RECTIFIED MEANFLOW: OVERCOMING THE CURVATURE BOTTLENECK IN MEANFLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-based models are powerful generators, but sampling typically requires costly numerical integration along curved generative trajectories. MeanFlow offers a promising framework to few-step generation by directly learning mean-velocity fields. However, we find that the highly curved generative trajectories of existing models induce a noisy loss landscape, severely bottlenecking convergence and model quality. We leverage a fundamental geometric principle to overcome this: mean-velocity estimation is drastically simpler along straight paths. Building on this insight, we propose **Rectified MeanFlow**, a *self-distillation* approach that learns the mean-velocity field over a straightened velocity field, induced by rectified couplings from a pretrained model. By smoothing the optimization landscape, our method achieves strong few-step generation performance. Under the same pretrained initialization and matched end-to-end post-training budgets, including coupling generation, Re-MeanFlow reduces guided one-step FID from 7.48 to 3.03 on ImageNet-512^2. In a controlled text-to-image proof of concept using Cosmos-Predict2-2B, it improves GenEval from 0.6331 to 0.7385 at 4 NFE, compared with approximately 0.80 for the 70-NFE teacher. Together, these results highlight the synergy between trajectory rectification and mean-velocity modelling as an effective route to compute-efficient few-step generation.

1 INTRODUCTION

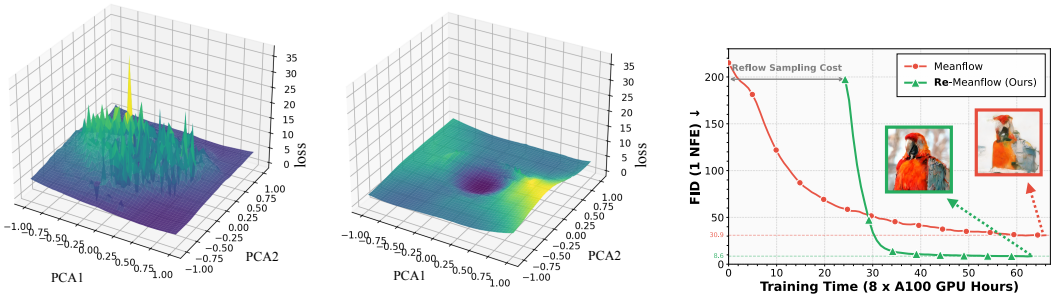
Flow models (Lipman et al., 2022; Liu et al., 2022) and diffusion models (Song & Ermon, 2019; Sohl-Dickstein et al., 2015) have become a central paradigm in generative modeling, enabling a wide range of applications across various data domains (Ho et al., 2022; Rombach et al., 2022; Zhang et al., 2025; He et al., 2023; 2024; Xu et al., 2025). Compared with earlier paradigms such as GANs (Goodfellow et al., 2020) and Normalizing Flows (Rezende & Mohamed, 2015; Zhai et al., 2024), these models offer stable training and superior fidelity, but at the cost of expensive sampling. A major obstacle to aggressive step reduction is the geometry of the learned transport. Under standard independent noise–data couplings, conditional velocity averaging can induce a nonlinear marginal velocity field and *curved* ODE trajectories, making coarse discretizations inaccurate.

This drawback has motivated works (Song et al., 2023; Frans et al., 2024; Geng et al., 2025; Zhou et al., 2025) to bypass ODE integration by directly learning time-indexed flow-map predictions. Among these methods, MeanFlow (Geng et al., 2025) is particularly appealing: it predicts the mean velocity between two time points and can therefore support direct finite-time transport without requiring perfectly straight trajectories. **MeanFlow removes straightness as a requirement, but not necessarily curvature as a learning difficulty.** We observe that directly fine-tuning a pretrained flow model with the MeanFlow objective can converge slowly when the inherited trajectories remain highly curved.

In this work, we identify trajectory curvature as a key bottleneck in mean-velocity learning. Motivated by this insight, we introduce **Re**ctified **MeanFlow** (**Re-MeanFlow**), a lightweight *self-distillation* approach that trains MeanFlow on *rectified* couplings from a pretrained model, yielding straighter trajectories and simplifying mean-velocity learning. Fig. 1 illustrates this contrast: MeanFlow learning on the original (*unrectified*) trajectories exhibits a rugged loss landscape[1] with sharp

[1]We slightly abuse the term *loss landscape*. In optimization theory, it usually denotes the objective as a function of model parameters. Here, we instead fix the model parameters and visualize $\mathcal{L}_{\mathrm{MF}}$ as a function of

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) MeanFlow Loss Landscape

(b) Re-MeanFlow (Ours) Loss Landscape

(c) Comparison of Convergence (Both initialized from the pretrained EDM2-S)

Figure 1: **Straightening flows smooths the loss landscape for MeanFlow modelling. (a)** MeanFlow exhibits a sharply peaked and irregular loss landscape, reflecting the difficulty of mean-velocity learning. **(b)** By learning mean velocities over straighter trajectories induced by rectified couplings, Re-MeanFlow (**Ours**) yields a smoother and more regular objective. **(c)** Re-MeanFlow achieves faster convergence and stronger one-step generation under matched end-to-end post-training budgets, including coupling generation. *All plots are from ImageNet-*512^2*, with both methods initialised from the same pretrained EDM2-S model.*

spikes, whereas Re-MeanFlow yields a substantially smoother regression surface. Compared with direct MeanFlow fine-tuning from the same pretrained model, Re-MeanFlow converges faster and achieves better one-step sample quality under matched end-to-end post-training budgets. Importantly, Re-MeanFlow is data-free: it constructs rectified couplings from model-generated samples, without access to the original training dataset.

We first evaluate Re-MeanFlow on ImageNet (Deng et al., 2009). With the same pretrained initialisation and end-to-end budgets, Re-MeanFlow consistently outperforms direct MeanFlow fine-tuning. For example, it reduces guided one-step FID from 9.42 to 2.87 at 64^2 and from 7.48 to 3.03 at 512^2. Comparisons with representative one-step post-training baselines further demonstrate a favourable trade-off between sample quality and measured end-to-end post-training cost. We additionally conduct a controlled proof of concept for text-to-image distillation using Cosmos-Predict2-2B (Ali et al., 2025). At 4 NFE, Re-MeanFlow improves GenEval from 0.6331 to 0.7385 ($+0.1054$), without score-distillation or adversarial objectives. Although a gap remains to the 70-NFE teacher, which achieves approximately 0.80, this result provides initial evidence that trajectory rectification also facilitates mean-velocity learning in text-conditioned generation.

Together, these results show that trajectory geometry remains consequential even when the student predicts a finite-time mean velocity, and that Re-MeanFlow provide a simple route to more efficient one- and few-step post-training.

2 Related Work

2.1 Transport Geometry and Trajectory Rectification for Flow Models.

The coupling used in flow matching (Lipman et al., 2022; Liu et al., 2022) shapes the induced marginal velocity field and its trajectory geometry. OT-inspired objectives and minibatch transport couplings seek shorter or straighter paths (Onken et al., 2021; Tong et al., 2023; Pooladian et al., 2023), while Rectified Flow (Liu et al., 2022) uses reflow to replace independent noise–data pairs with model-induced couplings and progressively reduce trajectory curvature. InstaFlow (Liu et al., 2023) scales this principle to one-step text-to-image generation by training an instantaneous-velocity student on reflowed pairs. These methods exploit straightness chiefly to reduce discretization error when integrating instantaneous velocities. Re-MeanFlow uses rectification for a different role: simplifying finite-time mean-velocity learning.

the intermediate state $\mathbf{z}_t$ over a local PCA plane. This input-space view is useful for our purpose because it reveals how the coupling geometry changes the local sensitivity and regularity of the mean-velocity regression problem.

2.2 Flow-Map Learning and MeanFlow Post-Training.

Flow-map methods bypass iterative ODE integration by directly learning finite-time transitions. Consistency Models (Song et al., 2023) map intermediate states to trajectory endpoints, while Consistency Trajectory Models (Kim et al., 2023) and Flow Map Matching (Boffi et al., 2024) extend this idea to transitions between arbitrary time pairs. MeanFlow (Geng et al., 2025) instead parameterizes each transition by its interval-average. Recent work improves MeanFlow-style learning at different levels. CMT (Hu et al., 2025) introduces trajectory-aligned mid-training before flow-map post-training; AlphaFlow (Zhang et al., 2026) employs a flow-to-MeanFlow curriculum to alleviate optimization conflict; Decoupled MeanFlow (Lee et al., 2026) repurposes pretrained flow models through localized second-time conditioning; and Improved MeanFlow (Geng et al., 2026) reformulates the training objective and guidance mechanism. Re-MeanFlow addresses a different axis: the geometry of the training coupling. It retains the MeanFlow parameterization while replacing independent pairings with teacher-induced rectified couplings.

3 Efficient Mean-Velocity Modeling by Taming the Curvature Bottleneck

3.1 One-Step Flow-Based Generative Modeling via MeanFlow

Flow-based generative models transport a prior $p_{\mathbf{z}}$ (taken as $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in this work) to the data distribution $p_{\mathbf{x}}$ by learning a time-dependent velocity field $v : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ that induces a path of intermediate densities $\{p_t\}_{t\in[0,1]}$ in $\mathbb{R}^d$. Concretely, given a data-noise coupling $p_{\mathbf{xz}}$, let $(\mathbf{x}, \mathbf{z}) \sim p_{\mathbf{xz}}$ and define the linear interpolation as:

$$\mathbf{z}_t = (1 - t)\mathbf{x} + t\mathbf{z}.$$

Let p_t denote the distribution of $\mathbf{z}_t$. The coupling $p_{\mathbf{xz}}$ together with the interpolation, therefore, specifies the probability path $\{p_t\}_{t\in[0,1]}$.

