RETHINKING PIXEL MEAN FLOWS VIA INTERVAL DENOISER

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern diffusion and flow-based models are increasingly moving toward few-step, latent-free generation to bypass the computational overhead of multi-step sampling and the reconstruction bottlenecks of external autoencoders. We propose the *Interval Denoiser*, a principled framework for latent-free generation. Derived directly from the flow matching ODE, it establishes an exact analytical mapping for intermediate trajectory states and identifies the image-space prediction target as an interval-weighted average of instantaneous denoisers, which supports the generalized manifold hypothesis under which regression is tractable for a network operating directly on pixels. The derivation recovers the substitution of Pixel MeanFlow (pMF) and the decoder of CTM as special cases. Furthermore, by placing the stop-gradient only on the time derivative, our update is the exact first-order gradient of the Interval Denoiser identity, whereas pMF's stop-gradient placement damps the bootstrapped derivative term by a factor r/t. By analyzing this objective, we show that the exact weighting is ill-conditioned for wide intervals and equip our framework with residual clipping and a time-sampling curriculum, enabling effective long-interval training and improving few-step performance. Trained from scratch on ImageNet 256×256, our model achieves an FID of 4.55 in one step (1-NFE) and 3.98 in two steps (2-NFE) without perceptual losses.

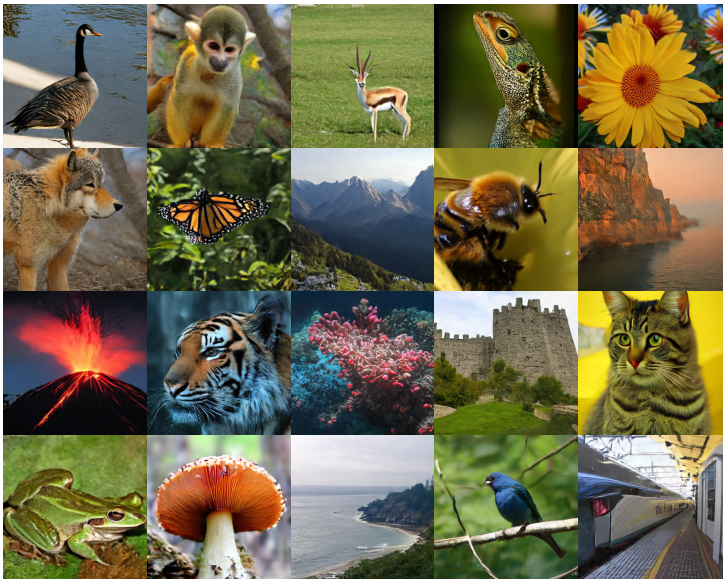

Figure 1: Selected 1-NFE pID-L/16 samples on ImageNet 256×256, generated in pixel space without autoencoders or perceptual losses (settings in Appendix G, uncurated samples in Figs. A3–A4).

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.

1 INTRODUCTION

Diffusion models and their flow-based variants (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023) simulate continuous-time ordinary differential equations (ODEs), but require multi-step numerical integration and rely on compressed latent spaces (Rombach et al., 2022), which limits pixel-level fidelity. Consistency Models (Song et al., 2023), Consistency Trajectory Models (CTM) (Kim et al., 2024), and MeanFlow (Geng et al., 2025; 2026) have drastically reduced sampling steps, while other works have demonstrated the feasibility of operating directly in raw pixel space (Li & He, 2025; Lei et al., 2026).

The foundations of Pixel MeanFlow (pMF) (Lu et al., 2026), the strongest method in this regime, remain partly empirical. pMF combines the Improved MeanFlow (Geng et al., 2026) objective with the algebraic substitution $u = (z_t - X)/t$, and characterizes the resulting target $X(z_t, r, t)$ only at the boundaries $r = t$ and $r = 0$, leaving the general case $0 < r < t$ as an empirical observation. Furthermore, written in terms of the image network, pMF's stop-gradient encloses not only the time derivative but also the network's own zeroth-order outputs, so its update is not the first-order gradient of the identity that X satisfies.

Contributions. We give a principled derivation of few-step latent-free generation. We analyze the flow matching ODE and derive mapping for intermediate trajectory states termed the *Interval Denoiser*, and show that it is an interval-weighted average of instantaneous denoisers, a convex combination of posterior means along the trajectory, which equals the denoiser at $r = t$ and the ODE endpoint at $r = 0$. This identifies the image-space target of pixel-space few-step models and supports the *generalized manifold hypothesis* of denoised images (Lu et al., 2026) that pMF states empirically. It derives pMF's substitution and recovers CTM's decoder parameterization. We then compare training objectives: our update is the exact first-order gradient of the Interval Denoiser identity, whereas pMF's stop-gradient placement damps the bootstrapped derivative term by r/t (Prop. 4), so the two coincide only once the identity holds. The same analysis shows that the exact weight $(t-r)/r$ is unbounded for wide intervals, which is why residual clipping (Lu & Song, 2025; Peng et al., 2026) is needed there. Empirically, a two-phase time-sampling curriculum (Sun, 2026) that shifts training from short to wide intervals is the main driver of few-step quality (9.25 to 7.55 FID for pID-B/16), while with clipping the pID and pMF objectives perform on par under a static sampler, consistent with their shared fixed point. Trained from scratch in pixel space on ImageNet 256×256 (Deng et al., 2009), pID-L/16 reaches an FID of 4.55 at 1-NFE and 3.98 at 2-NFE (Fig. 1) without perceptual losses, which can artificially lower FID (Kynkäänniemi et al., 2023; Song & Dhariwal, 2024).

2 RELATED WORK

Direct Pixel-Space Generation. Diffusion (Ho et al., 2020; Song et al., 2021) and flow matching (Lipman et al., 2023; Albergo et al., 2025) typically operate in the latent spaces of pretrained autoencoders (Rombach et al., 2022), which reduce cost but introduce reconstruction bottlenecks. Operating directly in pixel space exposes the network to high-dimensional inputs, and Vision Transformers (Dosovitskiy et al., 2021) degrade rapidly when the dimensionality per token becomes too large (Chen et al., 2025; Yao et al., 2025; Shi et al., 2025). While images reside on a low-dimensional manifold (Chapelle et al., 2006; Vincent et al., 2008), predicting unstructured noise or velocity fields in pixel space is difficult (Li & He, 2025). Therefore, recent methods let the network output a denoised image (x-prediction) that is algebraically transformed to velocity objectives (Karras et al., 2022; Lu et al., 2026).

Few-Step Generative Models. Consistency Models (Song et al., 2023) and CTM (Kim et al., 2024) learn mappings that enable large discrete transitions along the generation trajectory, and the MeanFlow family (Geng et al., 2025; 2026) predicts the average velocity over a time interval. Our Interval Denoiser identity (Eq. 9) is written for the x-space quantity that pixel models predict, with the specific weight $t(t-r)/r$ that connects it to pMF and CTM. pMF (Lu et al., 2026) adapted Improved MeanFlow (iMF) (Geng et al., 2026) to latent-free generation with an x-space prediction, motivated by a generalized manifold hypothesis that is established only at the interval boundaries, and with a stop-gradient placement inherited from the velocity parameterization. It further relies

on perceptual losses (Zhang et al., 2018), which can lower FID without a gain in sample quality (Kynkäänniemi et al., 2023). Kim et al. (2026) analyze MeanFlow dynamics and advocate shifting supervision from short to long intervals, and FACM (Peng et al., 2026) and sCM (Lu & Song, 2025) stabilize consistency training.

3 BACKGROUND

Flow Matching. Flow matching (Lipman et al., 2023) transports a Gaussian prior to the data distribution. For clean data $x_0 \sim p_{\text{data}}$ at $t = 0$ and noise $\epsilon \sim \mathcal{N}(0, I)$ at $t = 1$, the linear path and conditional velocity are $z_t = (1 - t)x_0 + t\epsilon$ and $v_t = \epsilon - x_0$. A network $v_\theta(z_t, t)$ is trained by regression against v_t, $\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,x_0,\epsilon}[\|v_\theta(z_t, t) - v_t\|_2^2]$, whose minimizer is the marginal velocity $v(z_t, t) = \mathbb{E}[v_t \mid z_t]$, and samples are generated by solving $dz_t/dt = v_\theta(z_t, t)$ from $t = 1$ to $t = 0$.

