IMPROVED DISTRIBUTIONAL DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributional Diffusion Models (DDMs) replace the standard mean-prediction denoiser with a *distributional* denoiser trained via a scoring rule objective, learning a stochastic approximation to $p(x_1 \mid x_t)$ rather than its conditional mean. However, scaling DDMs to modern image-generation settings faces two obstacles: (i) multi-particle training incurs overhead that scales with the number of particles, (ii) DDMs use globally fixed scoring rule hyperparameters, forcing a single trade-off across sampling budgets. We mitigate these limitations by deferring particle expansion to late transformer layers, and the hyperparameter trade-off by introducing time-dependent scoring rule schedules informed by the dynamical regimes of Biroli et al. (2024). Combined with a DiT-based latent setup, these changes make DDM training practical on class-conditional ImageNet-256^2, achieving 4.48 FID at 4 steps and 2.38 at 50 steps with DiT-XL/2, from a single model trained from scratch in one stage, without a teacher, self-distillation or JVPs. The result is a stochastic few-step generator whose FID does not degrade as the sampling budget grows from 4 to 50 NFE, and the same recipe transfers to text-to-image generation. Code and pre-trained models will be released with the camera-ready.



Figure 1: **Improved Distributional Diffusion Models (iDDM)** achieve competitive sample quality across 4–50 steps with a single model and an FID on ImageNet-256^2 of 4.48 and 2.38 while offering a *single-stage* training without distillation, CFG during training or self-bootstrapping.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) and flow-based variants (Lipman et al., 2023; Liu et al., 2023b; Albergo et al., 2025) have established themselves as a leading paradigm in generative modeling and found many applications in high-quality synthesis of images (Rombach et al., 2022; Saharia et al., 2022), video (Ho et al., 2022; Polyak et al., 2025), audio (Kong et al., 2021; Liu et al., 2023a) and proteins (Watson et al., 2023; Abramson et al., 2024). Adopting flow-matching notation (p_0 : noise, p_1: data), samples are generated by numerically

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.

integrating an Ordinary Differential Equation from noise to data. High-fidelity approximation of this dynamics typically requires a large number of function evaluations (NFE). This becomes especially challenging in the few-step regime, where the conditional law $p(x_1|x_t)$ can exhibit substantial variance at coarse discretizations. In this regime, trained denoisers which model the conditional mean $\mathbb{E}[x_1 \mid x_t]$ provide an insufficient summary of the posterior distribution.

Most existing approaches reduce this cost by improving the numerical solver (Karras et al., 2022; Lu et al., 2022; 2025; Zheng et al., 2023), distilling a multi-step model (Luhman & Luhman, 2021; Salimans & Ho, 2022; Sauer et al., 2025; Yin et al., 2024; Salimans et al., 2024), or directly training a few-step generator, often with consistency- or flow-map-based self-distillation objectives (Song et al., 2023; Frans et al., 2025; Boffi et al., 2025; Dieleman, 2026). Distributional Diffusion Models (DDMs) (De Bortoli et al., 2025b) pursue a complementary direction: they replace conditional-mean regression with a *stochastic* denoiser $\hat{x}_\theta(t, x_t, \xi)$ that, for each auxiliary noise $\xi \sim \mathcal{N}(0, \mathrm{Id})$, returns a sample from a learned approximation to $p(x_1 \mid x_t)$. The model is trained with a generalized energy score parameterized by an exponent $\beta \in (0, 2]$ and a trade-off weight $\lambda \in [0, 1]$, with a confinement term $\|x_1 - \hat{x}_\theta(x_t, t, \xi)\|_2^\beta$ which encourages the fidelity to the target, while an interaction term, weighted by λ, penalizes collapse of the sample population to a single prediction. Notably, setting $\lambda = 0, \beta = 2$ recovers the standard MSE diffusion loss. DDMs thus offer a non-adversarial alternative to modeling the full conditional distribution, not only its mean.

Despite their promise, DDMs remain impractical at modern model scale. We therefore ask: *what prevents distributional denoising from scaling, and what is needed to make it practical?* We identify two bottlenecks, both of which must be removed for DDMs to scale. **First**, the DDM objective draws m noise samples ξ per training example, with each particle requiring a separate denoiser evaluation, resulting in an $O(m)$ computational overhead. Yet most of this computation extracts the same features from x_t. **Second**, the original DDM uses fixed scoring rule hyperparameters (λ, β) across all noise levels, imposing the same fidelity–diversity trade-off throughout the diffusion trajectory, including near $t \approx 1$, where $p(x_1 \mid x_t)$ becomes concentrated, so that no single setting serves both coarse and fine sampling budgets. We remove both bottlenecks with two orthogonal modifications. **To reduce particle cost**, we process x_t once through most of the transformer and expand into m particles only in the final few layers, reducing the marginal cost of additional particles to a fraction of a full forward pass. **To adapt the objective across noise levels**, we replace fixed (λ, β) with time-dependent schedules $(\lambda(t), \beta(t))$ that tie the scoring rule to how concentrated $p(x_1 \mid x_t)$ is at each noise level. This lets the model emphasize diversity when $p(x_1 \mid x_t)$ is broad and more concentrated predictions when it is narrow. Whereas Biroli et al. (2024) analyze the dynamical regimes of the reverse process, we map them to the flow-matching SNR and use them to design the training objective. Removing both bottlenecks makes distributional denoising practical with large transformer backbones while retaining the goal of modeling $p(x_1 \mid x_t)$ rather than only its conditional mean. The result is a stochastic few-step generator, trained from scratch in a single stage, whose FID is non-increasing from 4 to 50 steps, so one checkpoint serves every budget in that range.

We demonstrate results on class-conditional ImageNet-256^2 with a DiT backbone (Peebles & Xie, 2023) and show that the same approach transfers to text-to-image (T2I).

2 Background

2.1 Flow Matching

Flow matching is a specific instantiation of diffusion models (Lipman et al., 2023; Albergo et al., 2025; Liu et al., 2023b) where we learn a time-dependent velocity field $v_\theta(t, x_t)$ that transports samples from p_0 (noise) to p_1 (data) distributions. Given a data sample $x_1 \sim p_{\text{data}}$ and noise $x_0 \sim \mathcal{N}(0, \mathrm{Id})$, we define the interpolant

$$x_t = (1 - t)\, x_0 + t\, x_1, \quad t \in [0, 1]\,, \tag{1}$$

The model is trained by regressing on the conditional velocity field $u_t(x_t|x_0, x_1) = x_1 - x_0$ via

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, x_0, x_1}\left[\|\hat{v}_\theta(t, x_t) - (x_1 - x_0)\|^2\right]\,. \tag{2}$$

The minimizer of this objective is the conditional mean velocity, i.e. $\hat{v}_{\theta^\star}(t, x_t) \approx \mathbb{E}[x_1 - x_0 \mid x_t]$. At inference, samples are generated by integrating the learned velocity field from $t{=}0$ to $t{=}1$ using

a numerical ODE solver (*e.g.*, Euler). For K steps with uniform spacing $\Delta t = 1/K$, starting from $x_0 \sim \mathcal{N}(0, \mathrm{Id})$, it yields the following iterative procedure.

$$x_{t+\Delta t} = x_t + \Delta t \cdot \hat{v}_{\theta^\star}(t, x_t)\,. \tag{3}$$

2.2 Distributional Diffusion Models

Standard diffusion models learn $\hat{x}_\theta(t, x_t) \approx \mathbb{E}[x_1|x_t]$, the conditional mean of the clean data, connected to the velocity field via $\hat{x}_\theta(t, x_t) = x_t + (1-t)\hat{v}_\theta(t, x_t)$. For sufficiently fine time discretization (large K), the Euler iterates equation 3 with this learned model approximate the target distribution p_{data}; see e.g. (Song et al., 2021). However, for coarse discretization (small K), this guarantee no longer holds.

Distributional Diffusion Models (DDMs) (De Bortoli et al., 2025b) address this by learning a generative model $\hat{x}_\theta(t, x_t, \xi)$ that takes an additional noise input $\xi \sim \mathcal{N}(0, \mathrm{Id})$ and aims to approximate $p(x_1|x_t)$. It is trained by maximizing a *generalized energy score* (De Bortoli et al., 2025b; Gneiting & Raftery, 2007; Bouchacourt et al., 2016) parameterized by $\beta \in (0, 2]$ and $\lambda \in [0, 1]$:

$$S_{\lambda,\beta}(p, y) = -\mathbb{E}_p\Big[\|X - y\|_2^\beta\Big] + \tfrac{\lambda}{2}\,\mathbb{E}_{p\otimes p}\Big[\|X - X'\|_2^\beta\Big]\,, \tag{4}$$

where p is the learned proposal distribution, $y \sim q$ is a sample from a target distribution. In equation 4 $\lambda \in [0, 1]$ controls the relative weight of the interaction term. When $\beta \in (0, 2)$ and $\lambda = 1$, $S_{\lambda,\beta}$ is strictly proper (Gneiting & Raftery, 2007) with the unique maximizer equal to the target distribution, while $\beta = 2, \lambda = 0$ recovers the standard MSE regression objective, targeting only the conditional mean.

The DDM training loss integrates this score over diffusion time, yielding the following objective

$$\mathcal{L}_{\text{DDM}}(\theta) = -\int_0^1 w(t)\,\mathbb{E}_{p(t)}\left[\mathbb{E}_{p(x_0,x_1,x_t)}\left[S_{\lambda,\beta}\big(p^\theta(\cdot \mid x_t),\, x_1\big)\right]\right]\mathrm{d}t\,, \tag{5}$$

where $w(t)$ is a loss weighting function and $p(t)$ is a time distribution (typically uniform). In practice, $p^\theta(\cdot \mid x_t)$ is represented by $\{\hat{x}_\theta(t, x_t, \xi_j)\}_{j=1}^m$ with m samples $\xi_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathrm{Id})$. The loss equation 5 can equivalently be expressed in terms of a velocity field $\hat{v}_\theta(t, x_t, \xi)$ using the relationship

$$\hat{x}_\theta(t, x_t, \xi) = x_t + (1-t)\hat{v}_\theta(t, x_t, \xi) \tag{6}$$

At inference, a single $\xi \sim \mathcal{N}(0, \mathrm{Id})$ is drawn per denoising step. The stochastic prediction $\hat{x}_\theta(t, x_t, \xi)$ induces a stochastic velocity $\hat{v}_\theta(t, x_t, \xi)$ via equation 6, which is then used in an Euler update (Eq. 3). Drawing multiple ξ samples for the same (x_t, t) allows inspecting the learned conditional spread.

2.3 Dynamical Regimes of the Reverse Process

Biroli et al. (2024) study the Ornstein–Uhlenbeck forward diffusion process[1]

$$\mathrm{d}x_\tau = -x_\tau\,\mathrm{d}\tau + \sqrt{2}\,\mathrm{d}W_\tau, \qquad x_0 \sim p_{\text{data}},$$

for which the noised distribution is obtained by convolution with a Gaussian of variance $\Delta_\tau := 1 - \mathrm{e}^{-2\tau}$. Under the exact empirical score assumption, they analyze the corresponding reverse dynamics in the large-dimensional regime and show that it passes through three regimes separated by two characteristic times. The first transition is the *speciation time* τ_s, defined by $\Lambda \mathrm{e}^{-2\tau_s} = 1$. where Λ is the top eigenvalue of the data covariance matrix $C_0 = \mathbb{E}[X_0 X_0^T]$ for $X_0 \sim p_{\text{data}}$ (assuming that it is centered). At this point, coarse structure in the data becomes distinguishable from noise. The second transition is the *collapse time* τ_c, defined by $H(\tau_c) = H^{\text{sep}}(\tau_c)$ where $H(\tau) = -\frac{1}{d}\int \mathrm{d}x\, p_\tau(x) \log p_\tau(x)$ is the per-dimension entropy of p_τ, and $H^{\text{sep}}(\tau) = \frac{\log n}{d} + \frac{1}{2} + \frac{1}{2}\log(2\pi\Delta_\tau)$ is the entropy of n well-separated Gaussians with variance Δ_τ (n : dataset size).

