BEYOND THE PF-ODE: LEARN THE REVERSE DYNAMICS FOR FEW-STEP DIFFUSION SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion solvers accelerate sampling by numerically approximating the reverse process with only a few function evaluations. This reverse process is commonly formulated as either a *deterministic probability-flow ODE (PF-ODE)* or a *stochastic reverse SDE*. Reverse SDEs can dissipate distribution mismatch accumulated during sampling, but stronger stochasticity also amplifies the effect of score error. PF-ODEs avoid this additional stochasticity, yet preserve inherited distribution mismatch under exact scores. We show that this creates a step-dependent trade-off between distribution correction and score reliability, suggesting that the preferable reverse dynamics *need not remain fixed* throughout sampling. Motivated by this observation, we propose **RD-Solver**, which jointly learns the reverse dynamics and their discretization. Specifically, RD-Solver learns an interval-wise stochasticity coefficient together with multistep predictor–corrector weights and calibrated score-evaluation times. We optimize these parameters with a frozen network using an endpoint loss that upper-bounds the squared 2-Wasserstein distance to a high-accuracy teacher in the supervision space. Experiments across diffusion and flow models demonstrate improved few-step generation, achieving FID scores of 2.71 on CIFAR-10 and 4.62 on ImageNet-64 with five function evaluations (5 NFEs).

1 INTRODUCTION

Diffusion and flow models have become a general-purpose generative paradigm (Ho et al., 2020; Song et al., 2021b; Karras et al., 2022; Sohl-Dickstein et al., 2015; Lipman et al., 2023; Liu et al., 2023; Kingma et al., 2021), with strong performance across image generation (Rombach et al., 2022; Labs et al., 2025; Labs, 2024), video generation (Zheng et al., 2024; Seedance et al., 2026), 3D content generation (Poole et al., 2023; Lin et al., 2023), and scientific domains (Watson et al., 2022; Hoogeboom et al., 2022; Kapoor et al., 2024). Their main computational bottleneck is iterative sampling: every reverse step evaluates a large denoising network, and reducing NFE rapidly magnifies numerical error. Accurate few-step sampling has therefore motivated a large body of specialized diffusion solvers (Liu et al., 2022; Lu et al., 2022; 2025; Zhang & Chen, 2023; Zhao et al., 2023). Specialized numerical solvers reduce this cost by taking fewer, larger steps (Lu et al., 2022; Zhao et al., 2023). Learned solvers further adapt sampling to a pretrained model by optimizing a small set of solver parameters while keeping the network fixed (Shaul et al., 2024; Zhou et al., 2024; Frankel et al., 2025; Wang et al., 2026; Yuan et al., 2026). This provides a parameter-efficient route to few-step generation.

ODE-based learned solvers typically optimize the discretization of a prescribed probability-flow ODE (*PF-ODE*). Yet this is only one choice of ***Reverse Dynamics***. The *PF-ODE* and reverse stochastic differential equation (*Reverse SDE*) belong to a family with identical target marginals under exact scores, initialization, and continuous-time integration (Song et al., 2021b; Albergo et al., 2025). In few-step sampling, however, each transition receives a distribution already affected by score and integration errors. Marginal equivalence therefore leaves a practical question unanswered: *which reverse dynamics should act on these imperfect distributions at a fixed NFE?*

We introduce **RD-Solver**, which learns ***Reverse Dynamics*** for distribution matching together with their discretization. We theoretically show that, with an exact score, common PF transport preserves an existing Kullback–Leibler (KL) mismatch, whereas additional score drift paired with diffusion

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.



Figure 1: **RD-Solver on FLUX.1-dev with only 7 NFEs.** Eleven selected samples generated at multiple native aspect ratios using prompts across different scenarios.

contributes KL dissipation. When the score is imperfect, stronger stochasticity also gives the score error a larger coefficient. The benefit of correction thus depends on both the accumulated mismatch and score reliability, motivating a learnable stochasticity coefficient for each interval. This continuous-time analysis motivates the choice of dynamics but does not guarantee error reduction under coarse discretization.