The MeanFlow Family. To enable large generation steps, MeanFlow (Geng et al., 2025) predicts the average velocity over an interval $[r, t]$, $u(z_t, r, t) = \frac{1}{t-r}\int_r^t v(z_\tau, \tau)d\tau$, and samples with $z_r = z_t - (t - r)u_\theta(z_t, r, t)$. Unlike MeanFlow, whose regression target depends on the network, Improved MeanFlow (Geng et al., 2026) regresses onto the network-independent target v_t, computing the time derivative via a stop-gradient Jacobian-Vector Product (JVP_{sg}) along the network's own instantaneous velocity v_θ. Omitting inputs for brevity, the objective is:

$$\mathcal{L}_{\text{iMF}} = \mathbb{E}_{t,r,x_0,\epsilon}\big[\|u_\theta + (t - r)\text{JVP}_{\text{sg}}(u_\theta; v_\theta) - v_t\|_2^2\big]. \tag{1}$$

For pixel-space generation, pMF (Lu et al., 2026) introduces an image-space network $X_\theta(z_t, r, t)$ via the algebraic substitution:

$$u_\theta(z_t, r, t) = \frac{z_t - X_\theta(z_t, r, t)}{t}. \tag{2}$$

pMF motivates Eq. 2 by a *generalized manifold hypothesis*: extending the argument of Li & He (2025) that denoised images, unlike noise or velocity, lie on a low-dimensional manifold, it posits that the induced field $X(z_t, r, t) := z_t - t\,u(z_t, r, t)$ lies on such a generalized manifold of denoised images for every (r, t). This holds at the boundaries, where X is the posterior mean ($r = t$) or the ODE endpoint ($r = 0$), but for $0 < r < t$ it is supported only empirically by visual inspection.

Consistency Trajectory Models. CTM (Kim et al., 2024) learns any-to-any timestep transitions. To strictly guarantee the boundary condition $f_\theta(z_t, t, t) = z_t$, CTM isolates a data predictor $g_\theta(z_t, t, r)$ to define its transition mapping:

$$f_\theta(z_t, t, r) = \frac{r}{t}z_t + \frac{t - r}{t}g_\theta(z_t, t, r), \tag{3}$$

trained by matching a direct step from t to r with a step through an intermediate $s \in (r, t)$ under a stop-gradient target network.

4 INTERVAL DENOISER MODELS

4.1 FUNDAMENTALS OF INTERVAL DENOISING

Analyzing the Flow Matching ODE. We seek a step from z_t to z_r that depends only on the instantaneous denoiser. The resulting model is the pixel Interval Denoiser (pID). For flow matching (Lipman et al., 2023; Albergo et al., 2025) operating in image space, the ODE is commonly defined using the instantaneous denoiser $x(z_t, t)$ at time t:

$$\frac{dz_t}{dt} = \frac{z_t - x(z_t, t)}{t}. \tag{4}$$

Dividing Eq. 4 by t and rearranging gives the exact differential $\frac{d}{dt}\left(\frac{z_t}{t}\right) = -\frac{x(z_t,t)}{t^2}$. Integrating from a target time r to the current time t provides the exact state update:

$$z_r = \frac{r}{t}z_t + r\int_r^t \frac{x(z_\tau, \tau)}{\tau^2}d\tau. \tag{5}$$

Definition of the Interval Denoiser. To parameterize this integral, we define the *Interval Denoiser* as the normalized, weighted aggregation of instantaneous predictions over $[r, t]$:

$$X(z_t, r, t) = \frac{t \cdot r}{t - r} \int_r^t \frac{x(z_\tau, \tau)}{\tau^2} d\tau. \tag{6}$$

The Generalized Manifold Hypothesis. We first characterize X exactly.

Proposition 1. *Let $(z_\tau)_{\tau \in (0,t]}$ be the solution of Eq. 4 through z_t, continuous at $\tau = 0$ with endpoint z_0, and assume the denoiser is optimal, $x(z_\tau, \tau) = \mathbb{E}[x_0 \mid z_\tau]$, and continuous in τ. Then for $0 < r < t$, $X(z_t, r, t) = \mathbb{E}_\tau[\mathbb{E}[x_0 \mid z_\tau]]$ with $\tau \sim p(\tau) = \frac{tr}{t-r}\tau^{-2}$ on $[r, t]$. Moreover, $\lim_{r \to t} X(z_t, r, t) = x(z_t, t)$ and $\lim_{r \to 0} X(z_t, r, t) = z_0$.*

Proof. For $0 < r < t$, $p(\tau) \geq 0$ integrates exactly to 1 on $[r, t]$, so it is a probability density. As $r \to t$, the mean value theorem for integrals gives $X \to x(z_t, t)$. For $r \to 0$, Eq. 5 reads $z_r = \frac{r}{t} z_t + \frac{t-r}{t} X(z_t, r, t)$, i.e., $X(z_t, r, t) = \frac{t z_r - r z_t}{t - r} \to z_0$ by continuity of the trajectory at 0. ■

In the terminology of Li & He (2025) and Lu et al. (2026), the generalized manifold hypothesis states that denoised images, in contrast to noise or velocity vectors, form a low-dimensional, image-like set $\mathcal{M}$, and that the image-space target belongs to it for every (r, t). pMF motivates this at the boundaries and supports it with visual inspection in between. Prop. 1 supplies the structure behind it: X is a weighted average of denoised images $\mathbb{E}[x_0 \mid z_\tau]$ along a single trajectory. Hence X lies in the convex hull of the data support, coincides with the instantaneous denoiser as $r \to t$, and its weight concentrates at the low-noise end of the interval:

$$\int_r^{kr} p(\tau)\, d\tau = \frac{t}{t - r}\Big(1 - \frac{1}{k}\Big) \geq 1 - \frac{1}{k} \qquad \text{for } kr \leq t, \tag{7}$$

so at least half of the weight lies on $[r, 2r]$. For every (r, t), X is therefore an average of denoised images rather than a noise or velocity vector, which is the property that Li & He (2025) identify as making direct pixel-space regression tractable. Two things Prop. 1 does not claim: membership in the data manifold of clean images, which holds only at $r = 0$, and a bound on the dimension of the set of such averages.

Boundary cases. At $r = t$, $X(z_t, t, t) = x(z_t, t)$ is the posterior mean. At $r = 0$ we define $X(z_t, 0, t)$ by the limit in Prop. 1, so $X(z_t, 0, t) = z_0$ is the endpoint of the ODE trajectory, a sample from p_{data} when $z_t \sim p_t$, since the flow of Eq. 4 transports p_t to p_{data}.

Interval Denoiser Identity. To construct a training objective, we isolate the integral in Eq. 6,

$$\frac{t - r}{t \cdot r} X(z_t, r, t) = \int_r^t \frac{x(z_\tau, \tau)}{\tau^2}\, d\tau, \tag{8}$$

differentiate both sides with respect to t and multiply by t^2, which yields the fundamental *Interval Denoiser Identity*:

$$X(z_t, r, t) + \frac{t(t - r)}{r} \frac{d}{dt} X(z_t, r, t) = x(z_t, t). \tag{9}$$

At $r = t$ the identity reduces to plain denoising, $X = x$. Multiplying Eq. 9 by r/t and letting $r \to 0$ gives $\frac{d}{dt} X(z_t, 0, t) = 0$, i.e., the Interval Denoiser must be constant along the trajectory, which is precisely the consistency condition of continuous-time consistency models (Song et al., 2023; Lu & Song, 2025). The identity thus interpolates between flow matching ($r = t$), consistency models ($r = 0$), and MeanFlow and CTM in between (Sec. 4.3).