These two transition times define three regimes: before speciation, the reverse process remains essentially undifferentiated; between τ_s and τ_c, the dynamics captures coarse structure without collapsing onto a training example; after collapse, a trajectory is attracted toward a particular training datum. The mapping to our parameterization and the associated schedule construction are in Sections 3.2 and Appendix C.

[1]For consistency with Biroli et al. (2024), we introduce the dynamical regimes using the diffusion model convention but will later use them using the flow matching convention of Section 2.1.

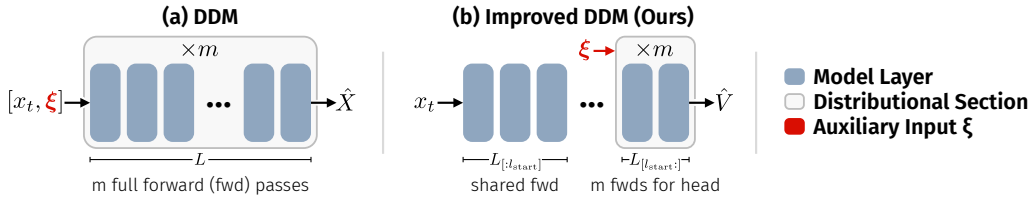

Figure 2: **Deferred Population Expansion.** **(a)** Standard DDM replicates the input m times and processes all particles with the full model ($m\times$ cost), concatenating ξ at the input. **(b)** Our approach processes the shared representation through layers $0, \ldots, \ell_{\text{start}}-1$ once, then replicates and injects ξ conditioning at layer ℓ_{start}. Only the remaining $L - \ell_{\text{start}}$ layers pay the $m\times$ cost.

3 METHOD

We build on DDMs in two orthogonal directions. Section 3.1 reduces the cost of multi-particle training through deferred *population expansion* and the associated ξ conditioning. Section 3.2 modifies the training objective through time-dependent scoring rule schedules.

3.1 EFFICIENT MULTI-PARTICLE TRAINING AND ξ CONDITIONING

Multi-Particle Training. The DDM objective (Eq. 5) requires, for each training example, a population of m predictions $\{\hat{v}_\theta(t, x_t, \xi_j)\}_{j=1}^m$ (Eq. 6) obtained from the same (x_t, t) with independent auxiliary noises $\xi_1, \ldots, \xi_m$. We call this replication *population expansion*. Naïvely, expansion happens at the network input, so the transformer reprocesses the same x_t m times (Fig. 2a), although particle-specific variation enters only through ξ_j. We instead *defer* expansion: for a minibatch of size B, layers $0, \ldots, \ell_{\text{start}} - 1$ run once at batch size B; the hidden state is then replicated, ξ is injected, and layers $\ell_{\text{start}}, \ldots, L - 1$ run at batch size $B \times m$ (Fig. 2b). Unlike shared-trunk architectures with independently parameterized heads, all particles pass through the same remaining blocks and differ only in the injected noise. This reduces training-step compute to $\big(\ell_{\text{start}} + (L - \ell_{\text{start}})m\big)/(Lm)$ of naïve DDM, i.e. $\approx 36\%$ for DiT-XL (L=28, ℓ_{start}=24, m=4); we ablate ℓ_{start} in Section 4.1.

ξ Conditioning. After expansion,we study four mechanisms for injecting ξ_j (Fig. 3). For channel-wise concatenation, we compare widening all late layers, widening only the residual stream at fixed inner width, and overwriting channels; our default, fixed-inner-width concatenation, keeps the model size roughly unchanged. A learned t-dependent gate (Eq. 24, App. E) lets the model control the magnitude of ξ. All choices are ablated in Section 4.1.

3.2 TIME-DEPENDENT SCORING RULE SCHEDULES

De Bortoli et al. (2025b) found that the best (λ, β) (Eq. 5) depends on the sampling budget: models trained with $\lambda \approx 1, \beta \approx 1$ sample best with few steps, whereas $\lambda \approx 0, \beta \approx 2$ is best with many steps. A key insight from DDM is that $\lambda < 1$ causes the model to underestimate the variance of $p(x_1|x_t)$, biasing learning toward the conditional mean. When the posterior is already concentrated (e.g. $t \approx 1$), this bias is beneficial; when it is broad, it is harmful. This motivates making (λ, β) depend on t.

We propose time-dependent schedules $(\lambda(t), \beta(t))$ to allocate the model's finite capacity by linking the training objective to the complexity of $p(x_1 \mid x_t)$ at each phase of the diffusion process, using the dynamical regimes of Biroli et al. (2024) to locate where the objective should transition from distributional toward regression-like behavior. App. D further motivates this orientation through a posterior-covariance bias criterion, which determines the direction of the schedule not its exact shape.

3.2.1 UNDERSTANDING THE POSTERIOR EVOLUTION THROUGH DYNAMICAL REGIMES

As described in Section 2.3, the dynamical regimes framework of Biroli et al. (2024) identifies three phases of the reverse dynamics separated by two characteristic transition times. We use these transition times as anchors for schedule design, linking the dynamical picture to the evolution of the conditional law $p(x_1 \mid x_t)$. In Biroli et al. (2024), we have that $x_\tau = \exp[-\tau]x_{\text{data}} + (1-\exp[-2\tau])^{1/2}z$ whereas in equation 1 we have $x_t = (1-t)z + t x_{\text{data}}$. In order to transfer the learning of Biroli et al. (2024) to our setting we perform the time change of variable $t \mapsto \phi(t)$ such that $\rho(t) = \frac{\exp[-2\phi(t)]}{1-\exp[-2\phi(t)]} = \frac{t^2}{(1-t)^2}$,

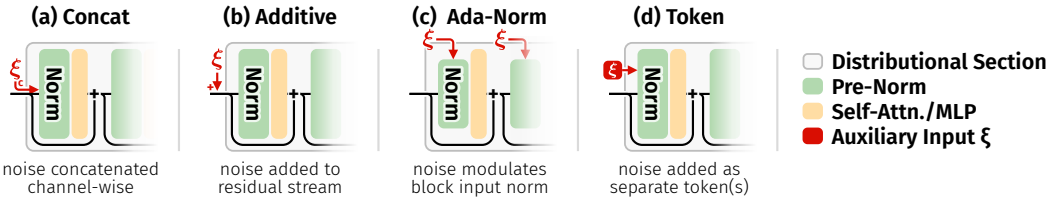

Figure 3: ξ**-conditioning mechanisms.** After expansion, ξ_j is injected by **(a)** channel-wise concatenation, which appends ξ-features to the residual stream, **(b)** residual addition, which adds projected ξ-features to hidden states, **(c)** AdaNorm modulation, which conditions the normalization path, **(d)** or register tokens, which expose ξ through additional tokens.

thereby effectively mapping the Signal-to-Noise Ratio (SNR) of Flow Matching to the one of Biroli et al. (2024). For schedule design, we therefore use the following regime picture:

Regimes I-II ($\rho < \rho_c$): Prior and Manifold Localization. Below speciation ($\rho < \rho_s$), $p(x_1 \mid x_t)$ is close to the data prior; between speciation and collapse it localizes onto a lower-dimensional manifold as a complex, multimodal distribution.

Scheduling Implication (Regime I and II): Favor a more distributional objective.

Regime III ($\rho \geq \rho_c$): Collapse. After collapse, the posterior is mode-dominated but not necessarily concentrated. We refine this regime with a geometric separation threshold $\rho_{\text{sep}}(\varepsilon) = 4\chi^2_{d,1-\varepsilon}/d^2_{\min}$, where $d_{\min}$ is the minimum pairwise distance between training points and $\chi^2_{d,1-\varepsilon}$ the $(1-\varepsilon)$-quantile of a χ^2_d variable: at this SNR, the $(1-\varepsilon)$-quantile noise ball around each noised training point fits within half the distance to its nearest neighbor (derivation in App. C.2). This splits Regime III into an ambiguous window IIIa ($\rho_c \leq \rho < \rho_{\text{sep}}$) and an unambiguous data-end regime IIIb ($\rho \geq \rho_{\text{sep}}$).

Scheduling Implication (Regime IIIa): Maintain spread while tapering to a sharper objective.

Scheduling Implication (Regime IIIb): Regression-dominated objective.

We instantiate time-dependent schedules for the generalized energy score equation 4, varying both the interaction weight $\lambda \in [0, 1]$ and the norm exponent $\beta \in (0, 2]$ through shape functions $s_\lambda(t), s_\beta(t) \in [0, 1]$:

$$\lambda(t) \;=\; \lambda_{\max} \cdot s_\lambda(t), \qquad \beta(t) \;=\; 2 - (2 - \beta_{\min}) \cdot s_\beta(t). \tag{7}$$

The shape functions control how strongly the objective departs from the regression-like limit: $s_\lambda(t) = 0$ removes the interaction term, $s_\beta(t) = 0$ gives $\beta(t) = 2$, and $\beta_{\min} = 1$ gives the standard energy score at the schedule peak. We use five profile families (Appendix D): *constant* (the original DDM), *linear*, *step*, *SNR*, and *dynamical-regimes* (*dyn-reg*), which is piecewise linear in t with $s(t) = 1$ for $t \leq t_s$ and $s(t) = 0$ for $t \geq t_{\text{sep}}$. The regime thresholds interpret the schedule direction but are not required in practice: a simple linear schedule achieves a strong trade-off across sampling budgets without dataset-specific threshold estimation.

4 Experiments

The section is organized as follows. We first ablate our design choices (Tab. 1), then compare against few-step methods in a controlled setting (Tab. 4a) and finally at system level at large scale (Tab. 4b).

Experiment Settings. We train B-scale latent DiTs (depth 12, width 768) (Peebles & Xie, 2023; Vaswani et al., 2017; Dosovitskiy et al., 2021; Rombach et al., 2022) on class-conditional ImageNet-256^2 (Russakovsky et al., 2015) for 400k steps using AdamW (Loshchilov & Hutter, 2019) unless noted otherwise (all hyperparameters in Table 5). Evaluations primarily use FID@50k↓ (Heusel et al., 2017) in many- (50) and few-step (4) settings, with individually swept CFG scales (Ho & Salimans, 2022). We report FD-DINOv2, KID, KDD, Inception Score, precision, and recall in Appendix B.

4.1 Ablation Study

We ablate the design decisions of Section 3 along three axes: (i) architectural choices, including deferred particle expansion (Sec 3.1), t-dependent ξ gating (App E), and noise conditioning variants (Fig 3); (ii) time-dependent scoring rule schedules $(\lambda(t), \beta(t))$ (Sec 3.2); and (iii) the training time distribution $p(t)$ in equation 5. Table 1 adds them step by step, from a DDM (De Bortoli et al.,

Table 1: **Main Ablation.** Starting from DDM (De Bortoli et al., 2025b) baseline (DiT-B, ImageNet-256[2]), we gradually add our proposed components. (CONFIG **F**) achieves the best performance.

Method	Train Cost	FID↓ (by steps)			
		50	20	10	4
– FM Reference	1.0×	4.97	5.64	8.11	26.53
A DDM Baseline (naïve adaptation to transformers)	4.0×	10.04	10.16	11.85	43.68
B + Deferred Expansion	1.5×	5.41	5.71	7.25	17.51
C + t-Adaptive Noise Gating	1.5×	5.50	5.70	7.02	16.02
D + Improved ξ Conditioning	1.5×	5.47	5.62	6.97	15.74
E + Time-dependent Scoring Rule Schedules	1.5×	5.69	5.98	7.11	14.33
F + Improved Train-time t Sampling	1.5×	**4.56**	**4.98**	**6.42**	**13.13**

Table 2: **Architecture Design Space.** Starting from CONFIG **D** (Table 1), we explore optimal hyperparameters for our architectural changes using FID↓. **Best**, 2nd, and final choices are highlighted.

(a) **Expansion Start Layer.** Few-step inference performs best for $\ell_{\text{start}} \in [8, 10]$. We choose $\ell_{\text{start}} = 10$ as it combines good few-step results with low extra cost.

ℓ_{start}	Cost	50-step	4-step	ℓ_{start}	Cost	50-step	4-step
0	4.00×	7.67	26.62	6	2.50×	6.39	20.27
1	3.75×	7.79	24.27	7	2.25×	6.30	16.23
2	3.50×	7.39	22.21	8	2.00×	5.55	15.43
3	3.25×	7.03	24.41	9	1.75×	5.59	**14.96**
4	3.00×	6.99	25.85	10	1.50×	5.47	15.74
5	2.75×	6.61	24.51	11	**1.25×**	**5.41**	18.15

(b) $\boldsymbol{\xi}$ **Conditioning.** Ada-norms/ξ tokens work poorly; fixed concat works best.