The associated velocity field $v(\mathbf{z}_t, t)$ is defined as the vector field whose continuity equation

$$\partial_t p_t + \nabla \cdot (p_t v(\cdot, t)) = 0$$

matches this density evolution. Along the interpolation trajectory, the instantaneous velocity is given by

$$\frac{d}{dt}\mathbf{z}_t = \mathbf{z} - \mathbf{x}.$$

Flow Matching (FM) (Lipman et al., 2022; Liu et al., 2022) models the time-dependent velocity field with a neural vector field $v_\theta(\cdot, t)$ and trains it by regressing the trajectory velocity induced by the chosen coupling:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim p_{\mathbf{xz}},\, t\sim p(t)} \left[\left\| \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z}_t - v_\theta(\mathbf{z}_t, t) \right\|_2^2 \right] \tag{1}$$

Once v_θ is learned, a new sample $\mathbf{x}$ can be generated by solving the ODE for $\mathbf{z} \sim p_{\mathbf{z}}$ using a numerical solver:

$$\mathbf{x}_\theta(\mathbf{z}) = \mathbf{z}_0 = \mathbf{z} - \int_0^1 v_\theta(\mathbf{z}_\tau, \tau) d\tau. \tag{2}$$

MeanFlow (Geng et al., 2025) instead parameterizes the *mean*-velocity between two time points $r < t$:

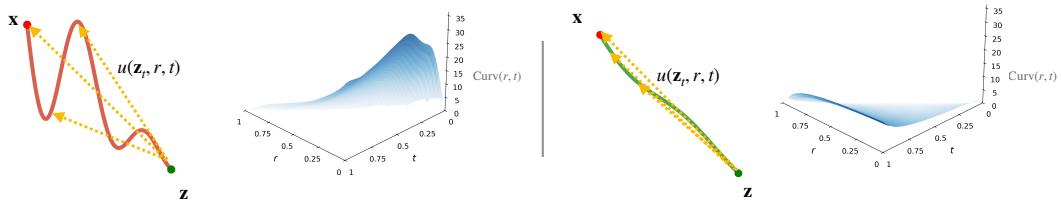
$$u(\mathbf{z}_t, r, t) \triangleq \frac{1}{t - r} \int_r^t v(\mathbf{z}_\tau, \tau)\, d\tau, \tag{3}$$

To train a neural network $u_\theta(\mathbf{z}_t, r, t)$ to approximate this mean-velocity, MeanFlow derives an implicit training target that connects the mean-velocity, the instantaneous velocity $v(\mathbf{z}_t, t)$, and the time derivative of u_θ:

$$\mathcal{L}_{MF}(\theta) = \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim p_{\mathbf{xz}},\, (r,t)\sim p_{r,t}} \left\| u_\theta(\mathbf{z}_t, r, t) - \text{sg}(u_{\text{tgt}}) \right\|_2^2 \tag{4}$$

$$\text{with } u_{\text{tgt}} = v(\mathbf{z}_t, t) - (t - r)\frac{d}{dt} u_\theta(\mathbf{z}_t, r, t). \tag{5}$$

Here, $\text{sg}(\cdot)$ denotes the stop-gradient operator. Once u_θ is trained, sampling reduces to a single evaluation $u_\theta(\mathbf{z}, 0, 1)$, avoiding the numerical integration required by Flow Matching (cf. Eq. 2).



(a) MeanFlow Models Mean-Velocity on High-Curvature Trajectories

(b) Re-MeanFlow (Ours) Models Mean-Velocity on Rectified (Straightened) Trajectories

Figure 2: **Method. (a)** MeanFlow learns the mean-velocity $u(\mathbf{z}_t, r, t)$ on highly curved trajectories, which makes optimization difficult. In contrast, **(b) Re-MeanFlow (ours)** models $u(\mathbf{z}_t, r, t)$ on a rectified coupling that induces substantially straighter generative trajectories, simplifying the underlying vector field and enabling faster and more stable training. $\mathrm{Curv}(r, t)$ denotes the curvature proxy measuring trajectory curvature between time points r and t (see Sec. 3.2.1).

3.2 Re-MeanFlow

Crucially, the generative trajectories underlying MeanFlow are determined by the data–noise coupling distribution $p_{\mathbf{xz}}$. In the absence of prior knowledge about how data and noise should be paired, MeanFlow, like many flow-based methods, adopts the independent coupling

$$p^0_{\mathbf{xz}}(\mathbf{x}, \mathbf{z}) = p_{\mathbf{x}}(\mathbf{x})\, p_{\mathbf{z}}(\mathbf{z}), \tag{6}$$

which is known to induce highly curved generative trajectories (Fig. 2a). Although MeanFlow can represent finite-time transport along such trajectories, we argue that their curvature makes the mean-velocity target more difficult to learn.

To make this connection explicit, we define the *MeanFlow correction gap*

$$\Delta_{\mathrm{MF}}(r, t) \triangleq \mathbb{E}_{\mathbf{z}_t \sim p_t}\left[\|v(\mathbf{z}_t, t) - u(\mathbf{z}_t, r, t)\|_2\right] = \mathbb{E}_{\mathbf{z}_t \sim p_t}\left[\left\|(t - r)\frac{d}{dt}u(\mathbf{z}_t, r, t)\right\|_2\right]. \tag{7}$$

Thus, $\Delta_{\mathrm{MF}}(r, t)$ is exactly the expected magnitude of the derivative correction that distinguishes the MeanFlow target from the instantaneous velocity in Eq. 5. MeanFlow does not predict this correction as a separate output, but the correction is embedded in its implicit regression target.

For an ideal straight, constant-velocity trajectory, $v(\mathbf{z}_\tau, \tau)$ remains unchanged over $[r, t]$. The mean and instantaneous velocities then coincide, $\Delta_{\mathrm{MF}}(r, t) = 0$, and the correction in Eq. 5 vanishes. Perfect straightness is not required, however: for a nearly rectified trajectory whose velocity changes only slightly, $\Delta_{\mathrm{MF}}(r, t)$ remains small. Therefore, any reduction in trajectory curvature can reduce the nontrivial correction that MeanFlow must account for. Empirically, we find that this also produces a smoother and more regular loss landscape, as analysed in Sec. 3.2.2.

Motivated by this observation, we introduce **Rectified MeanFlow** (**Re-MeanFlow**), which applies the standard MeanFlow objective to a rectified coupling. Rectified Flow and MeanFlow play complementary roles: Rectified Flow simplifies the geometry of the underlying transport, whereas MeanFlow learns its finite-time average velocity to avoid numerical integration.

Following Rectified Flow (Liu et al., 2022), we perform one reflow step using a pretrained flow model v_ϕ trained under the independent coupling. This induces the rectified coupling

$$p^1_{\mathbf{xz}}(\mathbf{x}, \mathbf{z}) = p_{\mathbf{z}}(\mathbf{z})\, \delta\left(\mathbf{x} - \left[\mathbf{z} - \int_0^1 v_\phi(\mathbf{z}_\tau, \tau)\, d\tau\right]\right), \tag{8}$$

which preserves the correspondence between each source noise $\mathbf{z}$ and its generated endpoint $\mathbf{x}$.

Then we initialize the student from the same pretrained model and optimize the MeanFlow objective over this coupling:

$$\mathcal{L}_{\mathrm{ReMF}}(\theta) = \mathbb{E}_{(\mathbf{x}, \mathbf{z}) \sim p^1_{\mathbf{xz}},\, (r, t) \sim p_{r,t}}\left[\|u_\theta(\mathbf{z}_t, r, t) - \mathrm{sg}(u_{\mathrm{tgt}})\|_2^2\right], \tag{9}$$

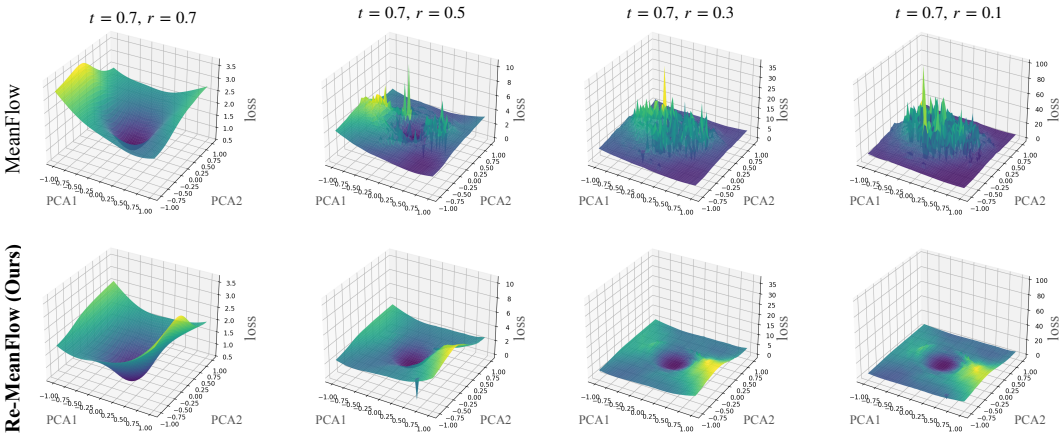


Figure 3: **Loss landscape on a PCA plane of $\mathbf{z}_t$.** We visualize the MeanFlow Loss as a function of the input $\mathbf{z}_t$ on a 2D grid spanned by the top two PCA directions of a batch of $\mathbf{z}_t$ samples. We fix $t = 0.7$ and sweep $r \in \{0.7, 0.5, 0.3, 0.1\}$ (left to right), where larger $t - r$ corresponds to a harder, longer-jump (r, t) regression task. **Top (MeanFlow):** the landscape becomes increasingly sharp and irregular as $t - r$ grows. **Bottom (Re-MeanFlow; ours):** the landscape remains substantially smoother and better-conditioned across all r. **Overall, learning mean velocity on straighter trajectories yields a markedly smoother objective, enabling more efficient and stable training.**

Therefore, Re-MeanFlow retains the MeanFlow parameterization, training identity, and one-step sampling rule. Its central intervention is to replace the independent coupling $p^0_{\mathbf{xz}}$ with the teacher-induced rectified coupling $p^1_{\mathbf{xz}}$. Re-MeanFlow benefits whenever rectification reduces $\Delta_{\mathrm{MF}}(r, t)$ and therefore the correction present in the MeanFlow regression target.

3.2.1 Measuring the MeanFlow Correction and Trajectory Curvature

We next estimate the correction gap in Eq. 7 and its directional, curvature-related component. Let $\tilde{u}(\mathbf{z}_t, r, t)$ denote the mean velocity over the interval $[r, t]$. In practice, we approximate it by integrating a learned velocity field v_θ trained under the corresponding coupling:

$$\tilde{u}(\mathbf{z}_t, r, t) = \frac{1}{t - r} \int_r^t v_\theta(\mathbf{z}_\tau, \tau)\, d\tau. \tag{10}$$

This gives the empirical MeanFlow correction gap $\widehat{\Delta}_{\mathrm{MF}}(r, t) = \mathbb{E}_{\mathbf{z}_t \sim p_t} \left[\| v_\theta(\mathbf{z}_t, t) - \tilde{u}(\mathbf{z}_t, r, t) \|_2 \right]$. Unlike a generic trajectory statistic, $\widehat{\Delta}_{\mathrm{MF}}(r, t)$ directly estimates the magnitude of the correction appearing in the MeanFlow target. To isolate the directional component of this gap, we retain an angle-based curvature proxy:

$$\mathrm{Curv}(r, t) = \mathbb{E}_{\mathbf{z}_t \sim p_t} \left[\angle \left(\tilde{u}(\mathbf{z}_t, r, t), v_\theta(\mathbf{z}_t, t) \right) \right], \tag{11}$$

where $\angle(\mathbf{a}, \mathbf{b}) = \arccos\left(\frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2} \right)$. Accordingly, $\widehat{\Delta}_{\mathrm{MF}}(r, t)$ measures the complete correction gap, while $\mathrm{Curv}(r, t)$ isolates its scale-invariant directional component. We visualize $\mathrm{Curv}(r, t)$ over the (r, t) plane in Fig. 2 and find that the rectified couplings used by Re-MeanFlow exhibit substantially lower directional disagreement.

3.2.2 Loss Landscape

To explain the training speedup enabled by learning mean-velocity on straighter trajectories, we visualize in Fig. 3 the loss landscape of $\mathcal{L}_{MF}$ and $\mathcal{L}_{ReMF}$ with respect to the input $\mathbf{z}_t$ for both MeanFlow and Re-MeanFlow. We fix $t = 0.7$ and plot the landscape for different values of r. As $t - r$ increases, the model must predict longer jumps, which makes the regression task more difficult and typically yields a more rugged objective.