4.2 Training and Inference Strategy

Following Geng et al. (2026), the network has two jointly trained heads: a primary head predicting $X_\theta(z_t, r, t)$ and an auxiliary head predicting the instantaneous denoiser $x_\theta(z_t, t)$, trained with the denoising loss on $x_\theta(z_t, t) - x_0$ under the same metric as Eq. 20. The auxiliary head only provides the JVP tangent during training and is discarded at inference.

Algorithm 1 pID: Training.

```
# net: interval denoiser network
# x0: clean images in pixels

t, r = sample_t_r()
e = randn_like(x0)
z = (1 - t) * x0 + t * e

x = net(z, t, t)
X, dXdt = jvp(net, (z, r, t), (z - x, 0, t))

error = X + (t - r)/r * stopgrad(dXdt) - x0
loss = metric(error)
```

Algorithm 2 pID: Sampling.

```
# net: trained interval denoiser network
# timesteps: from t_0 = 1 to t_N = 0

z = randn(shape)
for i in range(N):
    t, r = timesteps[i], timesteps[i+1]
    X = net(z, r, t)
    z = (r * z + (t - r) * X) / t
return z
```

Note: `jvp` returns the function output and the JVP, and `metric` is the loss of Sec. 4.4. For simplicity, the auxiliary head and its denoising loss, class and guidance conditioning, the clamp of r and residual clipping are omitted here, so the instantaneous prediction appears as `net(z, t, t)`.

Computing the Time Derivative. Evaluating Eq. 9 requires the total time derivative $\frac{d}{dt}X_\theta(z_t, r, t) = \partial_z X_\theta \frac{dz_t}{dt} + \partial_t X_\theta$ (the target time r is independent of t). Evaluating $\frac{dz_t}{dt}$ with the network's own auxiliary instantaneous prediction $x_\theta(z_t, t)$ yields:

$$\frac{d}{dt}X_\theta(z_t, r, t) = \partial_z X_\theta \left(\frac{z_t - x_\theta(z_t, t)}{t}\right) + \partial_t X_\theta. \tag{10}$$

It is computed by a Jacobian-Vector Product (JVP). Absorbing the factor t of the coefficient $\frac{t(t-r)}{r}$ into the tangent, $[z_t - x_\theta, 0, t]$, yields the quantity denoted JVP_{X_θ}.

Training. Because the exact denoiser $x(z_t, t)$ is intractable, we substitute the ground-truth image x_0. To avoid higher-order derivatives, the gradient is detached through the JVP, where θ^- denotes the current parameters with gradients detached (no separate target network). This yields the regression objective (Alg. 1):

$$\mathcal{L}(\theta) = \mathbb{E}_{t,r,x_0,\epsilon}\left[\left\|X_\theta + \tfrac{t-r}{r}\text{JVP}_{X_{\theta^-}} - x_0\right\|_2^2\right]. \tag{11}$$

Eq. (11) is the idealized objective. Its coefficient $(t-r)/r$ diverges as the target time approaches the data end, $r \to 0$. A related singularity appears in x-prediction flow matching, where converting the predicted image to the velocity of Eq. (4) divides by t, which vanishes at the data end, and this denominator is clamped in practice (Li & He, 2025). Unlike there, the sampling update (Eq. 12) involves no division by r, so the singularity affects only the training target. Sec. 4.4 gives the implemented objective, including the clamp.

Sampling. At inference, substituting the network X_θ back into the exact update rule (Eq. 5) yields the step from t to r :

$$z_r = \frac{1}{t}\Big(r z_t + (t-r)X_\theta(z_t, r, t)\Big). \tag{12}$$

The sampling procedure is summarized in Alg. 2.

4.3 Relation to Prior Work

Comparing Eq. 3 with Eq. 12 shows that CTM's data predictor is the Interval Denoiser, $g_\theta \equiv X_\theta$, and the continuous-time limit of CTM's loss (target network identified with the online one) formally recovers our objective scaled by r^2/t^4 (Prop. 3, Appendix A). We now relate the Interval Denoiser to MeanFlow.

Connection to MeanFlow. MeanFlow updates the trajectory from t to r via the average velocity, $z_r = z_t - (t-r)u(z_t, r, t)$. Our sampling update (Eq. 12) rearranges to the identical form $z_r = z_t - (t-r)\frac{z_t - X(z_t,r,t)}{t}$, and equating the two steps formally connects the Interval Denoiser to the average velocity:

$$u(z_t, r, t) = \frac{z_t - X(z_t, r, t)}{t}. \tag{13}$$

Whereas pMF introduces this mapping as a substitution, here it follows from the ODE: the quantity pMF predicts is the Interval Denoiser, which Prop. 1 characterizes for every (r, t). The two regression spaces are strictly equivalent up to a scaling factor (dependencies on z_t, r, t omitted):

Proposition 2. *Given $u = \frac{z_t - X}{t}$, $v = \frac{z_t - x}{t}$, and $\frac{dz_t}{dt} = v$, the MeanFlow identity is strictly equivalent to the Interval Denoiser identity scaled by $\frac{r}{t^2}$:*

$$v - u - (t - r)\frac{du}{dt} = \frac{r}{t^2}\left[X + \frac{t(t-r)}{r}\frac{dX}{dt} - x\right].$$

The squared norm of the residual thus carries the factor $\frac{r^2}{t^4}$ (proof in Appendix A).

Stop-Gradient Placement and First-Order Exactness. Prop. 2 concerns the identities themselves. The objectives differ in where the stop-gradient is placed. pMF constructs its loss by substituting $u_\theta = (z_t - X_\theta)/t$ into the Improved MeanFlow objective (Eq. 1), whose stop-gradient encloses the total derivative $\frac{d}{dt}u_\theta$. Denoting $X_\theta \equiv X_\theta(z_t, r, t)$ and by x_θ the instantaneous prediction used as the JVP tangent (for pMF, $x_\theta \equiv X_\theta(z_t, t, t)$), the quotient rule gives

$$\frac{d}{dt}u_\theta = \frac{X_\theta - x_\theta}{t^2} - \frac{1}{t}\frac{dX_\theta}{dt}, \tag{14}$$

so the pMF training objective takes the following form:

$$\mathcal{L}_{\mathrm{pMF}} = \mathbb{E}_{t,r,x_0,\epsilon}\Big[\frac{1}{t^2}\Big\|X_\theta + (t-r)\,\mathrm{sg}\Big(\frac{dX_\theta}{dt} - \frac{X_\theta - x_\theta}{t}\Big) - x_0\Big\|_2^2\Big]. \tag{15}$$

Here (and for the original MeanFlow objective, Appendix A) the stop-gradient encloses not only the time derivative $\frac{d}{dt}X_\theta$ but also the zeroth-order network outputs X_θ and x_θ, which are not bootstrapped quantities. Our formulation stops the gradient only through the time derivative. To make the consequence precise, let R_X be the residual of the Interval Denoiser identity with x_0 in place of x, and $\mathcal{L}_X(\theta) = \mathbb{E}\|R_X\|_2^2$ the corresponding objective without any stop-gradient:

$$R_X := X_\theta + \frac{t(t-r)}{r}\frac{dX_\theta}{dt} - x_0, \tag{16}$$

$$\nabla_\theta \mathcal{L}_X = 2\,\mathbb{E}\big[(\partial_\theta X_\theta)^\top R_X\big] + 2\,\mathbb{E}\Big[\tfrac{t(t-r)}{r}\big(\partial_\theta \tfrac{dX_\theta}{dt}\big)^\top R_X\Big]. \tag{17}$$

We call the first term the *first-order gradient*: it is the only part that does not differentiate through the time derivative (and hence needs no second-order derivatives of the network). The pID update (Eq. 11) is exactly this first-order gradient.

Proposition 4. *Let*

$$D := t\,\frac{d}{dt}X_\theta - \frac{r}{t-r}\big(x_\theta - X_\theta\big) \tag{18}$$

denote the violation of Eq. 9 by the current network (with x_θ in place of x). The residuals regressed by pID (Eq. 11) and pMF (Eq. 15) are

$$R_{\mathrm{pID}} = (x_\theta - x_0) + \frac{t-r}{r}\,D, \qquad R_{\mathrm{pMF}} = (x_\theta - x_0) + \frac{t-r}{t}\,D, \tag{19}$$

and both updates have the form $(\partial_\theta X_\theta)^\top R$ up to a positive scalar. In particular $R_{\mathrm{pID}} = R_X$ and $R_{\mathrm{pMF}} - R_{\mathrm{pID}} = -\frac{(t-r)^2}{r\,t}\,D$.