Mechanism	50-step	4-step
ada-norm	6.63	97.06
additive	5.49	17.00
tokens	6.38	70.17
concat subst.	5.51	16.02
concat+fixed	5.47	**15.74**
concat+scale[†]	**5.23**	16.29

(c) **Noise Gating.** Learned t-adaptive gating improves few-step results.

Gating	50-step	4-step
static	**5.35**	17.46
t-adaptive	5.47	**15.74**

[†]adds substantial additional trainable parameters (∼10M)

2025b) baseline (CONFIG **A**) to our final method (CONFIG **F**). To disentangle individual contributions, Table 8 adds each independently to CONFIG **B**, and all improve the 4-step result without retuning.

4.1.1 DEFERRED EXPANSION & ARCHITECTURAL IMPROVEMENTS

In Table 1, we gradually add deferred expansion and architectural improvements (CONFIGS **A**→**D**). Then in Table 2 and Fig. 4, we report results starting from CONFIG **D** to account for joint interactions.

Deferred Expansion (CONFIG B). Compared to a FM (Lipman et al., 2023) baseline, DDMs (De Bortoli et al., 2025b) incur a large training cost overhead proportional to the number of particles m [2].

How much should expansion be deferred? In Fig. 4 and Table 2a, we vary the number of distributional layers. Without guidance (Fig. 4a, $w = 1$), less deferral leads to better performance, leading to a direct trade-off between training cost and generation quality. However, in practical settings, CFG (Ho & Salimans, 2022) is typically applied to obtain better generation quality. Once applied, the behavior is reversed: models with *fewer* distributional layers tolerate high guidance scales, enabling *improved performance* under guidance (Fig. 4b). This leads to *simultaneous* reductions in training costs (to $1.5\times$ of FM cost compared to $4\times$ of baseline DDM) and improvements in generation quality.

Timestep-Adaptive Noise Gating (CONFIG C). We study the impact of a learned t-dependent gating mechanism equation 24 which controls the magnitude of noise ξ (see App. E). This slightly reduces many-step performance but boosts few-step generation quality (Tab 1 **B**→**C**; Tab 2c).

Improved $\boldsymbol{\xi}$ Conditioning (CONFIG D). Compared to DDM (De Bortoli et al., 2025b), which concatenates ξ channel-wise to the model input in pixel space, our deferred expansion affords a larger design space, explored in Tab. 2b. Among concatenation variants, full widening expands all late-layer projections, fixed-inner-width widening carries ξ only in the residual stream, and substitution overwrites channels. Fixed-inner-width concatenation gives the best trade-off, so we adopt it. Adaptive norms and token conditioning underperform *severely*, especially with few steps.

[2]The cost of the forward-backward pass scales *linearly* with m while the loss computation scales *quadratically* with m. We only focus on the forward-backward cost since it is the costliest operation in the training loop.

Table 3: **Training Objective Design Space.** We explore the optimal choice of hyperparameters and ablate our training objective improvements, using FID↓. **Best**, 2nd, and final choices are highlighted.

(a) **Initial Exponent.**

$\beta(t=0)$	50-step	4-step
1.00	**5.34**	14.80
0.50	5.43	15.04
0.10	5.69	**14.33**
0.01	5.89	14.42

(b) **Scoring Rule Scheduling.** (i) Linear scheduling gives the best few-step results. Note however that the landscape changes as the number of steps increase. For 50-steps, step combination, with early switch to diffusion-like behavior dominate (ii-iii) We also explore omitting either scheduler. While most of those combinations underperform, we find surprisingly that keeping $\lambda = 1$ and a step schedule for β yields good results.

(c) **Train t Sampling.**

$p(t)$	50-step	4-step
imf	5.18	13.98
jit	**4.57**	**13.13**
logit	5.70	14.58
uniform	6.61	15.39

(d) **Kernel sp. mode.**

Kernel	50-step	4-step
global	5.51	18.36
local	**5.42**	**15.88**

(i) **Schedule** $\beta(t), \lambda(t)$.

Sched. Profile	50-step	4-step
linear	5.69	14.33
dyn-reg	5.73	**14.27**
snr	6.42	14.64
step-0.90	7.94	19.94
step-0.75	12.28	25.16
step-0.50	8.77	20.18
step-0.25	5.42	16.30
step-0.10	**4.91**	16.23

(ii) **Ablation:** only $\beta(t)$, $\lambda = 1$

Sched. Profile	50-step	4-step
linear	6.12	16.90
dyn-reg	6.34	16.66
snr	6.96	17.71
step-0.90	7.95	22.31
step-0.75	12.64	28.58
step-0.50	9.21	18.35
step-0.25	5.61	14.38
step-0.10	**5.10**	**13.06**

(iii) **Ablation:** only $\lambda(t)$, $\beta = 1$

Sched. Profile	50-step	4-step
linear	5.35	14.87
dyn-reg	5.35	**13.60**
snr	**5.24**	14.26
step-0.90	5.46	16.49
step-0.75	5.62	14.07
step-0.50	5.38	14.11
step-0.25	5.28	15.08
step-0.10	5.74	17.31

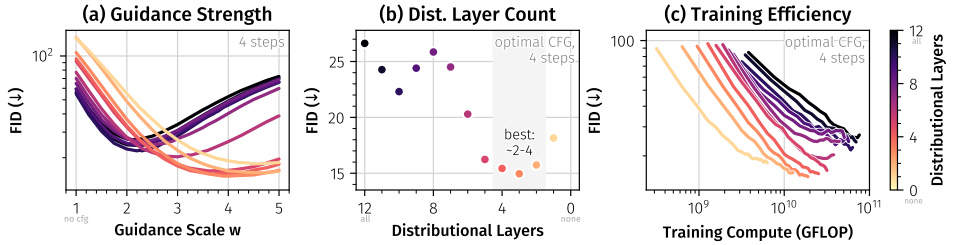

Figure 4: **Few-Step Generation Quality vs. Number of Distributional Layers.** We train a DiT-B on ImageNet for 400k steps and evaluate FID on 50k 4-step samples, optionally using CFG. **(a)** In the unguided setting, more distributional layers improve generation quality. However, this trend *reverses in the practical setting* with CFG, with models with *fewer* distributional layers performing better. **(b)** Under optimal guidance, 2-4 distributional layers lead to to the best results. **(c)** Training efficiency decreases significantly with many distributional layers, but convergence bottoms out early with too few (notably, also for some models with higher distributional layer count). Similarly to the step-matched setting in (b), around 2-5 distributional layers offer a good efficiency-quality trade-off.

4.1.2 Training Objective Improvements

We explore training improvements: time-dependent schedules $(\lambda(t), \beta(t))$ (Section 3.2), the training time distribution $p(t)$, and kernel spatial modes (see Appendix F). The results are in Table 3.

Time-dependent Scoring Rule Schedulers (Config E). Following section 3.2, we use time-dependent schedules for $(\lambda(t), \beta(t))$ – hyperparameters of the generalized energy score equation 4. The results for many options of different schedules are given in Table 3b. The linear schedule overall leads to the best trade off in performance for 50 and 4 steps regime. The rest of the results in Table 3 are using this linear schedule.

Do both the norm exponent β and the interaction weight λ require scheduling? We also explore omitting the scheduler for either β or λ in Table 3b-ii and Table 3b-iii. While most combinations underperform, scheduling only β with $\lambda = 1$ and an early switch at $t = 0.1$ yields particularly strong FID. We do not adopt this setting as our default because at $(\lambda, \beta) = (1, 2)$ the score reduces to $\|\mu - y\|^2$, independent of the population covariance, so the spread of the model's samples is left unconstrained for $t > 0.1$. Under our schedules, where $\lambda < 1$, the variance remains identified. A theoretically optimal choice of $\lambda(t)$ and $\beta(t)$ remains an open question.

How to set the initial norm exponent $\beta(t = 0)$? The schedule for $\beta(t)$ is an increasing function of t with the lowest value at $\beta(t = 0)$. In Table 3a we run the ablation over this value and find that a value of 0.1 yields the best trade-off between different sampling budgets.

Improved Train-time t Sampling (Config F). Finally, Table 3c compares different distributions from which to sample t during training. Since we apply no further t-dependent loss weighting, these control both the implicit time-dependent weighting of the objective and the distribution of

interpolants x_t seen during training. We observe that `jit` distribution leads to the best results. This noise distribution biases the time towards the initial noise, where distributional losses matter more.

For compactness, we report FID only in the main text. The improvements persist across all additional metrics, including FD-DINOv2, KID, KDD, Inception Score, precision, and recall and are reported in Table 9. Repeated sampling from fixed x_t further confirms that iDDM produces conditional diversity that deterministic FM cannot (Table 11). Accounting for iDDM's higher per-step training cost, a compute-matched comparison shows the same few-step advantage: at the training budget of the 400k-step FM baseline, iDDM (260k steps) improves 4-step FID from 26.70 to 14.33, while FM remains slightly better at 50 steps (4.97 vs. 5.62); see Table 10 for the full compute sweep.

4.2 Comparison with Previous Methods

Controlled Comparison on ImageNet-256[2]. Existing few-step methods (Zhou et al., 2025a; Geng et al., 2025; 2026; Potaptchik et al., 2026) typically report results in highly varying settings, making fair comparisons challenging. We therefore compare with them in a controlled setting, retraining all models from scratch at B-scale with comparable settings. We re-sweep relevant hyperparameters to ensure a fair comparison. As shown in Table 4a, iDDM substantially improves over naïve DDM, reducing 4-step FID from 43.68 to 13.13 and 50-step FID from 10.04 to 4.57, and the deterministic baselines, while remaining competitive with stochastic flow-map methods across sampling budgets.

System-Level Results. At XL scale, iDDM-XL/2 reaches 2.38 FID at 50 steps and 4.48 at 4 steps after 200 epochs (samples in Fig. 1). Table 4b compares iDDM against deterministic few-step flow-map methods trained from scratch (Frans et al., 2025; Geng et al., 2025; 2026; Zhou et al., 2025a) and against MFM (Potaptchik et al., 2026), which uses multi-stage training from a pretrained model. Entries marked * are taken from the respective papers, all others are evaluated by us with official checkpoints and code. Both DDM and iDDM are trained from scratch.

Positioning w.r.t. MeanFlow-style and few-step methods. At 1-2 steps, teacher-based distillation such as sCD (Lu & Song, 2025) and ADD (Sauer et al., 2025) achieves the lowest FID, but targets a different operating point: it requires a pretrained many-step teacher and a second training stage while iDDM doesn't. Among methods trained from scratch, published iMF and IMM checkpoints attain lower FID than iDDM at every inference budget, at a substantially higher training cost: iMF uses a 48-layer model and roughly $17.6\times$ our total training budget, and IMM $19\times$ more training epochs. iDDM instead combines three properties that no baseline in Table 4 offers together. (i) It is *stochastic*: it learns a conditional predictor trained to sample from $p(x_1 \mid x_t)$, whereas MeanFlow, iMF, and IMM learn deterministic maps. (ii) It is trained from random initialization in a *single stage*, whereas the published MFM results rely on a pretrained flow map and multi-stage training. (iii) A *single checkpoint* serves the full 4-50 step range without degrading: its FID is non-increasing in the number of steps ($4.48 \to 2.38$), whereas MeanFlow ($2.93 \to 3.29$) and MFM ($1.97 \to 5.74$) are stronger at 4-8 steps but get worse with additional sampling compute. iDDM is also relatively inexpensive to train: per training step it costs $1.41\times$ flow matching, versus $1.96\times$ for MeanFlow and $6.20\times$ for iMF (Table 12). Finally, against flow matching at matched training compute (Table 10), iDDM halves 4-step FID (13.13 vs. 26.36) while staying within 0.40 FID at 50 steps (4.61 vs. 4.21).