To construct these visualizations, we perform PCA on a batch of $\mathbf{z}_t$ samples to obtain the top two principal directions. We then evaluate the loss on a 2D grid spanned by these directions and centered at the sampled $\mathbf{z}_t$, and visualize the resulting surfaces. As shown in Fig. 3, Re-MeanFlow

consistently exhibits a smoother and better-conditioned landscape. In particular, as $t - r$ grows, the MeanFlow objective becomes increasingly sharp and irregular, whereas Re-MeanFlow remains substantially smoother and more stable.

3.3 Training Design

Re-MeanFlow retains the MeanFlow objective and sampling rule, but rectification changes the statistics of the regression target and post-training introduces a new reference-time input r. We adopt four simple design choices to enhance efficient Re-Meanflow distilling.

Pretrained-preserving time conditioning. MeanFlow conditions the student on both (r, t), whereas the pretrained model only receives the current time t. Since $u(\mathbf{z}_t, t, t) = v(\mathbf{z}_t, t)$, the pretrained instantaneous-velocity field provides a natural initialization for the interval-conditioned student. We parameterize time conditioning additively as $\mathrm{emb}(t, r) = \mathrm{emb}_t(t) + \mathrm{emb}_r(r)$, and zero-initialize the new r embedding. For checkpoints whose original time coordinate differs from t, we briefly align $\mathrm{emb}_t(t)$ with the pretrained time embedding. Consequently, $u_{\theta_0}(\mathbf{z}_t, r, t) \approx v_\phi(\mathbf{z}_t, t)$, so post-training begins from the pretrained velocity field and primarily learns the finite-interval correction. Architecture-specific initialization details are provided in the supplementary material.

Milder adaptive normalization. MeanFlow applies an adaptive Pseudo-Huber-style normalization to suppress high-variance regression targets. Because rectified couplings yield more regular targets, we find that the default normalization over-suppresses informative residuals. We therefore retain the same robust loss while reducing its normalization strength, producing stronger gradients on typical rectified pairs without sacrificing robustness to occasional large errors.

Rectification-aware time-pair sampling. The pair (r, t) jointly controls the input noise level and the prediction interval. We observe a loss–time profile similar to that of improved Rectified Flow (Lee et al., 2024a), motivating increased supervision near both trajectory endpoints. We sample the current time from a U-shaped distribution,

$$p(s) \propto \exp(as) + \exp(-as), \qquad s = 2t - 1 \in [-1, 1], \qquad a = 5, \tag{12}$$

and sample the interval length $t - r$ from a sigmoid-normal distribution that emphasizes medium-length transitions.

Avoid high-variance (r, t) region. A narrow region remains unstable when t is close to 1 and r is small but positive. Rather than discarding these samples, we redirect them to the clean endpoint:

$$\bar{r} = \begin{cases} 0, & t > \tau_t \text{ and } 0 < r < \tau_r, \\ r, & \text{otherwise.} \end{cases} \tag{13}$$

This replaces an ambiguous partial-denoising target with a direct high-noise-to-data transition and allocates additional supervision to the one-step prediction $u_\theta(\mathbf{z}, 0, 1)$. A detailed analysis and parameter settings are provided in Appendix B.3.

4 Experiments

4.1 Class-conditional generation on ImageNet

Setting. We conduct experiments on ImageNet (Deng et al., 2009) at 64^2 resolution in pixel space and at 256^2 and 512^2 resolutions in latent space (Rombach et al., 2022). We initialize Re-MeanFlow from a pretrained flow or diffusion model. For ImageNet-64^2 and ImageNet-512^2, we initialize Re-MeanFlow from the pretrained EDM2-S model (Karras et al., 2024b). For ImageNet-256^2, we initialize from the pretrained SiT-XL model (Ma et al., 2024). To rectify the trajectory, we generate 5M data-noise couplings using the default procedures described in the corresponding papers. Tab. 1 compares the final generation quality of Re-MeanFlow with representative multi-step reference models and prior one-step or few-step generators.

One-step generation quality. As shown in Tab. 1, Re-MeanFlow achieves competitive one-step generation quality across all three ImageNet resolutions. The qualitative results are shown in Fig. 4. On the EDM2-S backbone (Karras et al., 2024b), Re-MeanFlow achieves superior performance at

Table 1: **Class-conditional generation on ImageNet.** All results use classifier-free guidance (CFG) when supported; "×2" indicates that CFG doubles the effective NFE. † denotes methods initialized from, or directly comparable to. **Re-MeanFlow achieves competitive one-step FID across all three resolutions while operating as a self-distillation method.**

(a) ImageNet 64^2

Method	NFE	FID
Diffusion models		
EDM2-S†	63 × 2	1.58
+ Autoguidance	63 × 2	1.01
EDM2-XL	63 × 2	1.33
Few-step models		
2-rectified flow++	1	4.31
iCT	1	4.02
ECD-S†	1	3.30
sCD-S†	1	2.97
TCM†	1	2.88
AYF†	1	2.98
Re-MeanFlow (ours)†	**1**	**2.87**

(b) ImageNet 256^2

Method	NFE	FID
Diffusion models		
ADM	250 × 2	10.94
DiT-XL	250 × 2	2.27
SiT-XL†	250 × 2	2.06
SiT-XL + REPA	250 × 2	1.42
Few-step models		
iCT	1	34.6
SM	1	10.6
iSM	1	5.27
iMM	1 × 2	7.77
MeanFlow	1	3.43
Re-MeanFlow (ours)†	**1**	**3.41**

(c) ImageNet 512^2

Method	NFE	FID
Diffusion models		
EDM2-S†	63 × 2	2.23
+ Autoguidance	63 × 2	1.34
EDM2-XXL	63 × 2	1.81
Few-step models		
ECT†	1	9.98
ECD†	1	8.47
CMT†	1	3.38
sCT-S	1	10.13
sCD-S†	1	3.07
AYF†	1	3.32
Re-MeanFlow (ours)†	**1**	**3.03**

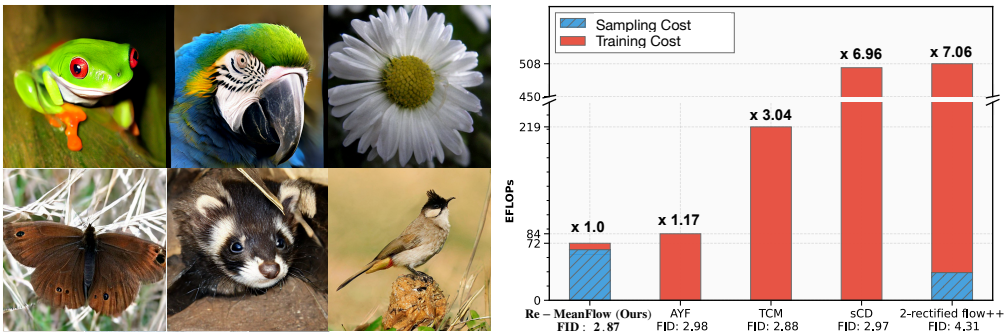


Figure 4: **Left:** Visualization of Re-MeanFlow one-step generation (NFE=1) on ImageNet-512^2. **Right:** Total compute comparison on ImageNet-64^2 for methods in our main FID table (Tab. 1). **Re-MeanFlow achieves the lowest end-to-end compute.**

both 64^2 and 512^2 resolutions. On ImageNet-64^2, Re-MeanFlow outperforms the closely related 2-rectified flow++ (Lee et al., 2024a) by **33.4%** in FID, and slightly improves over previous strong one-step baselines (Lu & Song, 2024; Lee et al., 2024b; Sabour et al., 2025). On ImageNet-512^2, Re-MeanFlow also delivers strong one-step image quality, achieving a **9%** FID gain over AYF (Sabour et al., 2025) and outperforming strong consistency distillation methods (Sabour et al., 2025; Hu et al., 2025). On the SiT backbone (Ma et al., 2024) for ImageNet-256^2, Re-MeanFlow matches the same FID as MeanFlow (Geng et al., 2025) training from scratch despite being trained *without* real-image supervision and relying solely on synthesized samples. Additional multi-step evaluations and coupling-budget ablations are provided in Appendices C.5 and C.6, respectively.

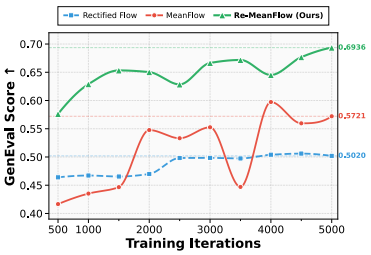
End-to-end Post-Training Efficiency Tab. 2 isolates the complementary roles of trajectory rectification and finite-time mean-velocity modeling under matched wall-clock budgets. Direct MeanFlow supports one-step transport through mean-velocity prediction, but must learn this quantity on the original high-curvature coupling and therefore converges slowly. Rectified Flow simplifies the transport geometry, but its instantaneous-velocity objective remains inaccurate for one-step generation. Re-MeanFlow combines both advantages, learning a finite-time mean velocity over substantially straighter trajectories. Under the same end-to-end budgets, it improves guided FID from 9.42 to 2.87, 14.23 to 3.41, and 7.48 to 3.03 at 64^2, 256^2, and 512^2, respectively. Its large

Table 2: **Matched post-training efficiency on ImageNet.** Total cost includes coupling generation and training. FID is reported as **CFG (no CFG)**. All wall-clock hours are measured on the same 8×A100 setup. **Re-MeanFlow consistently achieves substantially better final FID than direct MeanFlow under the same end-to-end budget.**

Res.	Method	Coupling gen.	Train	Total	FID ↓ CFG (no CFG)
64^2	MeanFlow	–	66 h	66 h	9.42 (26.73)
	Rectified Flow	33 h	33 h	66 h	– (67.4)
	Re-MeanFlow (ours)	33 h	33 h	66 h	**2.87 (7.28)**
256^2	MeanFlow	–	61 h	61 h	14.23 (30.70)
	Rectified Flow	28 h	33 h	61 h	– (53.3)
	Re-MeanFlow (ours)	28 h	33 h	61 h	**3.41 (8.56)**
512^2	MeanFlow	–	66 h	66 h	7.48 (30.92)
	Rectified Flow	33 h	33 h	66 h	– (87.8)
	Re-MeanFlow (ours)	33 h	33 h	66 h	**3.03 (8.63)**



(a) 4-NFE Visualizations



(b) Comparison of Convergence (Both initialized from the pretrained Cosmos-Predict2)

Figure 5: **Controlled text-to-image comparison on Cosmos-Predict2. (a)** 4-NFE samples from Rectified Flow, MeanFlow, and Re-MeanFlow. Re-MeanFlow produces more coherent few-step samples. **(b)** GenEval convergence under the same pretrained initialization. Re-MeanFlow converges consistently faster.

advantage over Rectified Flow in the unguided setting further shows that neither rectification nor mean-velocity modeling alone matches their combination.

Comparison with prior one-step distillation methods. We further place this controlled result in the broader quality–efficiency landscape by estimating the total training and coupling generation cost on ImageNet-64^2, following the protocol of Lee et al. (2024a). As shown in Fig. 4, Re-MeanFlow achieves the lowest estimated end-to-end compute among the considered distillation baselines while retaining a competitive one-step FID of 2.87.

4.2 Text-to-Image Generation

Experimental setting. We further investigate whether the benefits of rectified mean-velocity learning extend to large-scale text-conditioned generation. We conduct experiments on Cosmos-Predict2-2B (Ali et al., 2025) and build on the rCM's(Zheng et al., 2026) FlashAttention-2-compatible Jacobian–vector product implementation.