Three consequences follow (proof in Appendix A). *(i) Exactness and bias.* Since $R_{\mathrm{pID}} = R_X$, the pID update is the first-order gradient of $\mathcal{L}_X$, whereas the pMF update deviates from it by $-\frac{(t-r)^2}{rt}(\partial_\theta X_\theta)^\top D$ (up to a positive scalar), a bias proportional to the identity violation that vanishes only once the network satisfies Eq. 9. Both objectives therefore share the same fixed point ($D = 0$ with $x_\theta = \mathbb{E}[x_0 \mid z_t]$), so we expect them to end up close under the same training recipe, consistent with the similar quality we observe once training is stable (Sec. 5.2). *(ii) Interpretation.* pMF's placement damps the bootstrapped derivative term by r/t, so the two weights differ by at most a factor of two for $r \geq t/2$, and both vanish as $r \to t$. *(iii) Wide intervals.* The identity weight $\frac{t-r}{r}$ diverges as $r \to 0$, whereas pMF's weight $\frac{t-r}{t}$ is bounded by one. First-order exactness thus comes at the price of an ill-conditioned target for wide intervals, which is precisely where residual clipping is required (Sec. 4.4).

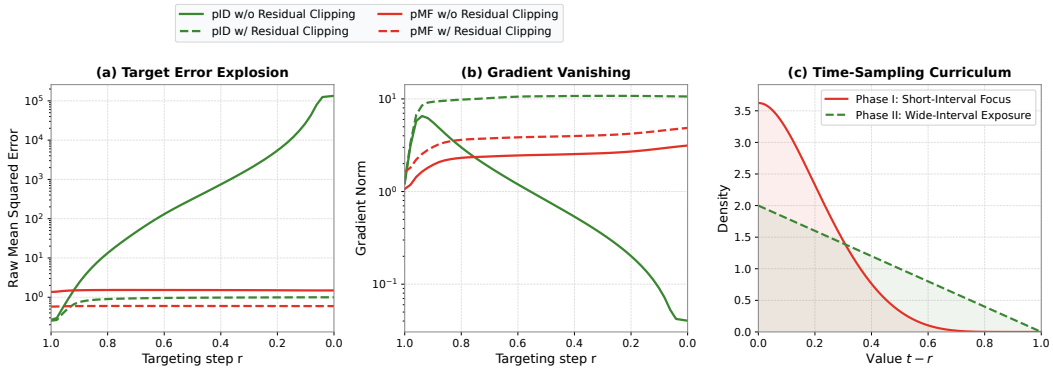

Figure 2: Training dynamics and interval sampling, pID vs. pMF. **(a)** Raw MSE of the regression target at $t = 1$, exploding for pID as $r \to 0$ without residual clipping, as predicted by the weight $(t-r)/r$ of Prop. 4. **(b)** Gradient norms of the logarithmic loss (Eq. 20) at $t = 1$, vanishing for pID over wide intervals without clipping, whereas pMF is stable. Residual clipping stabilizes pID. **(c)** Density of the interval $t - r$ in the two curriculum phases (Phase I logit-normal, Phase II uniform).

4.4 Design Decisions

Logarithmic Objective. We adopt the logarithmic objective of recent few-step models (Geng et al., 2026; Peng et al., 2026; Lu et al., 2026), whose gradient $\frac{1}{e+\delta}\nabla_\theta e$ (e the squared error, δ a small constant) equals that of the adaptively weighted loss $\frac{e}{\mathrm{sg}(e+\delta)}$, and evaluate the Interval Denoiser in MeanFlow-loss space, since x-prediction with a v-space loss performs best (Li & He, 2025). By Prop. 2 this induces the weight $\beta = r^2/t^4$ (with r replaced by $\tilde{r} = \max(r, r_{\min})$, see Denominator Clip). With X_{tar} the regression target,

$$\mathcal{L}(\theta) = \mathbb{E}_{t,r,x_0,\epsilon}\Big[\log\Big(\beta\big\|X_\theta - X_{\text{tar}}\big\|_2^2 + \delta\Big)\Big]. \tag{20}$$

For $\beta e \gg \delta$ the gradient equals $\nabla_\theta e/e$ irrespective of β, so β only reduces the contribution of samples with $\beta e \lesssim \delta$, i.e., of very short target times r. As in iMF and pMF, this weight is a robust per-sample reweighting, so the stationary points of the implemented losses are reweighted conditional means. In MeanFlow-loss space the residual

$$\frac{r}{t^2}R_X = \frac{r}{t^2}(X_\theta - x_0) + \frac{t-r}{t}\frac{dX_\theta}{dt} \tag{21}$$

is well defined at $r = 0$, where it reduces to the consistency condition $\frac{dX_\theta}{dt} = 0$ of Sec. 4.1.

Denominator Clip. In practice we replace r by $\tilde{r} = \max(r, r_{\min})$ with $r_{\min} = 0.05$ in the denominator of the identity weight and in β, which freezes the weight at $(t-r)/r_{\min}$ for $r < r_{\min}$. The value follows Li & He (2025) and the released code of pMF (Lu et al., 2026), which clip the denominator of their x-to-velocity conversion at 0.05. Under the curriculum below, the choice of $r_{\min}$ is immaterial ($r_{\min} = 0.01$ vs. 0.05: 7.53 vs. 7.55 FID for pID-B/16 at 160 epochs, Appendix F).

Residual Stabilization. Few-step generation requires training over wide intervals ($t \approx 1$, $r \to 0$), where pID applies the identity weight $(t-r)/r$ to the bootstrapped derivative term while pMF applies the bounded weight $(t-r)/t$ (Prop. 4). pID therefore produces extreme raw regression errors there (Fig. 2a), whereas pMF remains stable. The gradient of Eq. 20 with respect to the prediction is $2\beta R/(\beta\|R\|^2 + \delta)$, of magnitude $\approx 2/\|R\|$ once $\beta\|R\|^2 \gg \delta$. Since $\|R\|$ grows like $\frac{t-r}{r}\|D\|$, the gradient vanishes like r/t on exactly these samples (Fig. 2b), so the optimizer updates the network almost entirely from easier, short-interval samples. To restore a balanced learning signal, we apply residual clipping (Lu & Song, 2025; Peng et al., 2026). The raw error Δ is computed through the stop-gradient network, clipped to $[-1, 1]$, and turned into the regression target:

$$\Delta = X_{\theta^-} + \frac{t-r}{\tilde{r}}\mathrm{JVP}_{X_{\theta^-}} - x_0, \qquad X_{\text{tar}} = X_{\theta^-} - \mathrm{clip}(\Delta, -1, 1). \tag{22}$$

Together with the logarithmic weighting, the implemented update is therefore equal to $(\partial_\theta X_\theta)^\top \operatorname{clip}(R_X, -1, 1)$, with r replaced by $\tilde{r}$ in the weight of R_X, times a positive per-sample scalar. It keeps the stop-gradient structure of Prop. 4, coincides, per sample and up to this scalar, with the first-order gradient of $\|R_X\|_2^2$ wherever neither clipping nor the clamp is active, and bounds the derivative term for wide intervals without introducing the damping of Prop. 4(ii).

Time-Sampling Curriculum. Clipping stabilizes large steps, but standard time distributions still under-sample them, limiting few-step quality (Table A1). We therefore apply a two-phase time-sampling curriculum (Sun, 2026). Unlike α-Flow (Zhang et al., 2026), which anneals the loss from flow matching to MeanFlow, we keep the loss fixed and shift the sampling distribution of (t, r), the larger and smaller of two independent draws: Phase I uses logit-normal$(0.8, 0.8)$, concentrated on short intervals (Fig. 2c), so that the network first learns the local velocity field, and Phase II switches to the uniform distribution. The shift is large: the uniform distribution places 25% of the pairs on intervals $t - r > 0.5$ and 0.5% in the one-step corner $\{r < 0.05,\ t > 0.95\}$, against 2.6% and about 10^{-8} for logit-normal$(0.8, 0.8)$, so the one-step query receives direct supervision almost only in Phase II.

