2.0	2.0	2.0
Probability t equal r	N/A	0.25	0.25	0

Table 7: ImageNet 256×256 hyperparameter configurations across different experimental setups.

Hyperparameter	**DMF† 6ep**	**DMF† 12ep**	**DMF† 24ep**	**DMF† 48ep**	**DMF† 96ep**
Batch Size	1024	1024	1024	1024	1024
Training Epochs	6	12	24	48	96
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Learning Rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}	1×10^{-4}
EMA Rate	0.995	0.999	0.999	0.9995	0.9995
Network Dropout	0.0	0.0	0.0	0.0	0.0
Stages (K)	6	6	6	6	6
Decay Factor (q)	4	4	4	4	4
ϵ_t	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Loss Function	Cauchy	Cauchy	Cauchy	Cauchy	Cauchy
Robust c	0.3	0.3	0.3	0.3	0.3
Logitnormal P_{mean}	-0.4	-0.4	-0.4	-0.4	-0.4
Logitnormal P_{std}	1.0	1.0	1.0	1.0	1.0
Prob. $t = r$	0.0	0.0	0.0	0.0	0.0

A.4 UNCURATED SAMPLES CIFAR-10

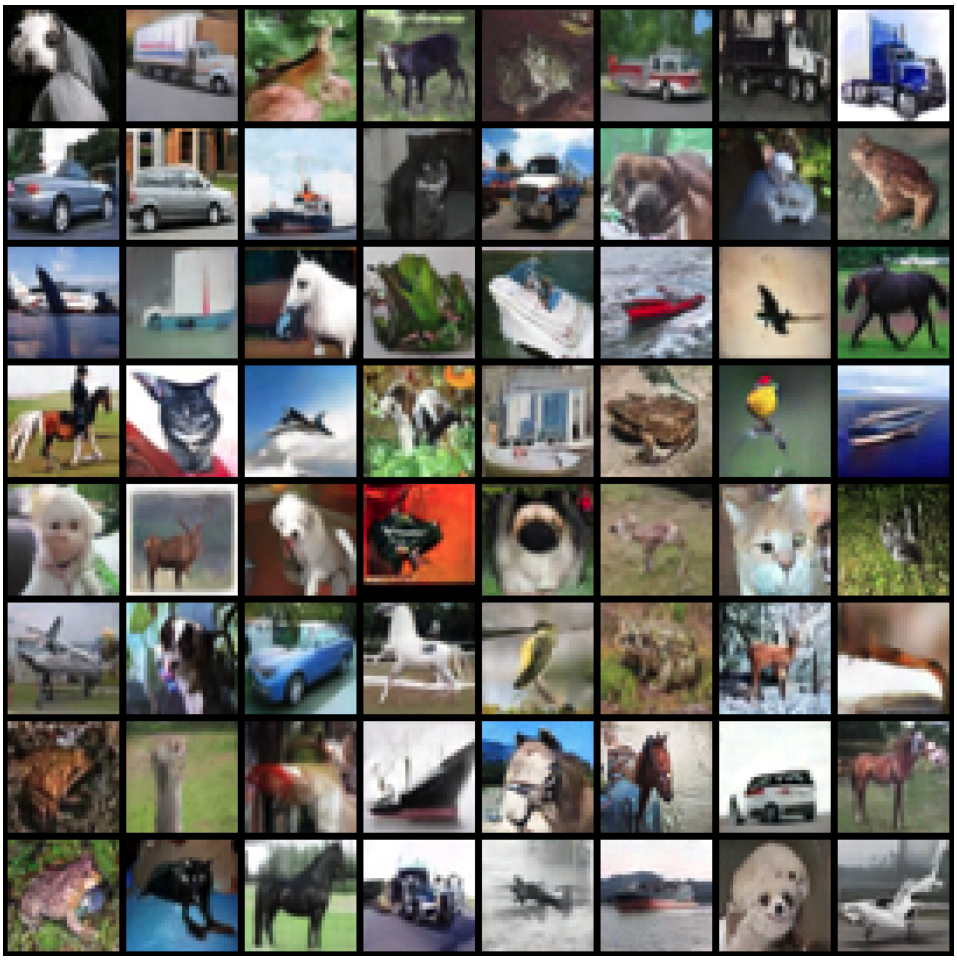

Figure 2: DMF† on CIFAR-10, FID=3.36.