Table 3: **Text-to-image generation on Cosmos-Predict2.** GenEval is reported at 4 NFE for all student methods; Rectified Flow, MeanFlow, and Re-MeanFlow follow our controlled post-training setup. **Re-MeanFlow achieves the strongest controlled result and substantially narrows the gap to the multi-step teacher.**

Method	NFE	GenEval ↑
Teacher and published references		
Cosmos-Predict2 (teacher)	2×35	0.8019
Controlled post-training comparison		
Rectified Flow	4	0.5060
MeanFlow	4	0.6331
Re-MeanFlow (ours)	**4**	**0.7385**

Because the original image training corpus is unavailable, we construct a common synthetic post-training set for all methods. Following the prompt selection used by DMD (Yin et al., 2024), we sample 50K prompts from LAION and generate two images per prompt using the original 70-NFE Cosmos-Predict2 model, resulting in 100K teacher-generated samples. All controlled methods are initialized from the same pretrained checkpoint and trained on this common corpus. For the rectification-based methods, we retain the correspondence between each generated image and its source noise. The teacher-generation cost is shared across the compared methods and is included when considering end-to-end post-training cost.

To minimize implementation confounds, we adopt the optimization settings reported by rCM with only the changes required by the respective training objectives. We train direct MeanFlow, Rectified Flow, and Re-MeanFlow for 10K iterations using the same training budget.

Controlled comparison. The matched-prompt examples in Fig. 5 reveal a clear qualitative difference among the three controlled methods. Re-MeanFlow produces the sharpest and most compositionally coherent samples, with cleaner object structure and fewer visible artifacts. Rectified Flow produces noticeably blurry outputs, indicating that straighter couplings alone are insufficient for accurate large-step generation when the student predicts only an instantaneous velocity. Direct

MeanFlow recovers more high-frequency detail than Rectified Flow, but its samples are generally less cohesive than those of Re-MeanFlow and contain noticeable structural artifacts.

The GenEval curves exhibit the same pattern. During the initial 5K-iteration, Re-MeanFlow improves substantially faster and maintains a comparatively smooth convergence trajectory. Rectified Flow improves more slowly, while direct MeanFlow exhibits pronounced fluctuations and occasional drops in GenEval as training progresses. This non-monotonic behaviour is consistent with our curvature-bottleneck hypothesis: although MeanFlow provides an appropriate finite-time parameterization, learning it over the original curved transport remains a difficult and unstable regression problem. It also echoes the challenges of scaling continuous-time consistency objectives, including MeanFlow, discussed in rCM (Zheng et al., 2026). Quantitatively, Tab. 3 shows that Re-MeanFlow reaches a 4-NFE GenEval score of 0.7385, compared with 0.6331 for direct MeanFlow and 0.5060 for Rectified Flow. This corresponds to an absolute improvement of 0.1054 over MeanFlow (16.6% relative) and 0.2325 over Rectified Flow.

We emphasize that this experiment is intended as a controlled proof of concept rather than a comprehensive text-to-image benchmark. We do not have access to the original Cosmos-Predict2 training corpus and train only on 100K teacher-generated samples, without incorporating external real-image pairs. Moreover, unlike DMD and rCM (Yin et al., 2024; Zheng et al., 2026), we do not augment the flow-map objective with score-distillation, nor do we introduce an adversarial objective. Consequently, a gap remains to the original teacher, which obtains 0.8019 GenEval using 70 NFEs. Within this deliberately restricted setting, however, the substantial and consistent gains over both direct MeanFlow and Rectified Flow provide initial evidence that Re-MeanFlow can scale beyond ImageNet to large-scale text-to-image tasks. Detailed experimental settings and category-wise GenEval results are provided in Appendices C.7 and C.8, respectively.

4.3 Coupling and Training-Design Ablations

Tab. 4 disentangles the effects of the training data, coupling geometry, and training design. Replacing real endpoints with the same teacher-generated endpoints used by Re-MeanFlow, while independently re-pairing them with noise, changes MeanFlow FID only marginally from 14.76 to 14.55. In contrast, preserving the teacher-induced source-noise correspondence reduces FID to 7.81, showing that the gain arises from the rectified coupling rather than from synthetic endpoints alone.

Table 4: **Coupling and training-design ablations on ImageNet-512^2.**

Configuration	MeanFlow FID ↓	Re-MeanFlow FID ↓
Endpoint and coupling control		
Real endpoints / independent pairing	14.76	–
Generated endpoints / independent pairing	14.55	–
Generated endpoints / preserved pairing	–	**7.81**
Shared training design (cumulative)		
Base recipe	14.76	7.81
(a) + Milder normalization	14.45	7.22
(b) + Time conditioning	10.91	4.60
(c) + Time-pair sampling	9.87	3.71
(d) + Avoid high-variance (r, t) region	**7.48**	**3.03**

The shared training design benefits both methods: MeanFlow improves from 14.76 to 7.48, while Re-MeanFlow improves from 7.81 to 3.03. Nevertheless, Re-MeanFlow remains substantially better at every matched configuration. Thus, the proposed training decisions provide broadly useful stabilization, while rectified coupling geometry accounts for the remaining performance advantage.

5 Conclusion

We introduced **Re**ctified **MeanFlow**, a lightweight self-distillation approach that trains the standard MeanFlow objective on teacher-induced rectified couplings, without access to the teacher's original training images. Our analysis connects trajectory geometry to the correction between instantaneous and interval-average velocities, explaining why even approximately rectified trajectories can simplify finite-time mean-velocity learning.

Across multiple ImageNet resolutions, Re-MeanFlow achieves competitive one-step generation quality while converging substantially faster than direct MeanFlow under matched end-to-end post-training budgets. A controlled Cosmos-Predict2 study further demonstrates consistent improvements over both direct MeanFlow and Rectified Flow, providing initial evidence that the approach extends to text-conditioned generation. Although a gap remains to the multi-step teacher, our results establish coupling geometry as an effective and complementary axis for efficient one- and few-step post-training.

AI USE STATEMENT

We used generative AI tools for writing assistance, including improving the grammar, clarity, and presentation of the manuscript. All AI-assisted text was reviewed and edited by the authors to ensure accuracy and consistency with the underlying research. The authors take full responsibility for the final content of this work, including all claims, results, and conclusions.

REFERENCES

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. In *The Twelfth International Conference on Learning Representations*, 2023.

Sina Alemohammad, Ahmed Imtiaz Humayun, Shruti Agarwal, John Collomosse, and Richard Baraniuk. Self-improving diffusion models with synthetic data. *arXiv preprint arXiv:2408.16333*, 2024.

Sina Alemohammad, Zhangyang Wang, and Richard G Baraniuk. Neon: Negative extrapolation from self-training improves image generation. *arXiv preprint arXiv:2510.03597*, 2025.

Arslan Ali, Junjie Bai, Maciej Bala, Yogesh Balaji, Aaron Blakeman, Tiffany Cai, Jiaxin Cao, Tianshi Cao, Elizabeth Cha, Yu-Wei Chao, et al. World simulation with video foundation models for physical ai. *arXiv preprint arXiv:2511.00062*, 2025.

Nicholas M Boffi, Michael S Albergo, and Eric Vanden-Eijnden. Flow map matching. *arXiv preprint arXiv:2406.07507*, 2, 2024.

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

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. *arXiv preprint arXiv:2410.12557*, 2024.

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.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30467–30476, 2026.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Xiaoxiao He, Chaowei Tan, Ligong Han, Bo Liu, Leon Axel, Kang Li, and Dimitris N Metaxas. Dmcvr: Morphology-guided diffusion model for 3d cardiac volume reconstruction. In *International conference on medical image computing and computer-assisted intervention*, pp. 132–142. Springer Nature Switzerland Cham, 2023.

Xiaoxiao He, Quan Dao, Ligong Han, Song Wen, Minhao Bai, Di Liu, Han Zhang, Martin Renqiang Min, Felix Juefei-Xu, Chaowei Tan, et al. Dice: Discrete inversion enabling controllable editing for multinomial diffusion and masked generative models. *arXiv preprint arXiv:2410.08207*, 2024.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

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.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024a.

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*, pp. 24174–24184, 2024b.

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. *arXiv preprint arXiv:2310.02279*, 2023.

Kyungmin Lee, Sihyun Yu, and Jinwoo Shin. Decoupled meanflow: Turning flow models into flow maps for accelerated sampling. In *International Conference on Learning Representations*, volume 2026, pp. 64555–64583, 2026.

Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in neural information processing systems*, 37:63082–63109, 2024a.

Sangyun Lee, Yilun Xu, Tomas Geffner, Giulia Fanti, Karsten Kreis, Arash Vahdat, and Weili Nie. Truncated consistency models. *arXiv preprint arXiv:2410.14895*, 2024b.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The twelfth international conference on learning representations*, 2023.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024.

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*, pp. 23–40. Springer, 2024.

Derek Onken, Samy Wu Fung, Xingjian Li, and Lars Ruthotto. Ot-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9223–9232, 2021.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

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*, pp. 10684–10695, 2022.

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

Kim Shin Seong, Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Balanced conic rectified flow. *arXiv preprint arXiv:2510.25229*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Qipan Xu, Zhenting Wang, Xiaoxiao He, Ligong Han, and Ruixiang Tang. Can large vision-language models detect images copyright infringement from genai? *arXiv preprint arXiv:2502.16618*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *arXiv preprint arXiv:2412.06329*, 2024.

Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. Alphaflow: Understanding and improving meanflow models. In *International Conference on Learning Representations*, volume 2026, pp. 144056–144094, 2026.

Xinxi Zhang, Song Wen, Ligong Han, Felix Juefei-Xu, Akash Srivastava, Junzhou Huang, Vladimir Pavlovic, Hao Wang, Molei Tao, and Dimitris Metaxas. Soda: Spectral orthogonal decomposition adaptation for diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4665–4682. IEEE, 2025.

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. In *International Conference on Learning Representations*, volume 2026, pp. 2582–2603, 2026.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. *arXiv preprint arXiv:2503.07565*, 2025.

A APPENDIX

TABLE OF CONTENTS

A More Discussion **1**

A.1 Limitation 1

A.2 Broader Impact 2

A.3 Comparison with One-Step Sampling via Rectified Flow 2

B Implementation Details **3**

B.1 Pretrained Model Conditioning 3

B.2 Time Distribution 4

B.3 Avoiding High-Variance Time Regions 4

B.4 Training with Guidance 4

C More Experiment Details and Results **6**

C.1 Loss Landscape 6

C.2 Computation Estimation of Each Method 6

C.3 Additional Convergence Comparison with Other Related Methods 7

C.4 Curvature Estimation 8

C.5 Evaluation with Additional Function Evaluations 8

C.6 Effect of the Coupling Budget 8

C.7 Text-to-Image Experimental Details 9

C.8 Category-wise GenEval Results 11

D 2D Toy Example **11**

E More Qualitative Results **13**

E.1 ImageNet-64^2 13

E.2 ImageNet-256^2 14

E.3 ImageNet-512^2 15

A MORE DISCUSSION

In this section, we provide additional analysis and context for our method. We begin by discussing its limitations and then examine the broader impact of our approach. Next, we revisit the theoretical motivation for using a single rectification iteration and explain how our perspective complements existing analyses.