5 Experiments

5.1 Experimental Setup

We evaluate on ImageNet 256×256 (Deng et al., 2009) with the pMF-B/16 and pMF-L/16 architectures plus an auxiliary head (Appendix D), directly in pixel space without pretrained autoencoders. Following Geng et al. (2026), classifier-free guidance (Ho & Salimans, 2022) is folded into training through guidance-scale conditioning (Appendix C), so every NFE is exactly one forward pass. Ablations train pID-B/16 from scratch for 160 epochs with a single seed. We report FID (Heusel et al., 2017) on 50,000 samples, and Inception Score (IS) (Salimans et al., 2016) for ablations (details in Appendix D).

5.2 Ablation Study

Residual Clipping. Clipping accelerates convergence for pID (Fig. A1, Appendix B), improving the final 1-NFE FID from 9.78 to 9.25, whereas for pMF it changes little (9.56 vs. 9.34), in line with unclipped pMF being stable at large steps. With stability restored, pID performs on par with pMF (9.25 vs. 9.34, single seed), consistent with the shared fixed point of Prop. 4(i).

Time-Sampling Curriculum. Table A1 (Appendix B) evaluates the curriculum on pID: best static samplers yield around 9.25 FID, whereas switching from LN$(0.8, 0.8)$ to Uniform lowers the 1-NFE FID to 7.55, a larger effect than residual clipping (9.78 to 9.25). The Phase I location matters ($\mu = 0.0$ instead of 0.8: 8.01 vs. 7.55), a full transition to the uniform distribution outperforms a 50% mix, and the transition epoch has a minor effect ($T_s \in \{120, 140, 150\}$: 7.55–7.69 FID). We make no claim that this gain is specific to pID (see Limitations).

Sampling with 2-NFE. Two-step sampling is sensitive to the intermediate time k (Fig. A2, Appendix B): FID reaches 6.87 at $k = 0.85$ (vs. 7.55 for 1-NFE), whereas $k < 0.5$ is worse than one step.

5.3 Main Results and Comparisons

Comparisons on ImageNet 256×256. Training for 320 epochs improves the 1-NFE FID of pID-B/16 from 7.55 to 6.28, and pID-L/16 reaches 4.55 (Table 1, left). The gray entries of Table 1 are not directly comparable: perceptual losses leak features of ImageNet-pretrained networks into the samples, which can lower FID (computed with an ImageNet-pretrained Inception-V3) without a gain in sample quality (Kynkäänniemi et al., 2023; Song & Dhariwal, 2024), latent-space models and GANs rely on pretrained autoencoders or adversarial training, and the multi-step pixel-space models use 50 to 1000 sampler steps. Among pixel-space fast-forward models without perceptual losses, pID-L/16 obtains the best reported 1-NFE FID (4.55).

Table 1: Class-conditional generation on ImageNet 256×256 (FID on 50,000 samples). **Left**: pixel-space diffusion/flow models (pID and pMF: 320 epochs, EPG-L/16: 560). Guidance is folded into training for pID (Appendix C), so NFE counts forward passes. The pMF rows without perceptual losses use pMF's original recipe (static sampler, no clipping or curriculum). The rows with perceptual losses (gray) are not directly comparable. **Right**: other model families for reference (GANs, latent-space few-step models, multi-step pixel-space models). They rely on adversarial training, pre-trained autoencoders or many sampling steps, and are not directly comparable either. For multi-step models, NFE is the number of sampler steps.

method	params	FID ↓
1-NFE pixel-space diffusion/flow from scratch		
EPG-L/16 (2026)	540M	8.82
pMF-B/16 (2026)	118M	8.71
pID-B/16 (ours)	118M	6.28
pID-L/16 (ours)	411M	**4.55**
1-NFE pixel-space diffusion/flow with perceptual losses		
pMF-B/16 (2026)	118M	3.12
pMF-L/16 (2026)	411M	2.52
2-NFE pixel-space diffusion/flow from scratch		
pID-B/16 (ours)	118M	5.61
pID-L/16 (ours)	411M	**3.98**

method	params	NFE	FID ↓
GANs			
BigGAN-deep (2019)	56M	1	6.95
GigaGAN (2023)	569M	1	3.45
StyleGAN-XL (2022)	166M	1	2.30
latent-space diffusion/flow, few-step			
iCT-XL/2 (2024)	675M	1	34.24
Shortcut-XL/2 (2025)	675M	1	10.60
MeanFlow-L/2 (2025)	459M	1	3.84
iMF-L/2 (2026)	409M	1	1.86
pixel-space diffusion/flow, multi-step			
ADM-G (2021)	554M	250	4.59
RIN (2023)	410M	1000	3.42
PixNerd-L/16 (2025)	458M	50	2.64
JiT-L/16 (2025)	459M	50	2.36

Where the gain comes from. At 160 epochs, the 2.0 FID gap between pMF's original recipe (9.56) and our best pID-B/16 (7.55) splits into 0.3 from clipping and parameterization combined (9.56 to 9.25, pMF with clipping: 9.34) and 1.7 from the curriculum (9.25 to 7.55). The pMF entry in Table 1 uses pMF's original recipe, so its gap to pID-B/16 (8.71 vs. 6.28) reflects mainly the training recipe. The curriculum changes only how (t, r) is sampled and can be applied to pMF unchanged.

6 Conclusion and Limitations

We presented the *Interval Denoiser*, a framework for few-step, latent-free generation derived from the flow matching ODE. It identifies the image-space target of pixel-space few-step models as an interval-weighted average of instantaneous denoisers, recovers pMF's substitution and CTM's decoder, and shows that our update is the exact first-order gradient of the identity while pMF's damps the bootstrapped term by r/t. The exact weighting is ill-conditioned for wide intervals, which residual clipping addresses. A time-sampling curriculum that shifts training toward wide intervals gives the largest empirical gain. Without autoencoders or perceptual losses, pID-L/16 reaches 4.55 FID at 1-NFE and 3.98 at 2-NFE. *Limitations.* All ablations use a single seed, so differences of a few tenths of an FID point (e.g., across T_s) should not be over-interpreted. We did not retrain pMF with the two-phase curriculum, so the pMF numbers in Table 1 do not isolate the parameterization from the training recipe. With clipping alone the two perform on par (Sec. 5.2), and since they share their fixed point (Prop. 4(i)) we expect the same under the full recipe, but we have not verified this.

AI use statement

We used generative AI tools (Claude Opus 5.5, Claude Fable 5.1) to polish the writing, draft some passages of this paper (e.g., parts of Appendix D), assist with LaTeX formatting, help write and debug code, search for related work, check references, and obtain feedback on drafts. All propositions and proofs were developed by the authors. The tools only rewrote the authors' existing statements and proofs for clarity, and every rewritten step was checked by the authors, and all AI-assisted code was reviewed and tested by the authors. We have not used generative AI tools for research ideation, experiment design, data generation or data analysis. We have reviewed all AI-assisted work, and we

take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

All architectural, optimization and sampling hyperparameters are listed in Appendix D (Table A2), the guidance mechanism is described in Appendix C, training and sampling are specified by Algs. 1 and 2 together with Sec. 4.4, complete proofs of Props. 2–4 are given in Appendix A, and the compute budget is reported in Appendix E.

References

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26:1–80, 2025.

A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, USA, 2006.

Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. In *International Conference on Learning Representations*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

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

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

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 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*, 2026.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour, 2017.

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. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. In *International Conference on Machine Learning*, pp. 14619–14637. PMLR, 2023.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cecista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://github.com/KellerJordan/Muon, 2024.

M. Kang, J.-Y. Zhu, R. Zhang, J. Park, E. Shechtman, S. Paris, and T. Park. Scaling up GANs for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

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

Jin-Young Kim, Hyojun Go, Lea Bogensperger, Julius Erbach, Nikolai Kalischek, Federico Tombari, Konrad Schindler, and Dominik Narnhofer. Understanding, accelerating, and improving MeanFlow training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of ImageNet classes in Fréchet inception distance. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=4oXTQ6m_ws8.