RD-Solver directly parameterizes the reverse drift–diffusion pair: each interval coefficient jointly controls score drift and noise variance. We approximate its drift integral with a learned multistep predictor–corrector that reuses historical evaluations (Zhao et al., 2023). A learned score-evaluation time further compensates for the predictor's possible mismatch with its nominal noise level. Changing the dynamics changes the integral and the states at which it is evaluated, so we fit these parameters jointly. We train the solver to match the output distribution of a high-accuracy teacher using the same pretrained model. Sharing initial noise yields a computable upper bound on their squared 2-Wasserstein distance in the supervision space. Empirically, RD-Solver achieves the lowest FID in most tested settings among the compared solvers.

Our contributions are threefold:

- We characterize the distribution matching action of reverse dynamics and its interaction with score error, motivating interval-wise learning of stochasticity.
- We develop RD-Solver, which jointly learns reverse dynamics and their multistep discretization through an upper bound on the output-distribution discrepancy in the supervision space.
- Extensive evaluations on multiple generative benchmarks show that RD-Solver achieves state-of-the-art performance with significant FID gains over existing solvers.

2 Related Work

Numerically designed diffusion solvers. DDIM (Song et al., 2021a) enables deterministic sampling through a non-Markovian process, while EDM (Karras et al., 2022) systematizes noise schedules, preconditioning, and numerical design. DEIS (Zhang & Chen, 2023) uses exponential integration of the semi-linear *PF-ODE*, DPM-Solver (Lu et al., 2022) develops high-order expansions, and DPM-Solver++ (Lu et al., 2025) adapts them to data prediction and guided sampling. PNDM (Liu et al., 2022) reuses model outputs, UniPC (Zhao et al., 2023) combines predictor–corrector updates, and DPM-Solver-v3 (Zheng et al., 2023) uses empirical model statistics. Restart (Xu et al., 2023) alternates renoising and ODE integration; SA-Solver (Xue et al., 2023) uses stochastic Adams updates. These methods prescribe the dynamics and integration coefficients.

Learning-based diffusion solvers. AYS (Sabour et al., 2024), LD3 (Tong et al., 2025), and TPDM (Ye et al., 2025) learn timesteps; INDIS (Yuan et al., 2026) adapts them to each input instance, and BézierFlow (Min et al., 2026) learns Bézier stochastic interpolant schedules. AMED-Solver (Zhou et al., 2024) and EPD (Zhu et al., 2025) learn where to evaluate additional model information. Bespoke Solvers (Shaul et al., 2024) and S4S (Frankel et al., 2025) search model-specific solver families, while Differentiable Solver Search (DSS) (Wang et al., 2025b) makes this search end-to-end differentiable. D-ODE (Kim et al., 2024) learns weights over previous model outputs; DLMS (Liang et al., 2025) learns trajectory-dependent multistep weights; GAS (Oganov et al., 2026) learns solver weights with adversarial supervision; DyWeight (Zhao et al., 2026) learns unconstrained history weights that also calibrate step size. AdaSDE (Wang et al., 2025a) learns renoising and midpoint corrections with stagewise supervision; Dual-Solver (Park & Lee, 2026) combines prediction parameterizations. We instead jointly learn the drift–diffusion pair and its multistep quadrature under endpoint supervision.

3 Preliminaries

Forward diffusion. Diffusion Models can be formulated in continuous time through a linear stochastic differential equation (SDE) that gradually perturbs data into noise (Ho et al., 2020; Song et al., 2021b; Lipman et al., 2023; Sohl-Dickstein et al., 2015; Liu et al., 2023). Taking $t = 0$ as data and $t = T$ as noise, the forward process is

$$\mathrm{d}x_t = f(t)x_t\,\mathrm{d}t + g(t)\,\mathrm{d}w_t, \qquad x_0 \sim p_{\mathrm{data}}, \quad t \in [0, T], \tag{1}$$

where w_t is a standard Wiener process and f, g are prescribed scalar coefficients. Its Gaussian transition kernel admits the representation

$$p_{t|0}(x_t \mid x_0) = \mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 I), \tag{2}$$

with mean and noise schedules α_t, σ_t, $\alpha_0 = 1$, $\sigma_0 = 0$, and the d-dimensional identity I. The marginal density is $p_t(x) = \int p_{t|0}(x \mid x_0)p_{\mathrm{data}}(x_0)\,\mathrm{d}x_0$. The terminal noise level is chosen so that p_T is well approximated by a tractable Gaussian prior.