Transfer to T2I. We transfer iDDM to a 1.6B-parameter text-to-image model. Relative to a matched FM baseline, iDDM degrades far less as the budget decreases, improving MS-COCO FID from 78.20 to 41.05 at 4 steps and from 26.33 to 19.40 at 8 steps (App. B.2).

5 Related Work

Accelerating diffusion sampling. Distillation (Luhman & Luhman, 2021; Sauer et al., 2025; Yin et al., 2024; Salimans & Ho, 2022; Liu et al., 2024; Song et al., 2023; Xu et al., 2024; Salimans et al., 2024) trains a few-step student against a many-step teacher; improved samplers (Lu et al., 2022; 2025; Zheng et al., 2023; Karras et al., 2022) reduce discretization error of a frozen model; parallel and speculative methods (Shih et al., 2023; Chen et al., 2024; De Bortoli et al., 2025a) address the sequential nature of diffusion models. Our work is complementary to all three: we modify the training objective, not the sampler. For instance, our method integrates with the Universal Inverse Distillation framework of Kornilov et al. (2025) by swapping the diffusion loss for a distributional one.

Table 4: **Few-Step Comparison on ImageNet-256²** (FID↓). iDDM *significantly* improves upon DDM and achieves competitive FID across all steps while training from scratch in a single stage.

(a) **Controlled Comparison** at DiT-B scale (depth 12, width 768), re-trained from scratch under same standard settings (400k steps/80ep).

Method	50-step	4-step
Deterministic Methods		
Shortcut	9.85[§]	15.02
IMM	7.57	24.85
Stochastic Methods		
MFM	7.72	**9.17**
DDM (naïve)	10.04	43.68
iDDM (ours)	**4.57**	13.13

(b) **System-level Comparison** at DiT-XL scale (depth 28, width 1152). [†]iMF uses a much deeper transformer (28→48 layers).

Method	#Epochs	50-step	16-step	8-step	4-step
Deterministic Few-step Flow-map models from scratch					
Shortcut-XL/2	162	4.68[§]	5.39	5.74	7.8 *
MeanFlow-XL/2	240	3.29	3.06	2.93	2.93
iMF-XL/2[†]	800	1.43	1.43	1.50	1.51
IMM-XL/2	3840	1.82	1.86	1.99*	2.51*
Stochastic Few-step Flow-map (multi-stage training)					
MFM-XL/2	800+80+28	5.74	3.29	2.19	1.97*
Stochastic Few-step diffusion/FM from scratch					
DDM-XL/2 (naïve)	200	4.71	4.50	5.70	23.25
iDDM-XL/2 (ours)	200	2.38	3.03	3.04	4.48

[§]evaluated at 64 sampling steps since Shortcut only supports powers of two.

Learning the diffusion posterior. Several approaches enrich the standard mean-prediction denoiser with (co)variance information for few-step sampling (Ho et al., 2020; Nichol & Dhariwal, 2021; Bao et al., 2022a;b; Ou et al., 2025). Diffusion-GAN (Xiao et al., 2022) and UFOGen (Xu et al., 2024) learn multimodal conditional generators via adversarial training conditioned on a noise variable ξ—architecturally close to DDMs but trained against a discriminator instead of a proper scoring rule. Aiello et al. (2024) and Galashov et al. (2025) use MMD on CLIP features and MMD gradient flows respectively; Wang & He (2025) regularises representations for diversity without explicit distributional training. We build directly on the generalized energy score formulation of (De Bortoli et al., 2025b; Shen et al., 2025), addressing its training cost and its static hyperparameters. Very recent works also investigate the use of *distributional losses* in representation space Yang et al. (2026). Extended related work on the distributional lineage of our method can be found in Section H.

Consistency, moment-matching, and fastforward methods. Consistency Models (Song et al., 2023; Song & Dhariwal, 2024) and extensions like Consistency Trajectory Models (Kim et al., 2024) and Shortcut Models (Frans et al., 2025) learn self-consistent few-step generators. MeanFlow (Geng et al., 2025) learns an average velocity field for one-step generation, and the improved iMF variant addresses training stability and recovers inference flexibility (Geng et al., 2026). AlphaFlow (Zhang et al., 2026) interpolates between flow matching and MeanFlow via a schedule. Our method takes a different approach and share the multi-particle philosophy with IMM (Zhou et al., 2025a) and our focus remains only on the few-step regime (4–50).

Meta Flow Maps and stochastic flow maps. MFM (Potaptchik et al., 2026) amortises stochastic flow maps producing i.i.d. samples from $p_{1|t}(\cdot \mid x_t)$ via diagonal flow matching and consistency losses. MFM and DDM target the same object ($p_{1|t}$) but differ in the training principle — consistency along auxiliary ODEs versus a proper scoring rule on populations. The emphasis of Potaptchik et al. (2026) is reward alignment through efficient value-function estimation from stochastic endpoint samples, whereas our focus is multi-step generation under direct scoring rule training. Our experiments therefore address a different operating regime with late expansion and time-dependent schedules.

6 Conclusion

We presented a method that makes efficient stochastic few-step generation with DDMs feasible at DiT scale via two complementary contributions. First, deferred population expansion enables efficient multi-particle training: it shares the early transformer layers across particles and injects particle-specific noise only in a late distributional section, reducing the cost of $m = 4$ DDM training from roughly $4\times$ flow matching to about $1.5\times$. Second, time-dependent scoring rule schedules and improved training-time sampling provide time-adaptive distributional objectives, improving few-step performance. Overall, our results suggest that shared computation, carefully placed stochasticity, and adaptive objectives make DDMs practical for stochastic few-step generation from scratch.

Limitations. The regime anchors are derived from a mean-field analysis and are approximate at finite scale. Smooth schedules are more robust than hard steps. At 1-2 steps, distilled and fastforward methods remain stronger. Our gains hold from ImageNet-256^2 and transfer to T2I.

AI USE STATEMENT

In this work, we used generative AI tools for designing and providing feedback on methodology and experiments, as well as assistance implementing methods. We have not used generative AI tools for help developing theoretical or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, assistance with translation, cleaning and reformatting datasets, support in qualitative and thematic data analysis, and interpretation of results and synthetic data generation, as well as assistance in the writing of proofs, are not applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures, creating and editing software code, summarizing existing literature, brainstorming, sourcing/searching for information, editing the research paper for readability, and the identification of relevant literature. We have reviewed all AI-assisted work. Methodology and experiments as well as brainstorming results were discussed among at least two researchers, LLM-generated code was verified and tested for correctness, LLM-assisted literature search was followed up by manual literature review as well as verification of correct summaries by at least two researchers. Finally, AI-generated figures were manually verified and improved. Figures were discussed with at least two researchers to ensure correctness and understandability. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

iDDM is trained using publicly available datasets. We detail data processing and training in Appendix. iDDM is a simple method that works with standard transformers requiring only minimal modifications. Additionally, we detail hyperparameter settings in the Appendix A. We build upon existing evaluation pipelines from and and detail evaluation protocols and metrics in Appendix B.1. Therefore, iDDM is easy to reproduce from scratch. However, we will additionally release code, model checkpoints, and evaluation pipelines upon acceptance.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024.

Emanuele Aiello, Diego Valsesia, and Enrico Magli. Fast inference in denoising diffusion models via mmd finetuning. *IEEE Access*, 12:106912–106923, 2024.

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1555–1584. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/bao22d.html.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=0xiJLKH-ufZ.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, Nov 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54281-3. URL https://doi.org/10.1038/s41467-024-54281-3.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Di5apl8HSH.

Diane Bouchacourt, Pawan K Mudigonda, and Sebastian Nowozin. Disco nets : Dissimilarity coefficients networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/c0e190d8267e36708f955d7ab048990d-Paper.pdf.

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022.

Haoxuan Chen, Yinuo Ren, Lexing Ying, and Grant M. Rotskoff. Accelerating diffusion models with parallel sampling: Inference at sub-linear time complexity. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=F9NDzHQtOl.

Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=WRIn2HmtBS.

Valentin De Bortoli, Alexandre Galashov, Arthur Gretton, and Arnaud Doucet. Accelerated diffusion models via speculative sampling. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=BTJSkPpH1t.

Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Patrick Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=N82967FcVK.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

Sander Dieleman. Learning the integral of a diffusion model, 2026. URL https://sander.ai/2026/05/06/flow-maps.html.

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. URL https://openreview.net/forum?id=YicbFdNTTy.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OlzB6LnXcS.

Alexandre Galashov, Valentin De Bortoli, and Arthur Gretton. Deep MMD gradient flow without adversarial training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Pf85K2wtz8.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uWj4s7rMnR.

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.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

Jiatao Gu, Tianrong Chen, David Berthelot, Huangjie Zheng, Yuyang Wang, Ruixiang Zhang, Laurent Dinh, Miguel Angel Bautista, Josh Susskind, and Shuangfei Zhai. Starflow: Scaling latent normalizing flows for high-resolution image synthesis. *NeurIPS*, 2025.

Jiaqi Han, Puheng Li, Qiushan Guo, Renyuan Xu, Stefano Ermon, and Emmanuel J. Candès. One-step generative modeling via wasserstein gradient flows, 2026. URL https://arxiv.org/abs/2605.11755.

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

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Jonathan Ho, Tim Salimans, Alexey A. Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=f3zNgKga_ep.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24174–24184. IEEE, 2024.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ymjI8feDTD.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Nikita Kornilov, David Li, Tikhon Mavrin, Aleksei Leonov, Nikita Gushchin, Evgeny Burnaev, Iaroslav Koshelev, and Alexander Korotin. Universal inverse distillation for matching models with real-data supervision (no gans). *arXiv preprint arXiv:2509.22459*, 2025.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

Kyungmin Lee, Sihyun Yu, and Jinwoo Shin. Decoupled meanflow: Turning flow models into flow maps for accelerated sampling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=P4VsQexEPe.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning with latent diffusion transformers, 2025. URL https://arxiv.org/abs/2504.10483.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise, 2025. URL https://arxiv.org/abs/2511.13720.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International conference on machine learning*, pp. 1718–1727. PMLR, 2015.

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

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 21450–21474. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/liu23f.html.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023b.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. InstaFlow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1k4yZbbDqX.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12009–12019, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LyJi5ugyJx.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=2uAaGwlP_V.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22(4):730–751, June 2025. ISSN 2731-5398. doi: 10.1007/s11633-025-1562-4. URL http://dx.doi.org/10.1007/s11633-025-1562-4.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

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 *Forty-first International Conference on Machine Learning*, 2024.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Zijing Ou, Mingtian Zhang, Andi Zhang, Tim Z. Xiao, Yingzhen Li, and David Barber. Improving probabilistic diffusion models with optimal diagonal covariance matching. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fV0t65OBUu.

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

Yansong Peng, Kai Zhu, Yu Liu, Pingyu Wu, Hebei Li, Xiaoyan Sun, and Feng Wu. FACM: Flow-anchored consistency models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=k9BpW1c4in.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2025. URL https://arxiv.org/abs/2410.13720.

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael S. Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment, 2026. URL https://arxiv.org/abs/2601.14430.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://github.com/CompVis/latent-diffusionhttps://arxiv.org/abs/2112.10752.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Tim Salimans, Thomas Mensink, Jonathan Heek, and Emiel Hoogeboom. Multistep distillation of diffusion models via moment matching. *Advances in Neural Information Processing Systems*, 37: 36046–36070, 2024.

Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393379. doi: 10.1145/3528233.3530738. URL https://doi.org/10.1145/3528233.3530738.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 87–103, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73016-0.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Xinwei Shen, Nicolai Meinshausen, and Tong Zhang. Reverse markov learning: Multi-step generative models for complex distributions. *arXiv preprint arXiv:2502.13747*, 2025.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bpzwUfX1UP.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

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. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/song23a.html.

George Stein, Jesse Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L Caterini, Eric Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. *Advances in Neural Information Processing Systems*, 36:3732–3784, 2023.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Peng Sun, Zhenglin Cheng, Deyuan Liu, Jun Xie, Xinyi Shang, and Tao Lin. Three-body scattering for generative modeling, 2026. URL https://arxiv.org/abs/2607.18198.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Runqian Wang and Kaiming He. Diffuse and disperse: Image generation with representation regularization, 2025. URL https://arxiv.org/abs/2506.09027.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. 2023.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the Generative Learning Trilemma with Denoising Diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022.

Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You Forward Once Large Scale Text-to-Image Generation via Diffusion GANs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jiawei Yang, Zhengyang Geng, Xuan Ju, Yonglong Tian, and Yue Wang. Representation fréchet loss for visual generation, 2026. URL https://arxiv.org/abs/2604.28190.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712. IEEE, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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 (CVPR)*, pp. 6613–6623, June 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

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.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. DPM-solver-v3: Improved diffusion ODE solver with empirical model statistics. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=9fWKExmKa0.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=pwNSUo7yUb.

Linqi Zhou, Mathias Parger, Ayaan Haque, and Jiaming Song. Terminal velocity matching, 2025b. URL https://arxiv.org/abs/2511.19797.

A Model Configuration

Table 5: Training configuration for latent-space distributional diffusion on ImageNet 256×256. DiT-B/2 and DiT-XL/2 use the same recipe except where model scale requires architecture-specific values.

Config	DiT-B/2	DiT-XL/2
Data and autoencoder		
Dataset	ImageNet-256^2 (Russakovsky et al., 2015)	
Preprocessing	resize, center crop, random horizontal flip	
Autoencoder	REPA-E (SD-VAE, f8d4), $32 \times 32 \times 4$ (Leng et al., 2025)	
Architecture		
Depth L	12	28
Hidden dim	768	1152
Head dim	64	72
Parameters	105M	515M
Patch size	2×2	
Positional encoding	Axial RoPE 2D (Su et al., 2024; Crowson et al., 2024)	
Normalization	Adaptive RMSNorm	
FFN	SwiGLU ($d_{\text{ff}} = 3d$)	
Attention scaling	Cosine similarity, learnable per head (Liu et al., 2022; Crowson et al., 2024)	
Conditioning		
Class dropout	0.1	
ξ conditioning	Concat. fixed Attn./FFN width	
ξ dim	$d_{\text{cat}} = 256$ per-token	
ℓ_{start}	10	24
Distributional objective		
Population size m	4	
$\lambda(t)$ & $\beta(t)$ schedule	linear	
Loss weighting	uniform	
Kernel spatial mode	local	
Training		
Training steps	400k	1M
Global batch size	256	
Optimizer	AdamW (Loshchilov & Hutter, 2019), $(\beta_1, \beta_2) = (0.9, 0.95)$	
Dropout	0.0	
Learning rate	1×10^{-4}	
LR schedule	constant, 300-step warmup	
Weight decay	0	
Gradient clipping	1.0	
Precision	bfloat16 AMP	
EMA	EDM-2 EMA (Karras et al., 2024), $\sigma_{\text{rel}} \in \{0.05, 0.10\}$	
t sampler	logitnorm($\mu = -0.8, \sigma = 0.8$)	
Sampling and evaluation		
ODE solver	Euler	
steps grid	$\{4, 8, 16, 50\}$	

Table 6: Training and evaluation configuration for FM and iDDM. Block indices are zero-based.

Configuration	FM	iDDM
Data and architecture		
Dataset	Recaptioned COYO (Byeon et al., 2022) subset	
Resolution	256×256	
Autoencoder	FLUX.2 (Black Forest Labs, 2025), frozen	
Text encoder	Qwen3-VL-Embedding-2B (Li et al., 2026), frozen language tower	
Text adapter	2 layers, output dimension 2048	
Maximum text length	256 tokens	
Depth / base width	28 / 1536	
Patch size / head dimension	2×2 / 96	
Objective and conditioning		
Text dropout	0.1	
ξ conditioning	–	Concat. fixed Attn./FFN width
ξ dim	–	$d_{\text{cat}} = 384$ per-token
ℓ_{start}	–	20
Width in blocks 20–25	1536	1920
Population size	–	4
Objective	Velocity MSE	Generalized energy-kernel score
$\lambda(t)$ / $\beta(t)$	–	linear, $\beta_{\text{start}} = 0.1$
Kernel spatial mode	–	local
Loss weighting	uniform	
Training		
Evaluated checkpoint	300k steps	
Global batch size	1024	
Optimizer	AdamW (Loshchilov & Hutter, 2019), $(\beta_1, \beta_2) = (0.9, 0.95)$	
Dropout	0.0	
Learning rate	1×10^{-4}	
LR schedule	constant, 300-step warmup	
Weight decay	1×10^{-3}	
Gradient clipping	1.0	
Precision	bfloat16 AMP	
Evaluation EMA	EDM-2 EMA (Karras et al., 2024), $\sigma_{\text{rel}} \in \{0.05, 0.10\}$	
t sampler	logitnorm($\mu = 0, \sigma = 1$)	
Sampling and evaluation		
ODE Solver	Euler	
steps grid	$\{4, 8, 16\}$	
CFG search grid	$\{3, 3.5, \ldots, 7\}$	
Benchmarks	MS-COCO FID, CLIPScore, GenEval, T2I-CompBench; HPS v2.1	

B Extended Quantitative Evaluations

B.1 ImageNet 256^2

All FID scores are computed using 50,000 generated samples compared against the full ImageNet training set statistics, following the ADM evaluation protocol (Dhariwal & Nichol, 2021). We use the same Inception-V3 features and reference statistics as prior work to ensure comparability. For each model (including methods from other papers), we sweep CFG scales in $[1.0, 6.0]$ at 0.1 intervals and report the best. Posthoc EMA (Karras et al., 2024) with $\sigma_{\text{rel}} = 0.05$ is used unless otherwise noted.

For completeness, Table 7 provides broader system-level context on ImageNet-256^2 across heterogeneous training paradigms. As they differ in architecture, parameter count, and training data pipeline, they should be read as context rather than a controlled comparison.

Table 7: **Reference Methods on ImageNet-256^2.** FID↓ of representative methods from other model families, as reported in the respective publications.

Method	Parameters	Steps	FID↓
GANs			
StyleGAN-XL (Sauer et al., 2022)	166M	1	2.30
Autoregressive/Masking			
STARFlow (Gu et al., 2025)	1.4B	–	3.80
VAR-*d*30 (Tian et al., 2024)	2B	10	1.92
Many-step diffusion/FM			
ADM-G (Dhariwal & Nichol, 2021)	554M	250	4.59
DiT-XL/2 (Peebles & Xie, 2023)	675M	250	2.27
SiT-XL/2 (Ma et al., 2024)	675M	250	2.06
LightningDiT-XL/2 (Yao et al., 2025)	675M	250	1.35
RAE + DiT^{DH}-XL (Zheng et al., 2025)	839M	50	1.13
Consistency Models			
iCT-XL/2 (Song & Dhariwal, 2024) via (Zhou et al., 2025a)	675M	2	20.3
FACM (Peng et al., 2026)	675M	2	1.32
Flow Maps			
TVM-XL/2 (Zhou et al., 2025b)	678M	4	1.99
α-Flow-XL/2+ (Zhang et al., 2026)	676M	2	1.95
DMF-XL/2 (Lee et al., 2026)	675M	4	1.51
Direct Distribution Matching			
Drifting Model, L/2 (Deng et al., 2026)	463M	1	1.54
W-Flow, XL/2 (Han et al., 2026)	679M	1	1.29
TBSM-XL/2 (Sun et al., 2026)	675M	1	1.63

Additional Metrics. In addition to FID, we report FD-DINOv2, precision, and recall. FD-DINOv2 follows Stein et al. (Stein et al., 2023) and computes the Fréchet distance in DINOv2-ViT-L/14 feature space. Precision and recall follow Kynkäänniemi et al. (Kynkäänniemi et al., 2019), using k-nearest-neighbor estimates of the real and generated manifolds. Precision reflects sample fidelity, while recall reflects distributional coverage, with higher values being better for both. KID follows Bińkowski et al. (Bińkowski et al., 2018) and computes the squared maximum mean discrepancy between Inception-V3 features of real and generated images using a polynomial kernel. KDD computes the analogous kernel distance in DINOv2-ViT-L/14 feature space. Inception Score follows Salimans et al. (Salimans et al., 2016) and measures the KL divergence between the conditional class distribution predicted by an Inception-V3 classifier and its marginal over the generated set. Precision and recall follow Kynkäänniemi et al. (Kynkäänniemi et al., 2019), using k-nearest-neighbor estimates of the real and generated manifolds. Precision reflects sample fidelity, while recall reflects distributional coverage, with higher values being better for both.

Additional Ablations. To verify that the gains in Table 1 are not an artifact of the A→F ordering, Table 8 adds each refinement independently to Config B rather than cumulatively. All three components improve 4-step FID in isolation (17.51 → 17.21 for ξ-conditioning, →17.34 for t-sampling, →15.77 for the (λ, β) schedule), confirming that none of the gains in Table 1 depends on the presence of the other components. Improved ξ-conditioning is the most uniformly beneficial

change, improving FID, FD-DINOv2, KID, KDD, and precision at both step counts, at the cost of a small reduction in 4-step recall (0.154 → 0.139).

Table 9 extends the main FM/DDM/iDDM comparison of Table 1 with the same additional metrics. At 4 steps, iDDM improves over every metric relative to both baselines simultaneously: FID (43.6→13.2 vs. DDM), FD-DINOv2 (801.8→318.8), KID (0.0337→0.0030), KDD (1.854→0.335), Inception Score (40.3→162.5), precision (0.704→0.778), and recall (0.164→0.226) – ruling out a fidelity/diversity trade-off as the source of iDDM's FID gain over naïve DDM. iDDM also matches or exceeds the FM reference on every metric at both 4 and 50 steps at matched training steps, including precision and recall, which are otherwise close to identical between the two at 50 steps (0.834 vs. 0.833 precision, 0.592 vs. 0.590 recall). Naïve DDM's precision remains close to FM's at both step counts (and is marginally higher at 50 NFE), indicating that its poor FID, FD-DINOv2, and Inception Score at 4 steps reflect a broader degradation in sample quality rather than a precision-specific failure.

Table 8: **Independent Component Ablations.** We independently add each proposed component to CONFIG B and evaluate generation quality across multiple metrics.

NFE	Configuration	FID ↓	FD-DINOv2 ↓	KID ↓	KDD ↓	IS ↑	Prec. ↑	Rec. ↑
4	CONFIG **B**	17.510	496.437	0.00416	0.7158	128.48	0.757	0.154
	B + Improved ξ-conditioning	17.214	464.758	0.00403	0.6272	128.85	**0.773**	0.139
	B + Improved t-sampling	17.338	528.844	0.00418	0.8124	**134.44**	0.763	0.153
	B + (λ, β) schedule	**15.770**	**353.751**	**0.00384**	**0.3968**	130.25	0.738	**0.226**
50	CONFIG **B**	5.447	204.023	0.00130	0.2354	142.84	0.840	0.568
	B + Improved ξ-conditioning	5.356	195.764	0.00123	0.2211	145.51	**0.847**	0.563
	B + Improved t-sampling	**5.141**	225.909	**0.00108**	0.2813	**161.42**	0.836	**0.576**
	B + (λ, β) schedule	5.851	**168.054**	0.00158	**0.1660**	143.15	0.837	0.562

Table 9: **Additional Evaluation Metrics.** We add additional metrics commonly used for evaluation on ImageNet-256 for our final models.

NFE	Method	FID ↓	FD-DINOv2 ↓	KID ↓	KDD ↓	IS ↑	Prec. ↑	Rec. ↑
4	FM	26.699	418.711	0.00870	0.5153	99.10	0.712	0.116
	DDM	43.637	801.812	0.03367	1.8544	40.33	0.704	0.164
	iDDM	**13.224**	**318.794**	**0.00300**	**0.3348**	**162.45**	**0.778**	**0.226**
50	FM	4.970	175.700	0.00121	0.1869	151.30	0.833	0.590
	DDM	10.094	244.954	0.00300	0.2768	111.77	**0.845**	0.487
	iDDM	**4.614**	**173.983**	**0.00109**	**0.1833**	**159.43**	0.834	**0.592**

Matched Compute Comparison with Flow Matching We report iDDM at fractions of its full training run, with the FM baseline extended correspondingly to 600k train steps, so both are compared at matched compute budget rather than matched update count. Training compute is normalized to FM's full budget (400k steps); per-step training cost is 1× (FM), 1.5× (iDDM), 4× (DDM). Step counts are chosen so FM and iDDM are compute-matched exactly (up to checkpoint granularity of 10k steps). Where exact matching is off-grid, iDDM and DDM checkpoints are *rounded down* to the nearest 10k steps, *i.e.*, they receive at most equal training compute. † marks each method's full training run as reported in the paper. The CFG scale is selected independently per method and per inference-step budget with respect to FID; FD-DINOv2 is reported at that same CFG. All other evaluation settings (number of samples, reference statistics, sampler) are identical across all cells.