J. Lei, K. Liu, J. Berner, H. Yu, H. Zheng, J. Wu, and X. Chu. There is no VAE: End-to-end pixel-space generative modeling via self-supervised pre-training. In *International Conference on Learning Representations*, 2026.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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

Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows, 2026.

Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. FACM: Flow-anchored consistency models. In *International Conference on Learning Representations*, 2026.

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. In *Advances in Neural Information Processing Systems*, 2025.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

A. Sauer, K. Schwarz, and A. Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022.

Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arXiv preprint arXiv:2510.15301*, 2025.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *International Conference on Learning Representations*, 2024.

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

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023.

Pengwei Sun. Curriculum sampling: A two-phase curriculum for efficient training of flow matching. In *2nd DeLTa Workshop at the International Conference on Learning Representations (ICLR)*, 2026.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, 2008.

Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. PixNerd: Pixel neural field diffusion, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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*, 2026.

R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

A PROOFS

Proof of Proposition 2. Using the quotient rule and substituting the ODE $\frac{dz_t}{dt} = \frac{z_t - x}{t}$, the total time derivative $\frac{du}{dt}$ expands as:

$$\frac{du}{dt} = \frac{1}{t}\frac{dz_t}{dt} - \frac{z_t - X}{t^2} - \frac{1}{t}\frac{dX}{dt} = \frac{X - x}{t^2} - \frac{1}{t}\frac{dX}{dt}. \tag{23}$$

Substituting this derivative, alongside u and v, into the MeanFlow residual yields:

$$\begin{aligned} & v - u - (t - r)\frac{du}{dt} \\ &= \frac{X - x}{t} - (t - r)\left(\frac{X - x}{t^2} - \frac{1}{t}\frac{dX}{dt}\right) \\ &= \frac{r}{t^2}\left[X + \frac{t(t - r)}{r}\frac{dX}{dt} - x\right]. \end{aligned} \tag{24}$$

Taking the squared L_2 norm of this residual extracts the $\frac{r^2}{t^4}$ scaling factor for the loss, establishing the formal equivalence of the two training objectives. ■

MeanFlow objective under the substitution. Substituting the parameterization $u_\theta = (z_t - X_\theta)/t$ into the original MeanFlow objective (Geng et al., 2025) also yields a zeroth-order term trapped inside the stop-gradient:

$$\mathcal{L}_{\mathrm{MF}} = \mathbb{E}\Big[\frac{1}{t^2}\Big\|X_\theta + (t - r)\,\mathrm{sg}\Big(\frac{dX_\theta}{dt} - \frac{X_\theta}{t}\Big) - \frac{r}{t}x_0\Big\|_2^2\Big]. \tag{25}$$

Connection to Consistency Trajectory Models. Comparing CTM's preconditioned mapping (Eq. 3) with our sampling update (Eq. 12) immediately establishes the structural equivalence of their denoiser parameterizations, $X_\theta \equiv g_\theta$.

Proposition 3. *The continuous-time limit of the CTM loss recovers the Interval Denoiser loss scaled by* $\frac{r^2}{t^4}$.

Proof. The discrete CTM objective minimizes trajectory discrepancy across a finite step h:

$$\mathcal{L}_{\text{CTM}} = \|f_\theta(z_t, t, r) - f_\theta(z_{t+h}, t+h, r)\|_2^2, \tag{26}$$

where, in the continuous-time limit considered here, the target network is identified with the online network. Dividing by h^2 and taking the limit $h \to 0^+$ converts this difference into the total time derivative

$$\lim_{h\to 0^+} \frac{1}{h^2}\mathcal{L}_{\text{CTM}} = \left\| \frac{d}{dt} f_\theta(z_t, t, r) \right\|_2^2. \tag{27}$$

Expanding this derivative with $X_\theta \equiv g_\theta$ yields:

$$\frac{d}{dt} f_\theta = -\frac{r}{t^2} z_t + \frac{r}{t}\frac{dz_t}{dt} + \frac{r}{t^2} X_\theta + \frac{t-r}{t}\frac{dX_\theta}{dt}. \tag{28}$$

Substituting $\frac{dz_t}{dt} = \frac{z_t - x}{t}$ and replacing x with x_0 yields:

$$\frac{d}{dt} f_\theta = \frac{r}{t^2}\left[X_\theta + \frac{t(t-r)}{r}\frac{dX_\theta}{dt} - x_0 \right]. \tag{29}$$

Therefore, it identically recovers Interval Denoiser prediction in MeanFlow loss space. ■

Proof of Proposition 4. Write $J := t\,\frac{d}{dt}X_\theta$ for the JVP in Alg. 1, so that by definition of D, $J = \frac{r}{t-r}(x_\theta - X_\theta) + D$. The pID residual (Eq. 11) is

$$R_{\text{pID}} = X_\theta + \tfrac{t-r}{r} J - x_0 = X_\theta + (x_\theta - X_\theta) + \tfrac{t-r}{r} D - x_0 = (x_\theta - x_0) + \tfrac{t-r}{r} D, \tag{30}$$

and it coincides with R_X because $\frac{t-r}{r} J = \frac{t(t-r)}{r}\frac{dX_\theta}{dt}$. Ignoring the (gradient-irrelevant) stop-gradient, the pMF residual (Eq. 15) is

$$\begin{aligned} R_{\text{pMF}} &= X_\theta - x_0 + \tfrac{t-r}{t}\big(J - X_\theta + x_\theta\big) = X_\theta - x_0 + \tfrac{t-r}{t}\Big[(x_\theta - X_\theta)\big(\tfrac{r}{t-r} + 1\big) + D\Big] \\ &= X_\theta - x_0 + (x_\theta - X_\theta) + \tfrac{t-r}{t} D = (x_\theta - x_0) + \tfrac{t-r}{t} D. \end{aligned} \tag{31}$$

Subtracting the two expressions gives $R_{\text{pMF}} - R_{\text{pID}} = \big(\frac{t-r}{t} - \frac{t-r}{r}\big)D = -\frac{(t-r)^2}{rt} D$. In both objectives every term other than the leading X_θ is inside a stop-gradient (or is data), so the updates are $2\,(\partial_\theta X_\theta)^\top R_{\text{pID}}$ and $\frac{2}{t^2}(\partial_\theta X_\theta)^\top R_{\text{pMF}}$ respectively. ■

B Additional Results

This appendix collects the convergence curves with and without residual clipping (Fig. A1), the time-sampling curriculum ablation (Table A1) and the 2-NFE analysis (Fig. A2) referenced in Sec. 5.2.

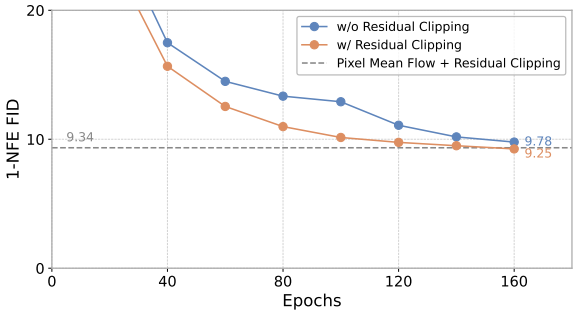

Figure A1: Effect of residual clipping on 1-NFE training. Bounding residual prevents gradient suppression, accelerating convergence and improving pID 1-NFE FID from 9.78 to 9.25. With gradient stability restored, our exact formulation achieves performance (9.25) comparable to pMF with residual clipping (9.34).

Table A1: Ablations on time-sampling curriculum for pID (ours). Models are trained for 160 epochs on ImageNet 256×256 (single seed). Phase I uses distribution p_1. At epoch T_s, Phase II introduces distribution p_2 with the specified mixing probability.