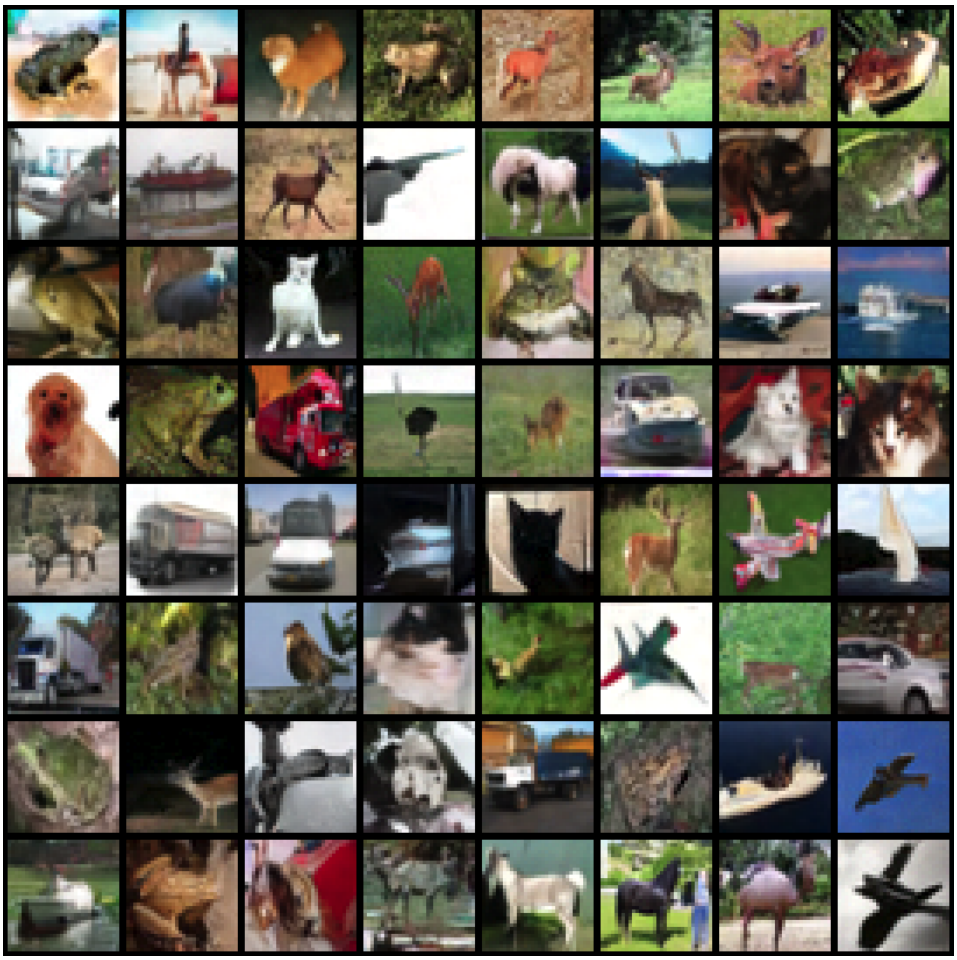

Figure 3: DMF† on CIFAR-10, FID=3.36.

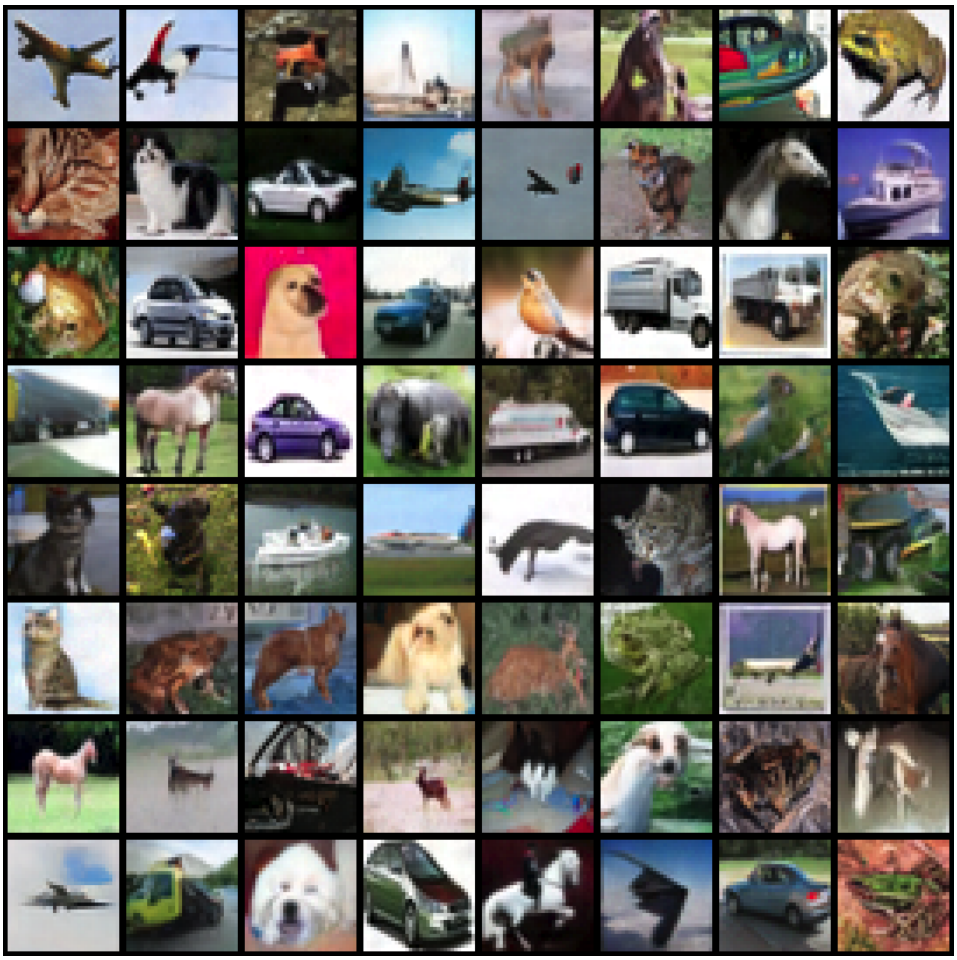

Figure 4: DMF† on CIFAR-10, FID=3.36.