A.1 LIMITATION

Since our method does not access real data during training and relies entirely on synthetic samples generated by pretrained diffusion or flow models, its performance naturally depends on the quality of these generated couplings. A promising direction is to incorporate real data into the training process, as explored in Lee et al. (2024a); Seong et al. (2025). Another complementary line of work

investigates how to improve generative models using only their own outputs Alemohammad et al. (2023; 2024; 2025), which suggests that self-generated data can still be leveraged effectively with appropriate regularization. Alternatively, one could improve the synthetic supervision directly by using stronger backbone models to generate the couplings.

A.2 Broader Impact

Beyond empirical gains, our method highlights a practical and socially relevant shift in how large generative models can be trained. Traditional few-step or one-step distillation pipelines often concentrate compute in expensive training workloads on high-end accelerators (e.g., A100-class GPUs), which limits accessibility to well-resourced institutions. In contrast, Re-MeanFlow shifts a substantial fraction of computation to an *inference-only* stage (sampling rectified couplings), followed by a lightweight MeanFlow training phase. Because inference workloads can be executed efficiently on widely available consumer- or inference-grade accelerators, our framework reduces reliance on scarce training GPUs and lowers the barrier to experimentation.

Practically, **the sampling process for the rectified couplings is amenable to pipeline parallelism**: coupling generation can run asynchronously alongside training as a data-preparation stream. For implementation convenience, we pre-generate couplings before training in this work; nevertheless, even under this conservative setup, **Re-MeanFlow achieves the lowest total compute among prior state-of-the-art distillation methods**, underscoring the end-to-end efficiency of our framework.

A.3 Comparison with One-Step Sampling via Rectified Flow

Rectified Flow Liu et al. (2022) supports few-step sampling via an iterative *reflow* procedure that progressively straightens trajectories by repeatedly training a new flow model on rectified couplings. Liu et al. (2022) show that, in the limit, iterating this procedure can yield perfectly straight couplings. In practice, however, **achieving near-linear paths for reliable one-step sampling typically requires multiple reflow rounds**, which is computationally expensive and can degrade performance due to accumulated approximation errors.

Lee et al. (2024a) analyze when multiple Reflow iterations are truly necessary for achieving straight trajectories in rectified flows. Their argument centers on how trajectory *intersections* affect the learned velocity field. Consider two 1-rectified couplings $(\mathbf{x}', \mathbf{z}')$ and $(\mathbf{x}'', \mathbf{z}'')$. A trajectory intersection occurs if there exists $t \in [0, 1]$ such that

$$(1-t)\mathbf{x}' + t\mathbf{z}' = (1-t)\mathbf{x}'' + t\mathbf{z}''.$$

If such an intersection happens, both trajectories pass through the same intermediate point. Because rectified-flow training regresses the conditional expectation $E[\mathbf{x}|\mathbf{x}_t]$, the model must assign a *single* velocity to this shared point. As a result, the learned velocity field cannot simultaneously point toward both $\mathbf{x}'$ and $\mathbf{x}''$, and it instead averages their directions. This averaging effect bends the local velocity field, producing curvature and consequently degrading the accuracy of one-step Euler sampling, which assumes the path to be straight.

To understand how often this phenomenon can happen, Lee et al. (2024a) show that an intersection implies

$$\mathbf{z}'' = \mathbf{z}' + \frac{1-t}{t}(\mathbf{x}' - \mathbf{x}'').$$

Under typical training, nearly all noise samples used to form 1-rectified couplings lie in the high-density region of the Gaussian prior. The $\mathbf{z}''$ required above usually lies far outside that region unless $||\mathbf{x}' - \mathbf{x}''||_2$ is extremely small or t is very close to 1. Therefore, intersections are statistically rare. Further assuming that the 1-rectified flow is approximately L-Lipschitz,

$$||\mathbf{x}' - \mathbf{x}''||_2 \leq L\,||\mathbf{z}' - \mathbf{z}''||_2, \tag{14}$$

nearby noise samples cannot map to widely separated data points. Combining the rarity of intersections with this Lipschitz condition, the authors conclude that the optimal 2-rectified flow is nearly straight. Hence, in their view, one additional Reflow step is sufficient, and any remaining performance gap should be attributed primarily to training inefficiency rather than insufficient straightening.

Table 5: Training settings of Re-MeanFlow on ImageNet.

	Resolution		
	64^2	256^2	512^2
Training Details			
Model Backbone	EDM2-SKarras et al. (2024b)	SiT-XLMa et al. (2024)	EDM2-SKarras et al. (2024b)
Global Batch size	128	128	128
Learning Rate	1e-4	1e-4	1e-4
Adam β_1	0.9	0.9	0.9
Adam β_2	0.99	0.95	0.99
Model Capacity (Mparams)	280.2	676.7	280.5
EMA Rate	0.9999	0.9999	0.9999
MeanFlow Setting Details			
Ratio of $r \neq t$	0.25	0.25	0.25
p for Adaptive Weight	0.5	0.5	0.5
CFG Effective Scale w'	Uniform(1.0, 3.0)	Uniform(1.0, 3.0)	Uniform(1.0, 2.5)
Avoiding High-variance (t, r) Region	$t > 0.95$ & $r < 0.4$	$t > 0.95$ & $r < 0.4$	$t > 0.95$ & $r < 0.4$
Sampling Details for Rectified Couplings			
Pretrained Model	EDM2-SKarras et al. (2024b)	SiT-XLMa et al. (2024)	EDM2-SKarras et al. (2024b)
Sampling Number	5M	5M	5M
Guidance Method	Autoguidance Karras et al. (2024a)	CFG Ho & Salimans (2022)	Autoguidance Karras et al. (2024a)

A.3.1 Relation to Our Work.

Our focus is complementary. While the above analysis suggests that trajectory intersections are rare for most couplings, our experiments indicate that realistic settings can still contain a small but influential subset of pairs with non-negligible curvature, particularly when the effective Lipschitz constant L is large due to geometric imbalance in the data distribution. As shown in Fig. 2b and Fig. 9, trajectories induced by a once-rectified coupling (i.e., one reflow step) are substantially straighter than those from the independent coupling (Fig. 2a), yet residual curvature remains and can still hinder reliable one-step sampling based on the instantaneous velocity. Re-MeanFlow is designed to be robust in precisely these challenging cases, enabling stable one-step generation even when curvature persists after a single rectification step.

B Implementation Details

In this section, we first describe the conditioning strategy of how Re-MeanFlow utilizes previous pretrained models. We then outline the key design choices during training Re-MeanFlow. We provide the training hyperparameters across all ImageNet resolutions in Tab. 5.

B.1 Pretrained Model Conditioning

In Re-MeanFlow, the velocity network is conditioned on two time variables, t and r. We implement this conditioning by learning two separate embeddings, $\mathrm{emb}_t(t)$ and $\mathrm{emb}_r(r)$, and summing them before passing the result to the rest of the network. When initializing from pretrained models, the original networks only contain a single time embedding for t. Replacing this embedding with our two-embedding design requires careful initialization to ensure the model initially behaves like the pretrained flow mode. Specifically, before MeanFlow fine-tuning, we want:

$$u(\mathbf{z}_t, r, t) \approx v(\mathbf{z}_t, t) \tag{15}$$

On ImageNet 64^2 and 512^2, which Re-MeanFlow is initialized from the pretrained EDM2-S model Karras et al. (2024b), following AYF Sabour et al. (2025), we perform a short alignment stage in which we train the new embeddings to reproduce the original time-embedding output for the corresponding noise level. Specifically, for EDM2-S the original embedding depends on $\log \sigma_t$, where $\sigma_t = \frac{t}{1-t}$. We train the new embeddings via:

$$\mathbb{E}_{t,r}[||\mathrm{emb}_t(t) + \mathrm{emb}_r(r) - \mathrm{emb}_{\mathrm{ori}}(\log \sigma_t)||_2^2], \tag{16}$$

for 10k iterations with learning rate 1e-3. This process takes only a few minutes. We also convert the original VE diffusion parameterization into the flow-matching setting following Lee et al. (2024a).

On ImageNet 256^2, we initialize Re-MeanFlow from SiT-XL Ma et al. (2024), which is already a flow-based model. In this case, only the additional r-embedding needs to be introduced. We simply zero-initialize $\mathrm{emb}_r(r)$ and keep the original SiT time embedding for $\mathrm{emb}_t(t)$.

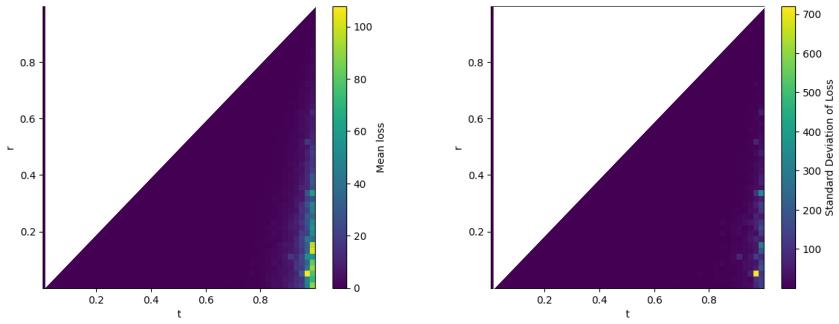


Figure 6: **Heatmaps of the mean loss (left) and the standard deviation of the loss (right)** for a trained Re-MeanFlow model on ImageNet-512^2 under configuration (d).

B.2 Time Distribution

We observe a similar loss–time profile to that reported in Lee et al. (2024a): the training loss as a function of t closely matches that of 2-rectified flow++.

Following AYF Sabour et al. (2025), after sampling t we draw the interval length $|t - r|$ from a normal distribution $\mathcal{N}(P_{\text{mean}}, P_{\text{std}})$ and apply a sigmoid transformation. We use the same parameters as AYF, $(P_{\text{mean}}, P_{\text{std}}) = (-0.8, 1.0)$, which emphasize medium-length intervals and substantially improve stability.

We also experimented with sampling t uniformly. Despite its simplicity, uniform sampling performs competitively—and in many cases better—than commonly used log-normal time distributions in diffusion and flow-matching models Karras et al. (2022; 2024b); Geng et al. (2025), which prioritize mid-range timesteps to avoid high variance near $t \approx 0$ and $t \approx 1$. We attribute the strong performance of uniform sampling to the significantly lower variance of rectified trajectories: after one reflow step, early-time and high-noise regions become much more stable, allowing us to allocate more samples to these challenging regimes without the usual degradation observed when training on independent couplings.