Phase I	**Phase II (Epoch T_s–End)**			**Metrics**	
p_1	p_2	Mix	T_s	**FID** ↓	**IS** ↑
Baselines (Static Sampling)					
Uniform	Uniform	—	—	9.26	178.1
LN(0.0, 0.8)	LN(0.0, 0.8)	—	—	10.67	152.8
LN(0.8, 0.8)	LN(0.8, 0.8)	—	—	9.25	188.0
Curriculum Ablations					
LN(0.8, 0.8)	Uniform	50%	120	7.87	200.0
LN(0.8, 0.8)	Uniform	50%	140	7.88	198.6
LN(0.8, 0.8)	Uniform	50%	150	7.85	198.3
LN(0.8, 0.8)	Uniform	100%	120	7.69	192.0
LN(0.8, 0.8)	Uniform	100%	140	**7.55**	**200.7**
LN(0.8, 0.8)	Uniform	100%	150	7.69	194.9
LN(0.0, 0.8)	Uniform	100%	120	8.10	186.2
LN(0.0, 0.8)	Uniform	100%	140	8.01	185.2
LN(0.0, 0.8)	Uniform	100%	150	8.27	180.3

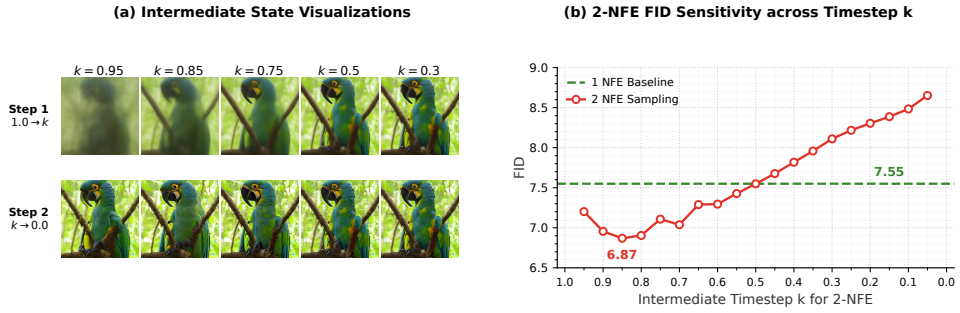

Figure A2: Analysis of pID 2-NFE intermediate sampling. (a) Step-1 outputs $X_\theta(z_1, k, 1)$ (top row) and final samples after Step 2, $k \to 0$ (bottom row). The state $z_k = kz_1 + (1-k)X_\theta$ itself is dominated by noise for large k and is not shown. (b) 2-NFE FID sensitivity across k, demonstrating that 2-step sampling outperforms the single-step baseline (7.55, dashed line) for $k > 0.5$, reaching optimal generation performance (FID 6.87) at $k = 0.85$.

C Classifier-Free Guidance

Following Improved MeanFlow (Geng et al., 2026), we incorporate classifier-free guidance (CFG) (Ho & Salimans, 2022) directly into training rather than at inference. By substituting our Interval Denoiser into the velocity guidance formula via the identity $u = (z_t - X)/t$, we construct the guided target:

$$x_{\text{cfg}} = x_0 + \left(1 - \frac{1}{\omega}\right)\Big(X_{\theta^-}(z_t, t, t \mid \mathbf{c}) - X_{\theta^-}(z_t, t, t \mid \emptyset)\Big). \tag{32}$$

Here, $\mathbf{c}$ and $\emptyset$ denote the conditional and unconditional classes, and ω is the guidance scale. We apply this by replacing the clean image x_0 with x_{cfg} in our regression objective. During training, both ω and the CFG interval are sampled and provided to the network as conditioning inputs. At inference, the guidance scale is an input of the network, so one sampling step costs exactly one forward pass. The analysis of Sec. 4.1 concerns the unguided ODE. As in Improved MeanFlow, guidance only replaces the regression target x_0 by x_{cfg}, i.e., the identities of Sec. 4.1 are applied to the guided flow whose instantaneous denoiser is the guided one.

D Implementation Details

We use the unmodified pMF-B/16 and pMF-L/16 architectures. Detailed configurations are in Table A2, and unspecified hyperparameters follow Pixel Mean Flow (pMF) (Lu et al., 2026).

Auxiliary head. Following Geng et al. (2026), the network employs two jointly trained output heads: a primary head predicting the Interval Denoiser $X_\theta(z_t, r, t)$, and an auxiliary head predicting the instantaneous denoiser $x_\theta(z_t, t)$. The total training objective is the sum of the pID loss for X_θ and the flow matching (denoising) loss for x_θ under the same metric. The auxiliary head is used only as the tangent of the JVP and receives no gradient from the pID loss. The stop-gradient network θ^- of Eqs. 11 and 22 is the current network with gradients detached (`stopgrad` in Alg. 1). No separate target or EMA network is used during training. Alg. 1 writes the boundary prediction of the auxiliary head as `net(z, t, t)` and omits its denoising loss, the clamp $\tilde{r}$ and the residual clipping of Sec. 4.4 for simplicity.

Time conditioning. Following MeanFlow and pMF, the network receives embeddings of both t and the interval length $t - r$ (the row "(t, r) cond" in Table A2), so the pair (t, r) is fully specified. The auxiliary head is evaluated at interval length 0.

Model selection. Following pMF, the EMA half-life, the guidance scale and interval, and the intermediate time k of 2-NFE sampling are selected by FID on 50,000 samples. Table 1 reports one fixed configuration per model, and Fig. A2 reports the full sweep over k.

EMA. Following pMF, we maintain multiple Exponential Moving Average (EMA) half-lives during training and select the best for inference.

Longer training. For 320-epoch runs (pID-B/16 and pID-L/16), we use the optimal ablation settings and transition to the uniform time sampler at epoch 140.

Inception Score of the final models. pID-B/16: 213.1 (1-NFE) and 224.5 (2-NFE). pID-L/16: 221.9 (1-NFE) and 243.0 (2-NFE), with $k = 0.8$ for 2-NFE sampling.

Baselines. Our pMF-B/16 reproduction at 160 epochs (without residual clipping) yields 9.56 FID, matching the value reported by Lu et al. (2026). The pMF numbers in Table 1 are those reported by Lu et al. (2026).

E Computational Budget

Models are trained on a single node with 8 NVIDIA H100 GPUs. pID-B/16 requires around 3 days for 160 epochs (576 H100-hours) and around 6 days for 320 epochs (1,152 H100-hours). pID-L/16 takes around 18 days for 320 epochs (3,456 H100-hours).

F Failed Experiments

We document directions that did not improve our framework using pID-B/16 at 160 epochs with residual clipping.

Denominator clipping values. The impact of $r_{\min}$ depends on the time sampler. Under a uniform sampler, reducing $r_{\min}$ from 0.05 to 0.01 improves FID from 9.26 to 8.62. Conversely, under logit-normal$(0.8, 0.8)$, it degrades FID from 9.25 to 9.62. However, our two-phase curriculum eliminates this sensitivity: setting $r_{\min} = 0.01$ yields 7.53 FID, comparable to our reported 7.55 FID using $r_{\min} = 0.05$.

Alternative consistency techniques. We explore several techniques from the broader consistency model literature. These include interpolating between the regression target and the network output (Peng et al., 2026), applying a tangent warmup to linearly scale the JVP term (Lu & Song, 2025;

Table A2: Configurations and hyperparameters. †: for ablation studies. Optimizer: Muon (Jordan et al., 2024). Class dropout follows Goyal et al. (2017). CFG settings follow Geng et al. (2026).

configs	pID-B/16	pID-L/16
epochs	160†/ 320	320
batch size	1024	
optimizer	Muon, $(\beta_1, \beta_2) = (0.9, 0.95)$	
learning rate	1e-3	
lr warmup	0 epoch	
weight decay, dropout	0.0	
ema half-life (Mimgs)	$\{1000, 2000\}$	
ratio of $r \neq t$	50%	
(t, r) cond	$(t,\ t-r)$	
cls drop	0.1	
CFG dist β_{cfg}	1	2
loss	$\log(\beta\|\cdot\|_2^2 + \delta)$, $\beta = \max(r, r_{\min})^2/t^4$	
δ	0.01	
denom. clip $r_{\min}$	0.05	
residual clip	$[-1, 1]$	
phase I sampler	logit-normal(0.8, 0.8)	
phase II sampler	uniform	
transition epoch T_s	140	

Sabour et al., 2025), and using alternative loss weightings (Kim et al., 2026). None tangibly improved generation quality or training stability.