Posterior Diversity Direct posterior metrics (*e.g.*, MMD or mean and variance errors against $p(x_1 \mid x_t)$) require reference samples from the posterior, which are unavailable at this scale, and importance-weighting the training set degenerates in the 4096-dimensional latent space. We therefore measure how much a model's samples vary for a fixed x_t, and whether that variation depends on x_t. For each of the 1,000 ImageNet classes, we noise one training image to a fixed t, draw 50 samples from each resulting x_t with 8 Euler steps from t to 1, and compute the FID of the 50,000 pooled samples against the standard full-dataset reference statistics (DiT-B; Table 11).

Table 10: Compute-matched comparison for 4-step and 50-step inference. † denotes the reference compute settings.

(a) 4-step inference

	Training Steps			FID ↓			FD-DINOv2 ↓		
Compute	FM	iDDM	DDM	FM	iDDM	DDM	FM	iDDM	DDM
0.60×	240k	160k	60k	29.04	16.94	67.85	462.63	416.87	1091.38
0.75×	300k	200k	70k	27.07	15.80	64.01	437.26	382.58	1060.85
0.90×	360k	240k	90k	27.85	14.43	63.05	434.98	362.17	1021.61
1.00×†	400k	260k	100k	26.70	14.33	59.91	418.71	355.91	999.49
1.20×	480k	320k	120k	27.70	13.69	53.09	436.41	337.14	927.10
1.35×	540k	360k	130k	27.00	13.25	55.47	413.03	326.29	945.77
1.50×†	600k	400k	150k	26.36	13.13	52.63	406.10	318.79	915.72

(b) 50-step inference

	Training Steps			FID ↓			FD-DINOv2 ↓		
Compute	FM	iDDM	DDM	FM	iDDM	DDM	FM	iDDM	DDM
0.60×	240k	160k	60k	6.35	7.13	32.92	195.51	223.20	535.76
0.75×	300k	200k	70k	5.64	6.42	30.25	186.62	208.76	492.68
0.90×	360k	240k	90k	5.32	5.79	26.03	171.86	202.33	438.58
1.00×†	400k	260k	100k	4.97	5.62	23.60	175.70	182.12	421.34
1.20×	480k	320k	120k	4.63	5.07	20.74	177.23	176.69	371.56
1.35×	540k	360k	130k	4.41	4.81	19.35	169.73	181.21	362.33
1.50×†	600k	400k	150k	4.21	4.61	17.76	163.87	173.98	331.29

A deterministic predictor produces only 1,000 distinct images, so its FID stays at this limit for every t, FM indeed shows this with FID≈ 38 independent of t. A predictor that ignored x_t would also be flat in t, but low. A sampler that follows the posterior must instead be diverse at low t and approach the 1,000-image limit as the posterior concentrates, so its FID increases with t. Both distributional models show this profile. iDDM is more diverse than DDM at low t and less at high t, consistent with its schedule, which moves the objective toward regression as the posterior concentrates. This diagnostic detects missing or unconditioned diversity, it does not measure posterior accuracy.

Table 11: FID@50k with samples drawn starting from 1,000 fixed x_t (DiT-B, 8 steps).

Method	t=0.2	t=0.4	t=0.6	t=0.8
FM (deterministic)	37.90	37.70	37.81	38.75
DDM	17.31	16.54	17.64	21.07
iDDM	13.77	16.12	18.96	23.88

Training Compute Comparison To fairly compare training compute, we reimplemented iDDM in JAX inside the MeanFlow/iMF repository and ran a performance benchmark on $4\times$A100-80GB with local batch 32 / global 128 and DiT-XL/2 architecture. As can be seen in Table 12, iDDM is the cheapest of the three alternatives per step. Combined with the reported training lengths (240 epochs for MeanFlow-XL/2 and 800 for iMF-XL/2 against 200 for iDDM-XL/2) MeanFlow's total training budget is roughly $1.67\times$ ours and iMF's roughly $17.6\times$ ours.

B.2 Text-to-Image

Architecture. Our backbone consists of 28 transformer blocks with hidden dimension 1536 and head dimension 96. We use adaptive RMSNorm (Zhang & Sennrich, 2019), SwiGLU feedforward layers (Shazeer, 2020) with expansion factor 3, and axial rotary position embeddings (Su et al., 2024; Crowson et al., 2024). Attention uses cosine similarity with learnable per-head scaling (Liu et al., 2022).

Table 12: Training throughput benchmarked on 4×A100-80GB, local batch 32 / global 128, DiT-XL/2

Method	it/s	img/s	ms/it	× slower vs. FM
FM	2.79	357	358.1	1.00×
iDDM	1.98	254	504.2	1.41×
MeanFlow	1.42	182	705.1	1.96×
iMF	0.45	57	2228.2	6.20×



Figure 5: Example images generated by our text-to-image model at 512^2 resolution. For this, we fine-tune our model pretrained on 256^2 resolution for 20k training steps.

Model. We encode images using the frozen FLUX.2 autoencoder (Black Forest Labs, 2025), producing $32 \times 32 \times 32$ latents at 256×256 resolution, and use 2×2 latent patches. Text conditioning uses the frozen language tower of Qwen3-VL-Embedding-2B (Li et al., 2026), followed by two trainable transformer layers. FM and iDDM share this backbone. For iDDM, we concatenate 384 gated noise channels before block 20 and remove them before block 26, increasing the width of these six blocks to 1920. Block indices are zero-based.

Data and Training. Both models use a recaptioned subset of COYO (Byeon et al., 2022). FM minimizes velocity prediction error, while iDDM uses a generalized energy-kernel objective with four predictions per training example. We evaluate the 300k-step checkpoints using post-hoc EMA (Karras et al., 2024). Table 6 summarizes the configuration.

Evaluation. Fidelity to natural images is measured by COCO-FID, computed against 30,000 reference images following the standard MS-COCO evaluation protocol, together with CLIP Cosine similarity between generated images and their conditioning captions. We additionally evaluate compositional alignment on GenEval (Ghosh et al., 2023) and T2I-CompBench (Huang et al., 2023), and human preference alignment using HPS v2.1 (Wu et al., 2023). We use uniform Euler sampling with 4, 8, and 16 steps and sweep CFG for each model and step combination for FID and CLIP Cosine similarity jointly, and sweep individually for HPS v2.1 and T2I-CompBench.

C Dynamical Regimes of Diffusion Models

This appendix gives the details needed for the schedule construction in Section 3.2.1. The central object is not an equivalence between the OU reverse SDE and the flow-matching probability path, but the conditional law induced by an isotropic Gaussian corruption after normalization.

Table 13: Performance at 16, 8, and 4 sampling steps. Parentheses show absolute changes relative to each method's 16-step result. For HPS v2.1 and T2I-CompBench, CFG is selected separately per method, step count, and benchmark. T2I-CompBench selection maximizes the mean of its six category scores.

Metric	Method	16 steps (reference)	8 steps (Δ)	4 steps (Δ)
FID ↓	FM	11.200	26.327 (+15.127)	78.195 (+66.995)
	iDDM	11.265	19.403 (+8.138)	41.046 (+29.781)
CLIP Cosine ↑	FM	0.3141	0.3080 (-0.0061)	0.2744 (-0.0397)
	iDDM	0.3102	0.3030 (-0.0072)	0.2840 (-0.0262)
GenEval ↑	FM	0.364	0.285 (-0.079)	0.122 (-0.242)
	iDDM	0.353	0.297 (-0.056)	0.209 (-0.144)
HPS v2.1 ↑	FM	22.845	19.141 (-3.703)	13.913 (-8.932)
	iDDM	22.510	19.820 (-2.691)	16.968 (-5.542)
T2I CompBench				
Color ↑	FM	0.6786	0.5282 (-0.1505)	0.3185 (-0.3602)
	iDDM	0.5680	0.4820 (-0.0860)	0.4151 (-0.1529)
Shape ↑	FM	0.4351	0.3476 (-0.0875)	0.2484 (-0.1867)
	iDDM	0.3857	0.3406 (-0.0451)	0.3061 (-0.0796)
Texture ↑	FM	0.6274	0.4960 (-0.1314)	0.3189 (-0.3085)
	iDDM	0.5932	0.5348 (-0.0585)	0.4755 (-0.1177)
Spatial ↑	FM	0.2901	0.1628 (-0.1272)	0.0443 (-0.2457)
	iDDM	0.1178	0.0657 (-0.0521)	0.0232 (-0.0946)
Non-spatial ↑	FM	0.3021	0.2879 (-0.0142)	0.2532 (-0.0489)
	iDDM	0.2923	0.2809 (-0.0114)	0.2596 (-0.0327)
Complex ↑	FM	0.3517	0.2956 (-0.0561)	0.2102 (-0.1415)
	iDDM	0.3525	0.3234 (-0.0291)	0.2891 (-0.0634)

C.1 SNR COORDINATE AND OU/FM MAPPING

Consider any isotropic Gaussian corruption

$$x = \alpha x_1 + \sigma\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \qquad \rho = \frac{\alpha^2}{\sigma^2}. \tag{8}$$

For $\alpha > 0$, the deterministic normalization

$$z = \frac{x}{\alpha} = x_1 + \rho^{-1/2}\epsilon \tag{9}$$

shows that the conditional law $p(x_1 \mid z)$ depends on the corruption process through the normalized observation z and the scalar ρ. This is the only sense in which we compare OU and FM time: matched SNR defines the same normalized denoising problem, not the same stochastic process, reverse dynamics, path measure, or raw-coordinate marginal entropy.

The Ornstein–Uhlenbeck forward process used by Biroli et al. (2024), which is the constant-rate case of the VP-SDE family, is

$$\mathrm{d}x_\tau = -x_\tau\,\mathrm{d}\tau + \sqrt{2}\,\mathrm{d}W_\tau, \qquad x_{\tau=0} \sim p_{\text{data}}(x)\,, \tag{10}$$

with solution kernel

$$x_\tau \mid x_1 \sim \mathcal{N}\big(e^{-\tau}x_1,\ (1 - e^{-2\tau})\,I\big)\ , \qquad \mathrm{SNR}_{\mathrm{OU}}(\tau) = \frac{e^{-2\tau}}{1 - e^{-2\tau}}\,. \tag{11}$$

Our models use the standard linear interpolant $x_t = (1-t)x_0 + t\,x_1$, $x_0 \sim \mathcal{N}(0, I)$, $t \in [0, 1]$. The corresponding kernel is

$$x_t \mid x_1 \sim \mathcal{N}\big(t\,x_1,\ (1-t)^2 I\big)\ , \qquad \mathrm{SNR}_{\mathrm{FM}}(t) = \frac{t^2}{(1-t)^2}\,. \tag{12}$$

Equating the two SNRs gives the coordinate conversion

$$t = \frac{\sqrt{\rho}}{1+\sqrt{\rho}} = \frac{e^{-\tau}}{e^{-\tau}+\sqrt{1-e^{-2\tau}}}, \qquad \tau = -\tfrac{1}{2}\log\frac{\rho}{1+\rho}. \tag{13}$$

Applying Eq. 13 to a threshold means only that the two corruptions induce the same normalized denoising problem of Eq. 9.

C.2 Regime anchors

Speciation ρ_s. Biroli et al. (2024) define τ_s as the OU time at which the noise variance along the principal axis matches the signal variance, $\Lambda \cdot e^{-2\tau_s} = 1$. Equivalently, the variance of the leading mode of the OU noised marginal becomes informative about the data-manifold direction at this point. Expressed as an SNR value, $e^{-2\tau_s} = 1/\Lambda$ gives

$$\rho_s = \frac{1/\Lambda}{1-1/\Lambda} = \frac{1}{\Lambda-1}. \tag{14}$$

For $\Lambda \gg 1$, $\rho_s \approx 1/\Lambda$. We use this OU-derived SNR as a schedule anchor; its FM coordinate is $t_s^{\mathrm{FM}} = 1/(1+\sqrt{\Lambda-1})$.