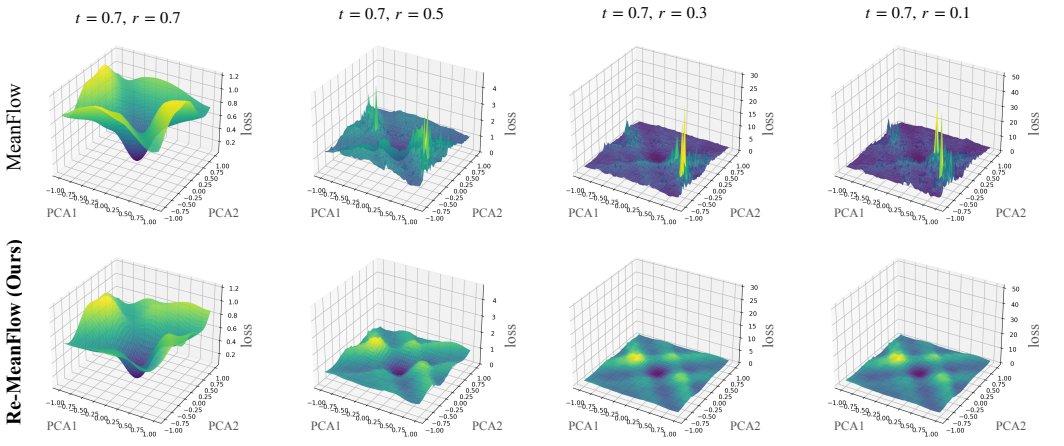
B.3 Avoiding High-Variance Time Regions

As discussed in Sec. 3.3, we observe that Re-MeanFlow exhibits unusually high variance when the noise level t is large while the reference time r is close to zero. Intuitively, this corresponds to asking the model to predict a *slightly denoised* sample ($r \in [0, 0.4]$) from an input that remains heavily corrupted ($t \approx 1$). To quantify this effect, we sample 100k pairs of (t, r) uniformly and evaluate a trained Re-MeanFlow model under configuration (d) in Table 4. As shown in Fig. 6, the resulting loss landscape displays a clear spike in both error and variance within this region, often even higher than the loss incurred when predicting directly from noise to a clean target.

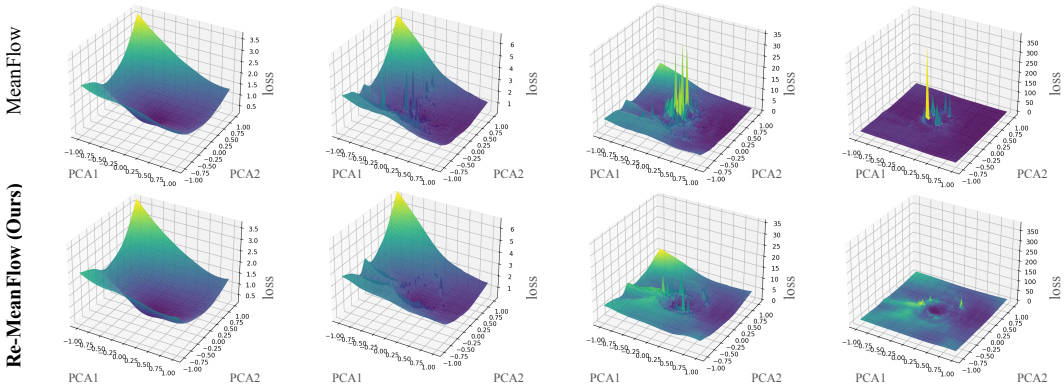
Inspired by the truncation strategy in TCM Lee et al. (2024b), we adopt a simple yet effective rule to avoid this problematic regime: whenever a sampled pair satisfies $t > 0.95$ and $r < 0.4$, we set $r = 0$. Empirically, this improves both training stability and FID. Our hypothesis is that predicting a *clean* image from pure noise ($r = 0$, $t \approx 1$) is substantially easier than predicting a lightly corrupted target: the latter requires the model to determine which noise components should be preserved, introducing ambiguity and variance at high t. By redirecting training toward these easier high-t targets, the model can allocate more capacity to learning accurate one-step predictions. This modification not only improves FID relative to configuration (c), but also accelerates convergence: in the high-t regime, more updates involve $r = 0$, allowing the model to refine its one-step outputs more quickly and reliably.

B.4 Training with Guidance

Classifier-free guidance (CFG) Ho & Salimans (2022) is widely used to boost the performance of diffusion and flow-based generative models. To incorporate CFG into the MeanFlow stage, we train



(a) Loss Landscape on ImageNet 64^2



(b) Loss Landscape on ImageNet 512^2

Figure 7: **Loss landscape of $\mathcal{L}_{MF}$ on a PCA plane of $\mathbf{z}_t$** for ImageNet 64^2 **(a)** and 512^2 **(b)**. We evaluate $\mathcal{L}_{MF}$ on a 2D PCA grid with $t = 0.7$ and $r \in \{0.7, 0.5, 0.3, 0.1\}$ (left to right). **MeanFlow (top)** becomes increasingly sharp and irregular as $t - r$ grows, whereas **Re-MeanFlow (bottom) remains substantially smoother and better-conditioned, supporting more stable and efficient training.**

Re-MeanFlow on the CFG-enhanced velocity field:

$$v^{\text{cfg}}(\mathbf{z}_t, t \mid c) \triangleq \omega\, v(\mathbf{z}_t, t \mid c) + (1 - \omega)\, v(\mathbf{z}_t, t). \tag{17}$$

MeanFlow Geng et al. (2025) further introduced an improved CFG method that mixes conditional and unconditional mean-velocity predictions:

$$v^{\text{cfg}}(\mathbf{z}_t, t \mid c) = \omega\, v(\mathbf{z}_t, t \mid c) + \kappa u^{\text{cfg}}(\mathbf{z}_t, t, t|c) + (1 - \omega + \kappa)\, u^{\text{cfg}}(\mathbf{z}_t, t, t). \tag{18}$$

which is equivalent to using an effective guidance of $\omega' = \frac{\omega}{1-\kappa}$. We adopt this improved CFG formulation for all experiments.

Empirically, we found that directly training MeanFlow on the CFG field is unstable, consistent with observations in Hu et al. (2025). To mitigate this, we use a simple two-stage strategy: first train u_θ on the unconditional flow, then on the CFG-modified flow. Usually, allocating half of the total training budget to each stage provides a good balance between stability and final quality.

We also found that sampling a random CFG scale ω' from a uniform distribution (rather than fixing it) gives better results. Large values of κ are also important for stable training. In practice, we sample ω' from a uniform distribution, then set $\kappa = \max(1.0, \omega' - 1)$, and finally compute the corresponding value for $\omega = \frac{\omega'}{1-\kappa}$.

C More Experiment Details and Results

In this section, we provide additional experimental details, computational analyses, and extended results. We first present loss-landscape visualizations at 64^2 and 512^2, then describe our protocol for estimating FLOPs and GPU-hours across all methods. Finally, we report additional convergence results that further support the synergy between trajectory rectification and mean-velocity modeling.

C.1 Loss Landscape

In Fig. 3, we visualize the loss landscape of $\mathcal{L}_{MF}$ with respect to the input $\mathbf{z}_t$ for MeanFlow and Re-MeanFlow across different (t, r) pairs. Here we report the corresponding results at additional resolutions in Fig. 7, following the same protocol.

Unlike conventional loss-landscape analyses that perturb model parameters, **we vary the input state $\mathbf{z}_t$ to probe the intrinsic conditioning of the mean-velocity regression problem.** This choice follows directly from our curvature-bottleneck view: the difficulty of learning mean velocity is primarily governed by trajectory geometry. When trajectories are highly curved, the interval mean velocity becomes a rapidly varying, locally unstable function of $\mathbf{z}_t$, making the induced loss highly sensitive to small input perturbations and yielding a rugged landscape in $\mathbf{z}_t$-space. Such ruggedness indicates that the regression target itself is poorly conditioned, which in turn hampers optimization. In contrast, rectified (straighter) trajectories markedly smooth the landscape as a function of $\mathbf{z}_t$, producing a better-conditioned learning problem and enabling more efficient training.

Concretely, we fix $t = 0.7$ and sweep r, and compute PCA directions by collecting 20 k samples of $\mathbf{z}_t$ (equivalently, $\mathbf{x}_t$ in our implementation). For each resolution, we then evaluate $\mathcal{L}_{MF}$ on a 2D grid spanned by the top two PCA directions using a minibatch of samples. Collecting the data and generating all surfaces for a single resolution takes approximately 3 hours on one A100 GPU. Overall, consistent with Fig. 3, **Re-MeanFlow exhibits a substantially smoother and better-conditioned loss landscape, which translates into more efficient and stable training.**

C.2 Computation Estimation of Each Method

C.2.1 FLOPS Estimation.

In Fig. 4 (right), we report efficiency in terms of estimated exaFLOPs (EFLOPs). To ensure comparability, we estimate total training and sampling compute for each method based on their reported FLOPs per forward pass. Specifically, we use the following assumptions:

- The FLOPs of a forward pass are reported by prior works (e.g., EDMKarras et al. (2022): 100 GFLOPs, EDM2-SKarras et al. (2024b): 102 GFLOPs, SiT-XLMa et al. (2024): 118.64 GFLOPs).
- The FLOPs of a backward pass are measured empirically and are approximately $2\times$ the cost of a forward pass. (One JVP operation is also counted as a backward pass), say for our example on the first stage where we will perform one forward of the model and one JVP operation and one back propagation with JVP counted as one back propagation we have total flop amount of $1 + (2 \times 2) = 1 + 4$ forward flops.
- For training, the total compute is computed as:

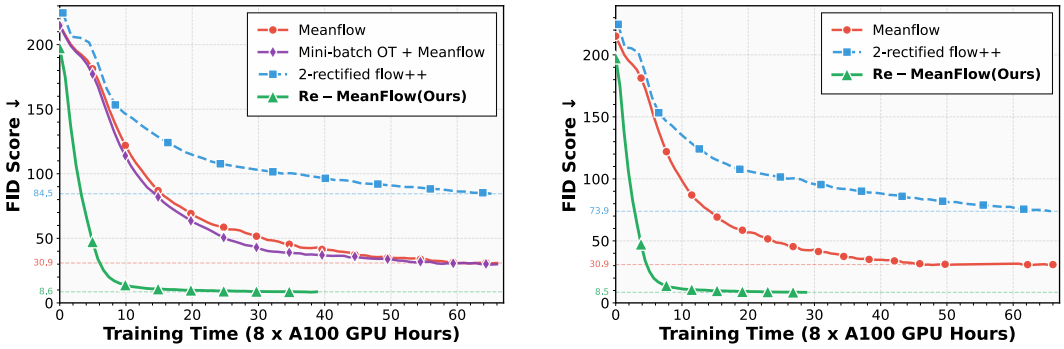
$$\begin{aligned}\text{Total Train FLOPs} =&(\#\text{iters}) \times (\text{batch size}) \times (\text{forward} + \text{backward}) \\ &\times (\text{GFLOPs per fwd}).\end{aligned}$$

- For sampling in the reflow process, the total compute is computed as:

$$\begin{aligned}\text{Total Sample FLOPs} =&(\#\text{samples}) \times (\#\text{steps}) \times (\text{forward passes per step}) \\ &\times (\text{GFLOPs per fwd}).\end{aligned}$$

- *Example: Re-MeanFlow (Ours) on ImageNet-64^2.* For sampling, we require:

$$\underbrace{5 \times 10^6}_{\text{\#samples}} \times \underbrace{63}_{\text{steps}} \times \underbrace{2}_{\text{fwd/step (auto-guidance)}} \times \underbrace{102}_{\text{GFLOPs/fwd}} \approx 64 \text{ Eflops}.$$



(a) ImageNet 512 **(b)** ImageNet 256

Figure 8: **Additional convergence comparison.** FID vs. training time (8×A100 GPU hours) for **(a)** ImageNet 512^2 and **(b)** ImageNet 256^2. We compare Re-MeanFlow against MeanFlow Geng et al. (2025) and closely related baselines, including 2-rectified flow++ Lee et al. (2024a) and MeanFlow trained with mini-batch OT couplings Tong et al. (2023) (512^2 only). Across resolutions and baselines, **Re-MeanFlow converges markedly faster and reaches substantially lower FID.**