G VISUALIZATION

Figure 1 shows selected pID-L/16 samples. Figures A3 and A4 provide uncurated pID-L/16 samples on ImageNet 256×256. Each block uses random seeds 1–24 in raster order. Both figures share the same initial noise, making corresponding cells directly comparable.

We use the settings from our reported evaluation. The intervals below are the guidance intervals of Appendix C, i.e., the range of t on which guidance is applied. 1-NFE (FID 4.55) uses CFG scale $\omega = 7.0$ on the guidance interval $t \in [0.1, 0.82]$, and 2-NFE (FID 3.98) uses $\omega = 7.0$ on $t \in [0.1, 0.74]$ and the intermediate time $k = 0.8$.

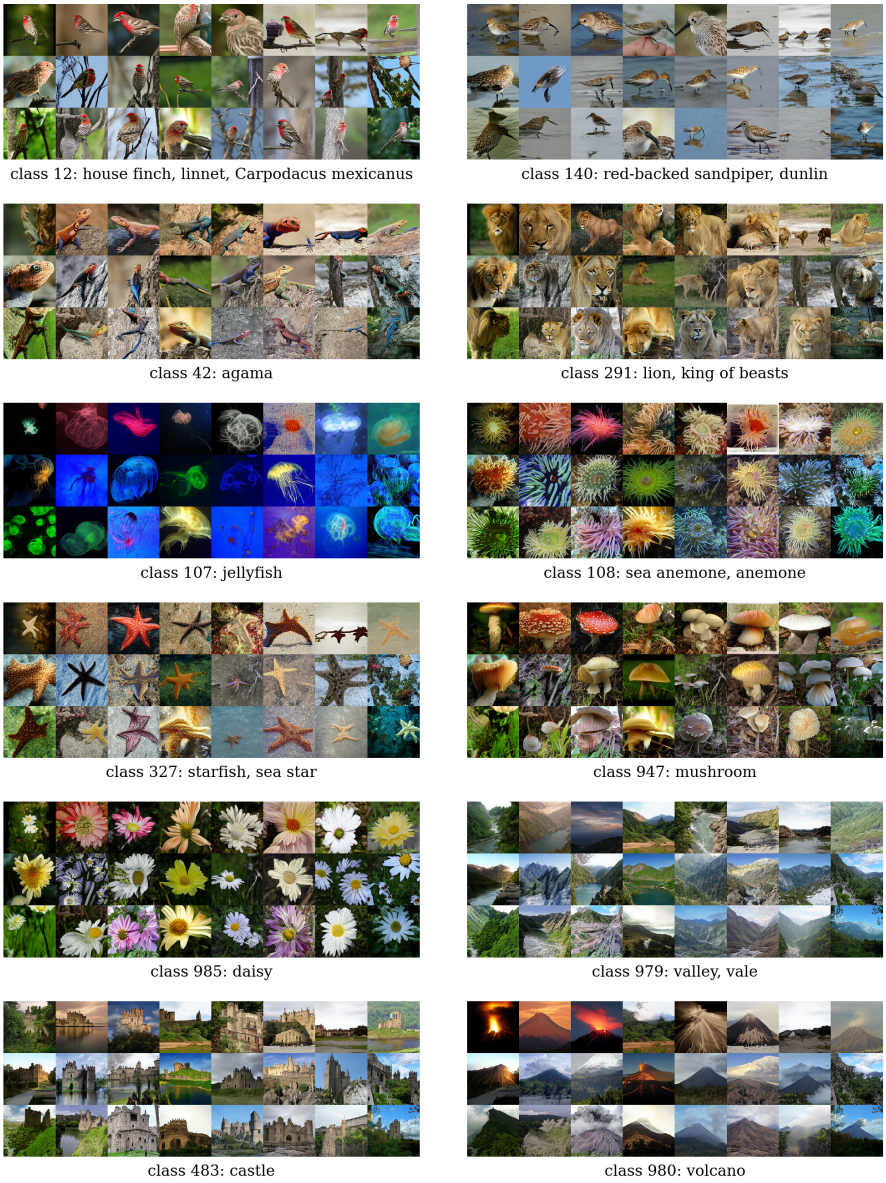

Figure A3: Uncurated 1-NFE pixel class-conditional generation samples of pID-L/16 on ImageNet 256×256.

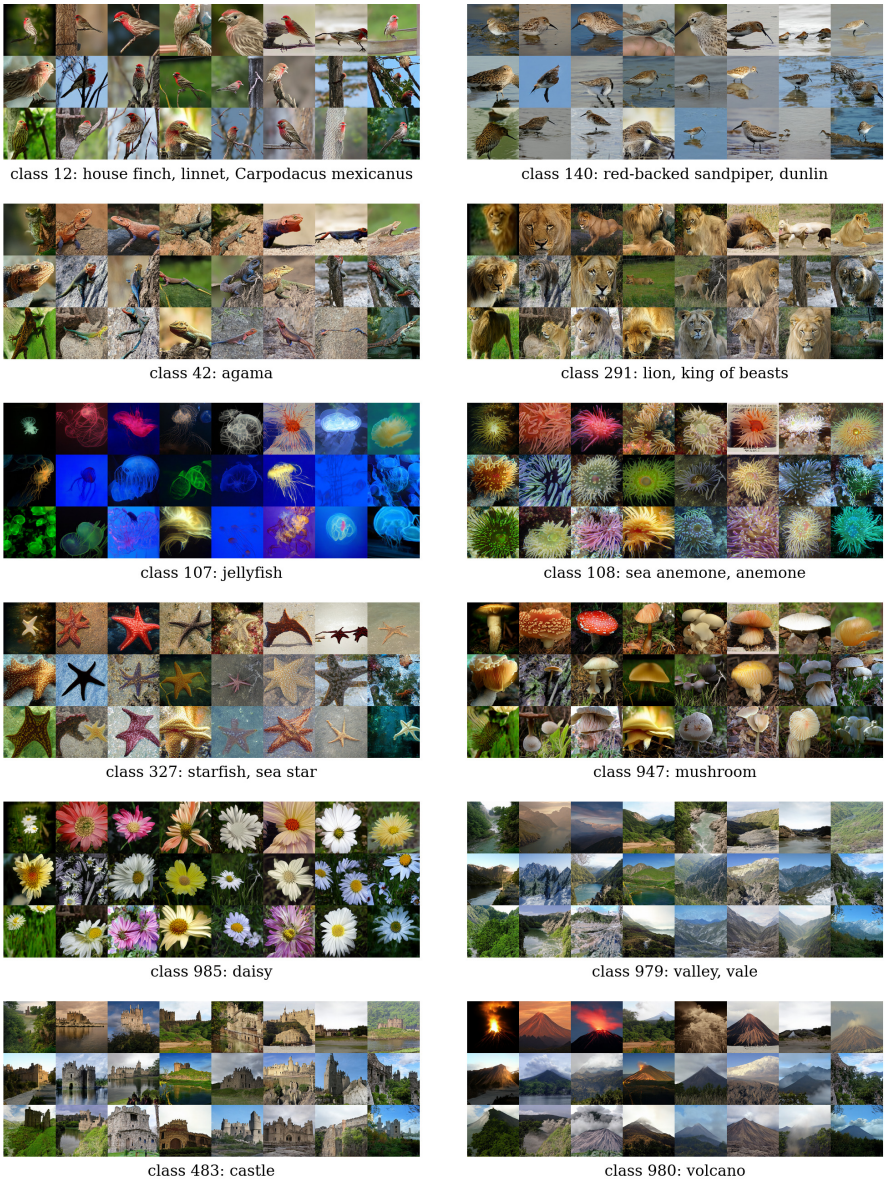

Figure A4: Uncurated 2-NFE pixel class-conditional generation samples of pID-L/16 on ImageNet 256×256.