Collapse ρ_c. The collapse criterion of Biroli et al. (2024) is OU-specific: it balances the differential entropy density of the empirical OU noised marginal $H(\tau)$ against the per-coordinate entropy density of n well-separated Gaussian blobs of variance $\Delta_\tau = 1 - e^{-2\tau}$,

$$H(\tau_c) = H^{\mathrm{sep}}(\tau_c), \qquad H^{\mathrm{sep}}(\tau) = \frac{\log n}{d} + \tfrac{1}{2}\big(1+\log(2\pi\Delta_\tau)\big). \tag{15}$$

The collapse time τ_c is the smallest OU time at which Eq. 15 holds; below τ_c, the empirical OU noised distribution is geometrically dominated by isolated training-example components, and a typical normalized observation has an identity posterior concentrated on a single index. We convert this OU diagnostic to an SNR anchor by setting

$$\rho_c = \frac{e^{-2\tau_c}}{1-e^{-2\tau_c}}. \tag{16}$$

Because $H(\tau)$ has no closed form, τ_c is computed numerically following Biroli et al. (2024).

Separation $\rho_{\mathrm{sep}}(\varepsilon)$. For the empirical finite-sample distribution, we use a geometric containment criterion: the $(1-\varepsilon)$-quantile OU noise ball around a training point should fit inside half the nearest-neighbor distance. With

$$d_{\min} = \min_{i \neq j} \|x_1^{(i)} - x_1^{(j)}\|,$$

and with $\chi^2_{d,1-\varepsilon}$ denoting the $(1-\varepsilon)$-quantile of a χ^2_d random variable, this criterion is

$$\sqrt{1-e^{-2\tau_{\mathrm{sep}}}}\,\sqrt{\chi^2_{d,1-\varepsilon}} = e^{-\tau_{\mathrm{sep}}}\,\frac{d_{\min}}{2}. \tag{17}$$

Solving and converting to SNR gives

$$\rho_{\mathrm{sep}}(\varepsilon) = \frac{e^{-2\tau_{\mathrm{sep}}}}{1-e^{-2\tau_{\mathrm{sep}}}} = \frac{4\chi^2_{d,1-\varepsilon}}{d^2_{\min}}. \tag{18}$$

This is a high-probability containment anchor. As with ρ_c, its ordering relative to the other anchors should be treated as a measured property of the dataset rather than a universal theorem.

C.3 ImageNet latent estimates and schedule construction

We compute the characteristic times on REPA-E (Leng et al., 2025) (SD-VAE backbone) latents of ImageNet-256. Images are center-cropped to 256×256, rescaled to $[-1,1]$, and encoded with the frozen autoencoder; the resulting latent is $32\times32\times4$, of which we keep the first channel only, giving

Table 14: Characteristic times on centered first-channel REPA-E latents for ImageNet-256 ($n = 50{,}000$ samples drawn across all classes). ρ_s and ρ_c are OU-derived posterior-concentration diagnostics; ρ_{sep} is an empirical high-probability containment diagnostic using the χ^2 noise norm. The τ and t columns are coordinate images from Eq. 14, not an equivalence of OU and FM dynamics.

Threshold	Definition	ρ	τ_{OU}	t_{FM}
ρ_s	$1/(\Lambda - 1)$	0.015	2.11	0.11
ρ_c	$H(\tau_c) = H^{\text{sep}}(\tau_c)$	0.03	1.77	0.15
$\rho_{\text{sep}}(0.1)$	$4\chi^2_{d,0.9}/d^2_{\min}$	33.11	0.015	0.85

feature dimension $d = 32\times32 = 1024$. The analyzed set $\{x_1^{(i)}\}_{i=1}^n$ is a uniformly sampled subset of $n = 50{,}000$ training latents drawn across all 1,000 ImageNet classes, centered to zero per-coordinate mean. From this set: Λ is the largest eigenvalue of the empirical $d \times d$ covariance, giving ρ_s; ρ_c is obtained by solving the OU entropy balance $H(\tau_c) = H^{\text{sep}}(\tau_c)$ numerically, with the empirical entropy density estimated by Monte Carlo from the n-component mixture using $n' = 20{,}000$ noisy probe samples; $d_{\min}$ is computed by an exact pairwise distance scan over the centered set, and used with containment level $1 - \varepsilon = 0.9$ to instantiate $\rho_{\text{sep}}(\varepsilon)$ via Eq. 18. Table 14 reports the resulting SNR characteristic times and their OU/FM coordinate images.

We instantiate our piecewise schedule to be 1 when $\rho < \rho_s$, 0 when $\rho > \rho_{\text{sep}}$ and a linear interpolation in between.

D On Scheduling $\lambda(t)$ and $\beta(t)$

Regarding $(\lambda(t), \beta(t))$ schedules, conditional on introducing regression bias, its orientation toward the high-SNR/data end minimizes an explicit posterior-covariance bias criterion.

In our FM convention,

$$Y_\rho = X_1 + \rho^{-1/2} Z, \qquad \rho = \frac{t^2}{(1-t)^2}.$$

For $X_1 \in L^2$, the average posterior variance

$$v(\rho) = d^{-1}\mathbb{E}\,\operatorname{tr}\operatorname{Cov}(X_1 \mid Y_\rho)$$

satisfies $v(\rho) \le \rho^{-1}$, since the conditional mean has no larger MSE than the estimator Y_ρ. It is also non-increasing in ρ, because a lower-SNR Gaussian observation is a degradation of a higher-SNR one. Thus, throughout IIIb, $v(\rho) \le \rho_{\text{sep}}^{-1}$.

Under the isotropic-Gaussian population surrogate of De Bortoli et al. (2025b), the generalized energy-score optimum preserves the conditional mean and multiplies its covariance by

$$f(\lambda, \beta) = \left(2\lambda^{-2/(2-\beta)} - 1\right)^{-1},$$

for $0 < \lambda \le 1$, $0 < \beta < 2$. Hence, writing $a_t = 1 - f_t$, the expected covariance removed per coordinate is

$$a_t v(\rho_t) \le \frac{a_t}{\rho_t}.$$

A finite FM transition introduces only the additional factor $((u-t)/(1-t))^2$; Proposition C.7 in De Bortoli et al. (2025b) gives the analogous decomposition for the Gaussian diffusion bridge.

Finally, for a fixed collection of shrinkage values, define

$$\mathcal{B}(a) = \int a_t v(\rho_t)\, d\nu(t).$$

If $t_1 < t_2$, then $v(\rho_{t_1}) \ge v(\rho_{t_2})$; swapping incorrectly ordered values $a_{t_1} > a_{t_2}$ changes $\mathcal{B}$ by

$$(a_{t_1} - a_{t_2})\big(v(\rho_{t_2}) - v(\rho_{t_1})\big) \le 0.$$

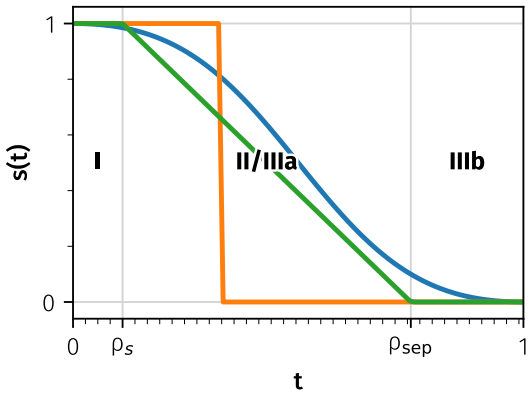

Figure 6: Schedule & Regime Illustration.

Therefore $\mathcal{B}$ is minimized by assigning the largest shrinkage to the highest SNRs. This proves the schedule orientation for this covariance criterion, not a uniquely optimal smooth profile or terminal-FID optimum. The regime anchors motivate the transition locations, while the precise profile and separate λ/β schedules remain empirical and are ablated in Table 3.

Empirically, we consider the following shape functions:

$$\text{Constant:} \quad s(t) = 1 \quad \big(\lambda(t) = \lambda_{\max},\ \beta(t) = \beta_{\min}\big), \tag{19}$$

$$\text{Linear:} \quad s(t) = 1 - t, \tag{20}$$

$$\text{Dynamical regimes:} \quad s(t) = \text{clip}\left(\frac{t_{\text{sep}} - t}{t_{\text{sep}} - t_s},\ 0,\ 1\right), \tag{21}$$

$$\text{Step:} \quad s(t) = \mathbb{1}[t \le \kappa], \tag{22}$$

$$\text{SNR:} \quad s(t) = \frac{(1-t)^{2p}}{(1-t)^{2p} + t^{2p}} = \frac{1}{1 + \text{SNR}(t)^p}, \qquad \text{SNR}(t) = \frac{t^2}{(1-t)^2}, \tag{23}$$

where step-κ in Table 3 refers to Eq. 22.

Finally, we provide some visualization of the different schedules as a function of the SNR in Figure 6.

E ξ Conditioning

We explore various conditioning mechanisms, summarized in Section 3.1 and Fig. 3 and Table 2b. After deferred expansion, each population member carries an independent noise tensor $\xi \in \mathbb{R}^{H' \times W' \times d_{\text{noise}}}$ that must be injected into the model (starting) at layer ℓ_{start}. We consider four relevant families of mechanisms, each combined with adaptive gating, detailed in the following paragraphs.

Adaptive Gating. With the exception of the input-level concatenation baseline from DDM (De Bortoli et al., 2025b), every mechanism passes ξ_j through a shared adaptive multiplicative gate

$$\tilde{\xi}_j = \xi_j \odot (W_{\text{gate}}\, c + 1), \tag{24}$$

where W_{gate} is a zero-initialized linear layer mapping the conditioning vector $c \in \mathbb{R}^{d_{\text{cond}}}$ (a combined embedding of timestep t and, in the case of class-conditional image generation, the class) to a per-element multiplicative gate. Zero initialization ensures that the gate starts at 1 (identity scaling), so the model initially behaves identically to standard DDM model (De Bortoli et al., 2025b) and gradually learns to modulate ξ as needed.

E.1 Conditioning Mechanisms

Channel-wise Concatenation. Concatenation appends $\tilde{\xi}_j$ to the residual stream along the channel dimension at layer ℓ_{start}. In the simplest setting, this results in a widening of the per-token feature

width from d to $d + d_{\text{cat}}$. This widened residual stream is processed by all subsequent layers, and the velocity predictions are read out *only* from the original channels of the final hidden state as

$$\hat{v}_\theta \;=\; \text{Unembed}\big(h_L[\ldots, :d]\big). \tag{25}$$

We study three variants of this family:

- **Cat (widen).** All projections in layers $\ell_{\text{start}}, \ldots, L-1$ operate at the widened width $d + d_{\text{cat}}$. Since attention QKV and FFN intermediate dimensions are typically determined based on the residual stream width, all are enlarged proportionally. This is the most expressive variant, but adds extra learnable parameters and FLOPS in the late layers, conflating results with overall improved capacity.
- **Cat (fixed inner).** The residual stream is widened to $d + d_{\text{cat}}$ as before, but each block's inner computation is held at the original width. The signal $\tilde{\xi}_j$ can persist in the residual stream across layers, while the bulk of the computation is unchanged. This is our default, as it preserves the approx. parameter count of the baseline.
- **Cat (substitute).** The final variant preserves the number of trainable parameters and the computation of the distributional layers *exactly* identical. This is achieved by *not* widening the residual stream to accommodate d_{cat}, but instead narrowing the previous residual information to $d - d_{\text{cat}}$ dimensions (by dropping a set of dimensions) before concatenation.

Residual Addition. Instead of concatenating the gated noise, it is directly added to the residual stream at layer ℓ_{start}. To project it to the model width, a linear map $W_{\text{add}} \in \mathbb{R}^{d_{\text{cat}} \times d}$ is used, resulting in the residual stream update

$$h \;\leftarrow\; h + \tilde{\xi}_j W_{\text{add}}. \tag{26}$$

This preserves both width and parameter count of the late layers, with only the linear map introducing additional learnable parameters.