For training, we have two stages, with the first stage trained on the original flow and the second stage trained on the CFG velocity field:

$$\underbrace{50{,}000}_{\text{iters}} \times \underbrace{128}_{\text{batch}} \times \underbrace{(1+4)}_{\text{fwd+back}} \times \underbrace{102}_{\text{GFLOPs/fwd}} + \underbrace{50{,}000}_{\text{iters}} \times \underbrace{128}_{\text{batch}} \times \underbrace{(3+4)}_{\text{fwd+back}} \times \underbrace{102}_{\text{GFLOPs/fwd}} \approx 8 \text{ Eflops}.$$

- *Example: AYF Sabour et al. (2025).* AYF does not require sampling, so only the training computation is considered:

$$\underbrace{50{,}000}_{\text{iters}} \times \underbrace{2048}_{\text{batch}} \times \underbrace{(4+4)}_{\text{fwd+back}} \times \underbrace{102}_{\text{GFLOPs/fwd}} \approx 8.36 \times 10^{10} \text{ GFLOPs} \approx 84 \text{ Eflops}.$$

C.3 Additional Convergence Comparison with Other Related Methods

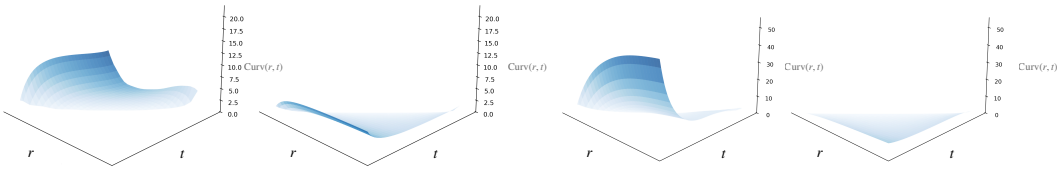
In Fig. 1c, we compare the convergence of Re-MeanFlow with MeanFlow Geng et al. (2025). Here, we provide additional convergence results against closely related baselines.

C.3.1 2-rectified flow++.

We compare against 2-rectified flow++ Lee et al. (2024a), which improves Rectified Flow but still models the *instantaneous* velocity on rectified couplings, in contrast to Re-MeanFlow, which models *mean*-velocity. We observe that one-step sample quality under 2-rectified flow++ improves slowly. We attribute this to the sensitivity of one-step Euler updates to residual curvature: accurate one-step sampling with instantaneous velocity effectively requires near-perfect straightness, since even mild curvature can cause overshooting. Because rectified trajectories are not perfectly linear in practice, this sensitivity compounds and slows convergence. This observation aligns with our efficiency comparison: relative to 2-rectified flow++, **Re-MeanFlow achieves a** 33.4% **lower FID.**

C.3.2 MeanFlow with Mini-batch OT couplings.

We also train MeanFlow using Mini-batch OTTong et al. (2023) couplings, obtained by locally solving an OT matching within each mini-batch (via an OT server) to pair sampled noises and data points. Results on ImageNet 512^2 are shown in Fig. 8a. While mini-batch OT slightly improves over standard MeanFlow, its convergence behavior remains similar and still lags substantially behind Re-MeanFlow. This is consistent with our discussion in Sec. 2.1: Mini-batch OT provides only local transport structure and does not guarantee globally straight trajectories.



(a) Curvature Estimation for the Generative Trajectories on ImageNet 64^2

(b) Curvature Estimation for the Generative Trajectories on ImageNet 256^2

Figure 9: **Curvature estimation of generative trajectories** on ImageNet 64^2 **(a)** and 256^2 **(b)**. Using the proxy in Sec. 3.2.1, we visualize $\mathrm{Curv}(r,t)$ over the (r,t)-plane. **Rectified couplings used by Re-MeanFlow induce substantially straighter trajectories (lower curvature) than the independent couplings used by MeanFlow.**

Overall, Re-MeanFlow converges substantially faster than these closely related alternatives, highlighting the synergy of modeling mean-velocity on rectified trajectories.

C.4 Curvature Estimation

In Fig. 2, we visualize the curvature proxy on ImageNet 512^2. Here we provide the corresponding curvature estimates at additional resolutions in Fig. 9. The same trend holds across settings: compared to independent couplings, **the rectified couplings used to train Re-MeanFlow consistently induce substantially lower curvature over the (r,t)-plane.**

C.5 Evaluation with Additional Function Evaluations

While our main experiments focus on one-step generation, we additionally evaluate the same trained Re-MeanFlow models using 2 and 4 NFEs, without further training. Table 6 reports the results on ImageNet at 64^2, 256^2, and 512^2 resolutions. Across all three resolutions, additional function evaluations consistently improve FID. Most of the improvement is obtained by increasing the sampling budget from 1 to 2 NFEs, with smaller gains from 2 to 4 NFEs. These results show that **the models used for one-step generation also benefit from a modest increase in inference-time computation.**

Table 6: **Evaluation at different NFEs.** FID (↓) of the same Re-MeanFlow models on ImageNet, evaluated with 1, 2, and 4 NFEs.

Resolution	1 NFE	2 NFEs	4 NFEs
64^2	2.87	2.51	**2.48**
256^2	3.41	3.03	**2.98**
512^2	3.03	2.67	**2.60**

C.6 Effect of the Coupling Budget

Our main experiments use a default budget of 5M rectified couplings. To examine the effect of this choice, we additionally train Re-MeanFlow with 3M and 7M couplings on ImageNet 512^2. As shown in Table 7, increasing the coupling budget from 3M to 5M reduces one-step FID from 4.25 to 3.03, while increasing it further to 7M yields an FID of 2.85. Thus, **larger coupling budgets improve FID, with diminishing gains over the evaluated range.** The default 5M budget captures most of the improvement observed in this sweep, although additional couplings can further improve generation quality.

Table 7: **Coupling-budget ablation.** One-step FID of Re-MeanFlow on ImageNet 512^2 with different numbers of rectified couplings.

# Couplings	FID ↓
3M	4.25
5M (default)	3.03
7M	**2.85**

Table 8: **Category-wise GenEval results on Cosmos-Predict2-2B.** Higher scores are better in all columns. The multi-step teacher is included as a reference; **bold** denotes the best result among the controlled student methods. The teacher, MeanFlow, and Re-MeanFlow use a guidance scale of 4.5.

Method	NFE	Overall	Single obj.	Two obj.	Counting	Colors	Position	Attr. binding
Multi-step teacher reference								
Cosmos-Predict2	2×35	0.8019	0.9938	0.9722	0.7594	0.8511	0.5700	0.6650
Controlled post-training comparison								
Rectified Flow	4	0.5060	0.8281	0.5732	0.4188	0.5984	0.2850	0.3325
MeanFlow	4	0.6331	0.9688	0.8005	0.5719	0.6649	0.3400	0.4525
Re-MeanFlow (ours)	4	**0.7385**	**0.9844**	**0.8965**	**0.7188**	**0.7739**	**0.4625**	**0.5950**

Table 9: **Text-to-image training settings.** Controlled post-training comparison on Cosmos-Predict2-2B. All methods share the pretrained initialisation, synthetic image–prompt corpus, hardware, and training-iteration budget. Teacher NFEs count both classifier-free guidance branches.

Setting	**MeanFlow**	**Rectified Flow**	**Re-MeanFlow**
Backbone and synthetic corpus			
Backbone		Cosmos-Predict2-2B	
Number of prompts		50K, selected from LAION-500K	
Generated images per prompt	2	2	2
Number of training images	100K	100K	100K
Teacher sampling steps	35	35	35
Teacher CFG scale	4.5	4.5	4.5
Teacher NFEs per image	70	70	70
Corpus-generation cost		Approximately one day on $8\times$ A100	
Optimisation			
Training hardware		$8\times$ NVIDIA A100	
Global batch size	24	24	24
Batch size per GPU	3	3	3
Learning rate	5×10^{-7}	5×10^{-7}	5×10^{-7}
Training iterations	10K	10K	10K
EMA	None	None	None
$\sigma_{\max}$	80	80	80
Loss-normalisation exponent p_{loss}	0	0	0
Regression loss	Squared ℓ_2	Squared ℓ_2	Squared ℓ_2
Coupling and time sampling			
Noise–image coupling	Independent	Rectified	Rectified
Current-time distribution	Logit-normal $(-0.8, 1.6^2)$	U-shaped (ImageNet setting)	U-shaped (ImageNet setting)

C.7 Text-to-Image Experimental Details

Experimental framework. As discussed in the main text, our text-to-image experiments provide a controlled proof of concept for extending rectified mean-velocity learning beyond ImageNet. We use the pretrained Cosmos-Predict2-2B model (Ali et al., 2025) and build on the rCM implementation (Zheng et al., 2026), including its FlashAttention-2-compatible Jacobian–vector product computation. Within this shared implementation, we implement Rectified Flow, MeanFlow (Geng et al., 2025), and Re-MeanFlow using their respective training objectives. All three methods are initialised from the same pretrained checkpoint. We adopt rCM's implementation and optimisation configuration as a starting point, with the modifications described below to accommodate our computational budget and the respective objectives. We do

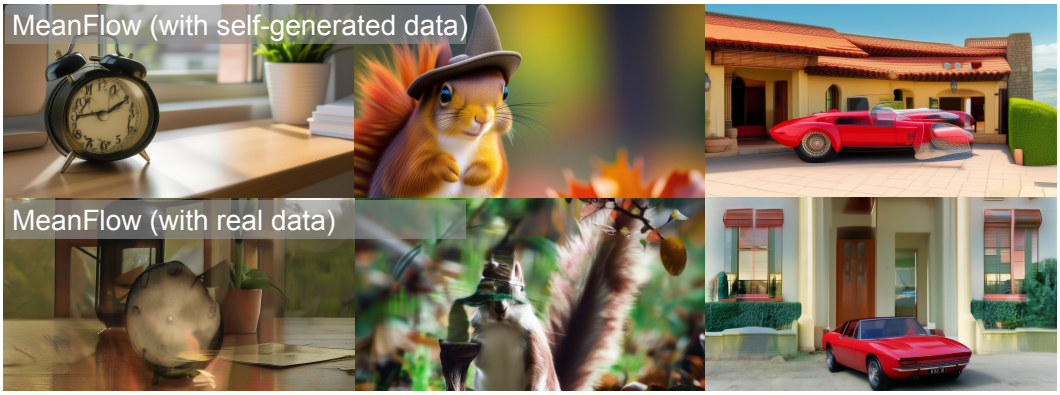


Figure 10: **Instability of direct MeanFlow fine-tuning on real-image–prompt pairs.** In our preliminary experiment using the selected LAION pairs, GenEval decreases from approximately 0.4 to 0.2 during fine-tuning. This observation motivates constructing a teacher-generated corpus for the controlled comparison.

not use score-distillation or adversarial losses; consequently, no auxiliary fake-score network or discriminator is trained.

Prompt selection and preliminary real-image experiments. Since we do not have access to the original training corpus of Cosmos-Predict2, we select a curated subset of 50K prompts from the LAION-500K prompt pool used in DMD (Yin et al., 2024). We first attempted direct MeanFlow fine-tuning using the corresponding real-image–prompt pairs. Under this preliminary configuration, training was highly unstable: GenEval deteriorated from approximately 0.4 to 0.2 as training progressed, as shown in Fig. 10. This observation motivates the use of teacher-generated images for our controlled post-training comparison.