Adaptive Normalization (Ada-Norm). $\tilde{\xi}_j$ is mapped through a per-token MLP to scale (and, optionally, shift, if using LayerNorm instead of RMSNorm) parameters that are added to the existing adaptive pre-norms typically used for timestep and class conditioning in diffusion transformers. This keeps the residual stream width unchanged.

Register Tokens/Sequence-wise Concatenation. $\tilde{\xi}_j$ is projected and reshaped into n_{tok} additional sequence tokens with learned positional embeddings, and appended to the token stream at layer ℓ_{start}. The extra tokens participate in self-attention but are dropped from the velocity readout, similar to the additional channels introduced in channel-wise concatenation.

Concatenation at Input (Naïve Baseline). Finally, we also consider the original DDM formulation (De Bortoli et al., 2025b), where $\tilde{\xi}_j$ is reshaped to the spatial grid of x_t and concatenated to it along the channel dimension before the patch embedding, with all L transformer layers processing the noise. This variant is incompatible with deferred expansion ($\ell_{\text{start}} > 0$) and is reported as a reference only; it is strictly dominated by the post-expansion variants in our experiments.

Discussion. Table 2b shows that fixed-inner concatenation performs substantially better than Ada-Norm and token conditioning in the few-step regime. We hypothesize that, with only a few particle-specific layers after expansion, directly carrying ξ in the residual stream makes the auxiliary signal easier to preserve than routing it indirectly through normalization parameters or attention.

Fixed-inner concatenation gives every spatial token a persistent particle-specific feature subspace throughout the remaining residual blocks, while leaving the expensive internal attention and MLP dimensions unchanged. Additive conditioning instead superposes the auxiliary signal onto the shared representation, making it easier for subsequent transformations to attenuate or entangle that signal. Ada-Norm routes (ξ) indirectly through modulation parameters, while token conditioning requires attention to propagate particle information from a small number of auxiliary tokens to every spatial token. This may be difficult when only a few distributional layers remain after expansion.

These mechanisms are all reasonable a priori; Table 2b provides the empirical reason for selecting fixed-inner concatenation. We present this explanation as an hypothesis consistent with the observed results, rather than as a theoretically established mechanism.

F KERNEL SPATIAL MODES

The generalized energy score in Eq. (4) is defined for vector-valued predictions, but in our setting, each prediction $\hat{x}_\theta(t, x_t, \xi_j) \in \mathbb{R}^{H' \times W' \times C}$ is a sequence of patch tokens. The score, therefore, admits two natural instantiations that differ in the geometry over which the kernel norm $\|\cdot\|_2^\beta$ is evaluated.

Global Kernel. Each prediction is flattened into a single $H'W'C$-dimensional vector and the score $S_{\lambda,\beta}$ is evaluated on the resulting vector space:

$$S^{\text{global}}_{\lambda,\beta}(p, y) \;=\; -\,\mathbb{E}_p\Big[\|\text{vec}(X) - \text{vec}(y)\|_2^\beta\Big] + \tfrac{\lambda}{2}\,\mathbb{E}_{p\otimes p}\Big[\|\text{vec}(X) - \text{vec}(X')\|_2^\beta\Big]. \tag{27}$$

This is the implicit choice in most MMD/energy-distance work (Gneiting & Raftery, 2007; De Bortoli et al., 2025b) – it imposes a single global geometry over the entire prediction.

Local Kernel. Alternatively, the score is evaluated independently on each spatial position $X_{h,w} \in \mathbb{R}^C$ and averaged over positions,

$$S^{\text{local}}_{\lambda,\beta}(p, y) \;=\; \frac{1}{H'W'} \sum_{h,w} S_{\lambda,\beta}\big(p_{h,w},\, y_{h,w}\big), \tag{28}$$

which corresponds to a separable product kernel over the spatial grid. Each per-position score is strictly proper on its slice of $\mathbb{R}^C$, so the average is also strictly proper as a scoring rule for the per-position marginals of p. The local mode emphasizes the local structure of the velocity prediction, at the cost of decoupling the score from cross-spatial dependencies. We use the local mode by default since it offers significant practical performance improvements; the comparison is reported in Table 3d.

G TRAINING-TIME t SAMPLING

The DDM loss in Eq. (5) composes a per-time score with a sampling density $p(t)$ and a loss weighting $w(t)$. We hold $w(t) \equiv 1$ throughout and ablate only the sampling density. Since reweighting the sampling density without compensating in $w(t)$ amounts to an effective t-dependent loss weighting, this also controls the relative supervision allocated to each noise level.

We compare four samplers, three of which belong to the logit-normal family $t = \sigma(\mu + sZ)$ with $Z \sim \mathcal{N}(0, 1)$ and σ the logistic sigmoid:

- `uniform`: $t \sim \mathcal{U}(0, 1)$. Unbiased baseline with constant density on $[0, 1]$.
- `logit-normal` ($\mu = 0$, $s = 1$): the default of Esser et al. (2024); symmetric and peaked at $t = 0.5$.
- `jit` ($\mu = -0.8$, $s = 0.8$): the sampler used by Li & He (2025); mode at $t \approx 0.25$, shifted toward the noise end (low SNR).
- `imf` ($\mu = -0.4$, $s = 1$): the sampler used by Geng et al. (2026); mode at $t \approx 0.32$, with a similar bias to `jit` but broader support.

In terms of the anchors of Table 14, jit places its mode at $t \approx 0.25$ ($\rho \approx 0.1$), inside Regime IIIa. It allocates 72.5% of training times to $\rho \in [\rho_c, 1]$, compared with 35.2% for uniform and 46.0% for logit-normal$(0, 1)$. This is the low-SNR part of the post-collapse regime, where the posterior is mode-dominated but still broad and where our schedule moves the objective from distributional toward regression-like behaviour. Table 3c shows that this allocation improves both few- and many-step FID.

The FM reference and Configs A-E use the `logit-normal` sampler. Only F switches to `jit`. The sampler-matched comparison is FM and Config E and therefore isolates the distributional components, which reduce 4-step FID from 26.53 to 14.33 (about 91% of the total 4-step gain).

H MMD LINEAGE

The generalized energy score (Eq. 4) can be modified to yield a generalized kernel score (see (De Bortoli et al., 2025b)) by replacing $-||x - y||_2^\beta$ by a positive definite kernel $k(x, y)$. Such kernel

Algorithm 1 DDM training with deferred population expansion. The first ℓ_{start} layers run once at batch size B; after population expansion and ξ conditioning, the final H layers run at effective batch size $B \times m$.

```
# model: DistributionalTransformer with L layers; expansion at layer L-H
# kernel: GeneralizedKernelScore with schedules lambda(t), beta(t)

# FM time; t=0 noise, t=1 data
t  = sample_t(B)
x0 = randn_like(x1)
xt = (1 - t) * x0 + t * x1
# one xi per population member
xi = sample_xi(B * m)

h, cond = model.embed(xt), model.condition(t, y)

# shared trunk at batch B
for layer in model.layers[:L-H]:
    h = layer(h, cond)

# population expansion
h    = repeat(h,    "b ... -> (b m) ...", m=m)
cond = repeat(cond, "b ... -> (b m) ...", m=m)
# AdaLN / additive / concat / tokens
h, cond = inject_xi(h, cond, xi)

# repeated head at batch B * m
for layer in model.layers[L-H:]:
    h = layer(h, cond)
v_pred = model.unembed(h)

v_pred = rearrange(v_pred, "(b m) ... -> b m ...", m=m)
tgt    = repeat(x1 - x0,   "b ... -> b m ...", m=m)
loss   = -kernel.score(v_pred, tgt, t)
loss.mean().backward()
```

score then reduces to a Dirac-target MMD objective up to a target-dependent constant when λ=1; MMD-based generative training has a long history spanning GMMNs (Li et al., 2015) and early kernel score objectives such as DISCO Nets (Bouchacourt et al., 2016). IMM (Zhou et al., 2025a) is the population-vs-population MMD generalisation in which the right-hand argument is itself a population, with Consistency Models (Song et al., 2023; Song & Dhariwal, 2024) appearing as a single-particle, first-moment limit in the IMM analysis. DDMs and IMM are therefore instances of the same kernel score family with different right-hand arguments; our proposed schedules clarify this connection.

I Compute Requirements

All iDDM models were trained on ImageNet-256^2 with 4 $\times$ H200 for B-sized models for roughly 12 hours and 8 $\times$ H200 for XL-sized models. Reproduced models may differ in computational requirements based on their additional training overhead.

J Additional Visualizations

We provide additional uncurated samples using 4 and 50 steps, respectively, in Figure 7-Figure 18.



Figure 7: Uncurated ImageNet-256^2 samples using 4 steps.

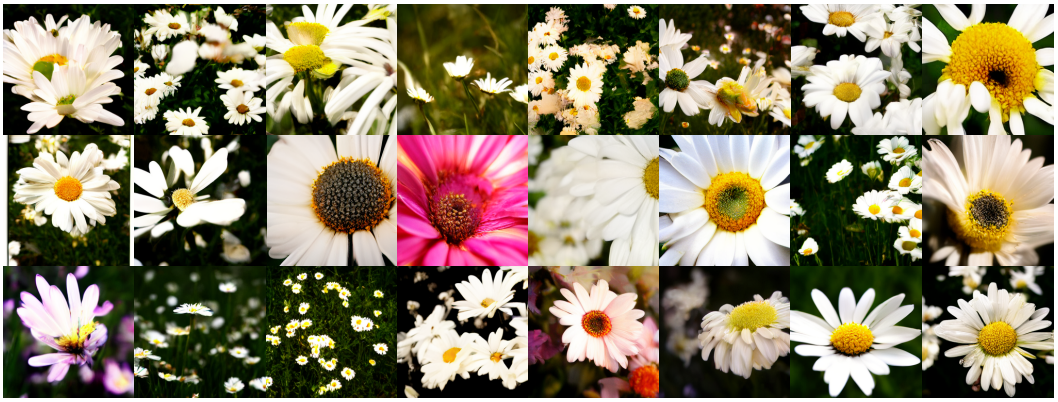

Figure 8: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 9: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 10: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 11: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 12: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 13: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 14: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 15: Uncurated ImageNet-256^2 samples using 4 steps.



Figure 16: Uncurated ImageNet-256^2 samples using 50 steps.



Figure 17: Uncurated ImageNet-256^2 samples using 50 steps.



Figure 18: Uncurated ImageNet-256^2 samples using 50 steps.

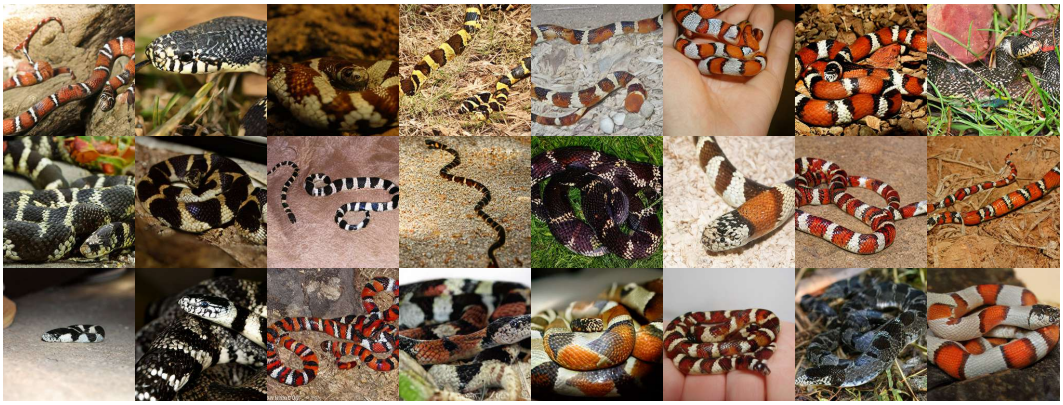

Figure 19: Uncurated ImageNet-256^2 samples using 50 steps.



Figure 20: Uncurated ImageNet-256^2 samples using 50 steps.



Figure 21: Uncurated ImageNet-256^2 samples using 50 steps.



Figure 22: Uncurated ImageNet-256^2 samples using 50 steps.



Figure 23: Uncurated ImageNet-256^2 samples using 50 steps.



Figure 24: Uncurated ImageNet-256^2 samples using 50 steps.