Construction of the synthetic training corpus. For each selected prompt, we generate two images using the pretrained Cosmos-Predict2 teacher, yielding 100K synthetic training examples. We use the default teacher sampler with 35 sampling steps and a classifier-free guidance scale of 4.5. Counting the conditional and unconditional evaluations required by classifier-free guidance, this corresponds to 70 NFEs per generated image. For each example, we retain the prompt, generated image, and associated source noise. The resulting corpus can be written as

$$\mathcal{D}_{\text{syn}} = \{(c_i, z_{i,j}, x_{i,j}) \mid x_{i,j} = G_\phi(z_{i,j}, c_i),\ i = 1, \ldots, 50{,}000,\ j \in \{1, 2\}\}\,, \tag{19}$$

where G_ϕ denotes the pretrained teacher sampler. Rectified Flow and Re-MeanFlow retain the correspondence between each generated image and its source noise to construct rectified couplings. The MeanFlow baseline uses the same generated images with independently sampled noise. Thus, the methods share the same image–prompt corpus while differing in their coupling construction and training objectives.

Generating this corpus takes approximately one day on a single node with eight NVIDIA A100 GPUs. The corpus is generated once and reused across all three methods. In end-to-end post-training cost accounting, we charge the same corpus-generation cost to each method in addition to its own training cost. The controlled comparison therefore does not assign a synthetic-data generation overhead exclusively to Re-MeanFlow.

Optimisation settings. We train each method for 10K iterations on eight A100 GPUs, using a global batch size of 24, corresponding to three examples per GPU. To accommodate this substantially smaller batch size, we reduce the learning rate from the rCM reference setting of 10^{-6} to 5×10^{-7} for all methods. For this short post-training experiment, we do not use exponential moving averages and evaluate the non-EMA student parameters. We retain $\sigma_{\max} = 80$. In addition, we set the adaptive loss-normalisation exponent $p_{\text{loss}} = 0$ across all methods, corresponding to a squared ℓ_2 loss on their respective regression residuals. The principal shared and method-specific settings are summarised in Table 9.

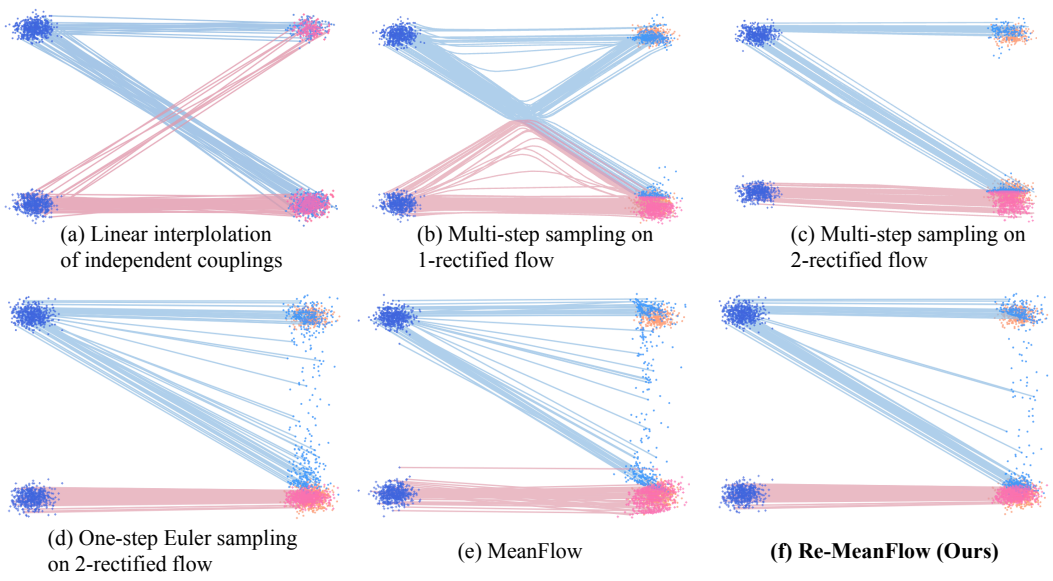


Figure 11: **A 2D Toy Example.** We consider a controlled 2D setup where a flow model transports a two-component Gaussian mixture on the left to a mixture on the right. **Panels (a-c):** *(a)* Linear interpolation of independently sampled couplings $p_{\mathbf{x}} \times p_{\mathbf{z}}$, which serves as the training signal for the first velocity model. *(b)* The resulting 1-rectified flow learned from these independent couplings; the learned velocity field remains noticeably curved. *(c)* Using the velocity field from (b), we generate a new set of couplings and train a second velocity model on their linear interpolations, yielding the 2-rectified flow. **Panel (d):** Due to imperfect straightening, one-step Euler sampling on the 2-rectified flow still yields noticeable outliers. **Panel (e):** MeanFlow trained directly on independent couplings fails to converge within the training budget because high-variance conditional velocities destabilize learning. **Panel (f): Re-MeanFlow combines trajectory rectification with mean-velocity modeling, eliminating most outliers and achieving more accurate one-step generation.**

Time sampling and stabilisation. For Rectified Flow and Re-MeanFlow, we use the same U-shaped current-time sampling distribution as in our ImageNet experiments. For the MeanFlow baseline, we use a logit-normal current-time distribution with parameters $(-0.8, 1.6^2)$. For Re-MeanFlow, we also retain the ImageNet time-pair stabilisation rule: when a sampled pair satisfies $t > 0.95$ and $0 < r < 0.4$, we set $r = 0$. This redirects the corresponding high-noise, long-interval training example to the clean endpoint rather than discarding it.

C.8 Category-wise GenEval Results

Table 8 provides the full category-wise breakdown of the GenEval results reported in the main text. All three student methods are evaluated at 4 NFEs, while the pretrained Cosmos-Predict2 teacher serves as a 70-NFE reference. **Re-MeanFlow outperforms both controlled baselines across all six categories**, improving the overall score from 0.6331 for MeanFlow to 0.7385. The largest absolute improvements over MeanFlow occur in counting (+0.1469), attribute binding (+0.1425), and position (+0.1225), showing that the aggregate improvement extends across multiple compositional categories. Nevertheless, the multi-step teacher retains higher scores in every category, with the largest remaining gap in position (0.4625 for Re-MeanFlow versus 0.5700 for the teacher).

D 2D Toy Example

To visualize the synergy of Re-MeanFlow relative to MeanFlowGeng et al. (2025) and Rectified Flow Liu et al. (2022), we present a 2D toy transport problem in Fig. 11. The task maps a two-mode Gaussian mixture to another two-mode mixture. We compare one-step generation under a fixed budget of 20k training iterations for three approaches: (i) 2-rectified flow, (ii) MeanFlow, and

(iii) Re-MeanFlow (ours), which trains a velocity model for 10k iterations to obtain a 1-rectified coupling, followed by 10k iterations of MeanFlow on the resulting couplings.

Fig. 11d: 2-rectified flow. Even after one rectification step, some samples traverse *disproportionately long distances* toward the denser mode (e.g., due to approximation error or mild mode imbalance). As a result, nearby noise samples can map to widely separated data points, yielding a large effective L in Eq. 14 and inducing early-time curvature near $t \approx 0$. The pronounced one-step deviations in Fig. 11d illustrate this failure mode: one-step Euler updates using instantaneous velocity can overshoot when trajectories are not perfectly straight. Similar geometric imbalances can arise in real datasets, motivating robustness to residual curvature.

Fig. 11e: MeanFlow. MeanFlow trained directly on independent couplings struggles to learn a coherent transport map within the same budget, resulting in poor one-step generation.

Fig. 11f: Re-MeanFlow (ours). Re-MeanFlow achieves more accurate one-step generation with substantially fewer invalid samples. This illustrates the complementary roles of the two components: rectification reduces curvature enough to make MeanFlow optimization efficient, while mean-velocity modeling removes the requirement of perfectly straight trajectories.

E MORE QUALITATIVE RESULTS

In this section, we present additional selected qualitative samples generated by Re-MeanFlow across all ImageNet resolutions.

E.1 IMAGENET-64^2

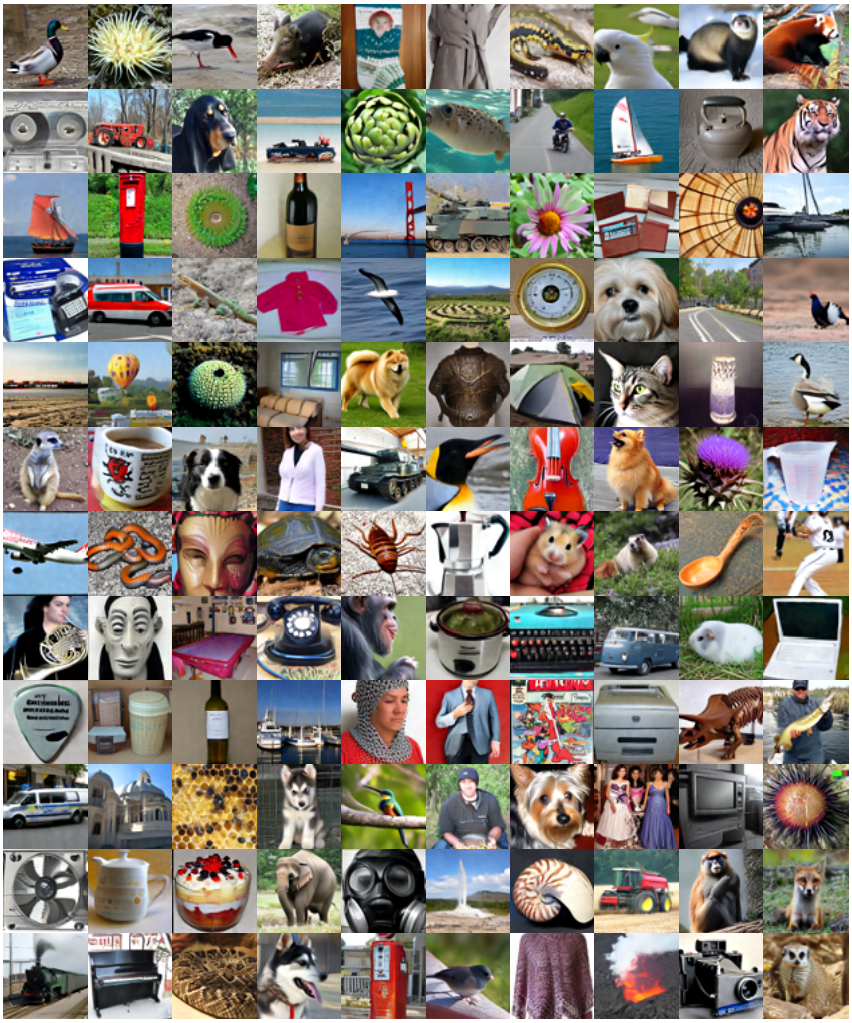


Figure 12: Selected qualitative results for Re-MeanFlow (NFE=1) on ImageNet 64^2.

E.2 IMAGENET-256^2



Figure 13: Selected qualitative results for Re-MeanFlow (NFE=1) on ImageNet 256^2.

E.3 IMAGENET-512^2



Figure 14: Selected qualitative results for Re-MeanFlow (NFE=1) on ImageNet 512^2.