Reverse sampling. Let $s_t^\star(x) = \nabla_x \log p_t(x)$ be the marginal score. Starting from $x_T \sim p_T$, both the *probability-flow ODE* (*PF-ODE*) and *Reverse SDE* recover the same marginal path in reverse (Anderson, 1982; Song et al., 2021b):

$$\begin{aligned} \mathrm{d}x_t &= \left[f(t)x_t - \tfrac{1}{2}g^2(t)s_t^\star(x_t)\right]\mathrm{d}t, && \blacktriangleright\ \textit{PF-ODE}, \\ \mathrm{d}x_t &= \left[f(t)x_t - g^2(t)s_t^\star(x_t)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{w}_t, && \blacktriangleright\ \textit{Reverse SDE}. \end{aligned} \tag{3}$$

Here time runs from T to 0, $\mathrm{d}t < 0$, and $\bar{w}_t$ is a reverse-time Wiener process. Marginal equivalence assumes an exact score, initialization from p_T, and exact continuous-time integration.

Learning the score. A network $s_\theta(x, t)$ estimates $s_t^\star(x)$ by minimizing the denoising score-matching objective (Song et al., 2021b; Song & Ermon, 2019)

$$\mathcal{L}_{\mathrm{score}}(\theta) = \mathbb{E}_{t,x_0,x_t}\left[\lambda(t)\left\|s_\theta(x_t, t) - \nabla_{x_t}\log p_{t|0}(x_t \mid x_0)\right\|_2^2\right], \qquad \lambda(t) > 0, \tag{4}$$

where $\lambda(t)$ is a positive weighting function on the score-matching error at time t, t is uniform on $(0, T)$, $x_0 \sim p_{\mathrm{data}}$, and $x_t \sim p_{t|0}(\cdot \mid x_0)$. In practice, sampling substitutes s_θ into Eq. (3) and applies a numerical solver.

4 RD-Solver

RD-Solver learns reverse dynamics and their discretization at a fixed NFE with a frozen score network. We analyze the effect of stochasticity on distribution mismatch, then construct and train a multistep solver.

4.1 Learning *Reverse Dynamics* for distribution matching

Varying diffusion strength together with the corresponding score drift preserves the ideal marginals (Song et al., 2021b; Albergo et al., 2025):

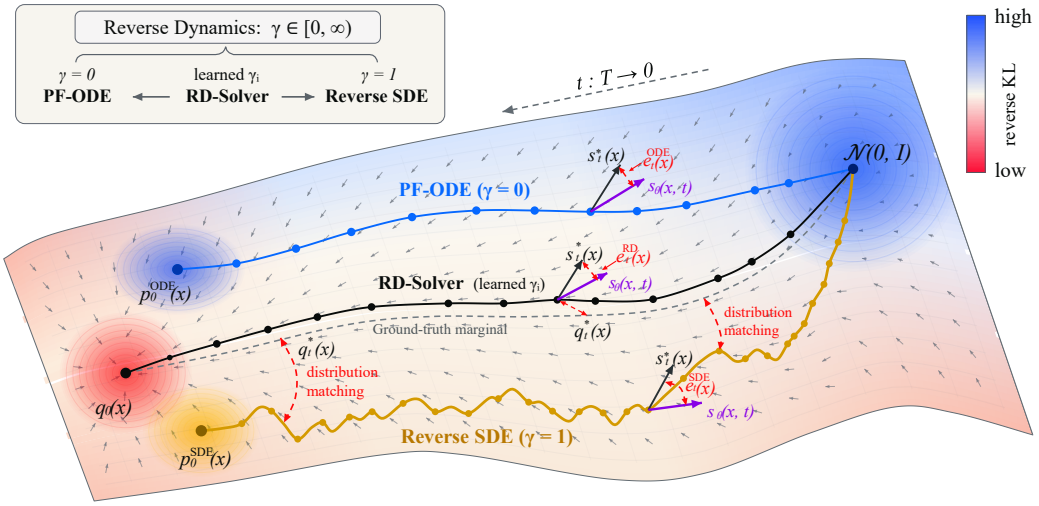

Figure 2: **Overview of reverse dynamics.** Comparison of the PF-ODE ($\gamma = 0$), Reverse SDE ($\gamma = 1$), and RD-Solver with learned, step-dependent γ_i. The target and sampling distributions are shown in beige and blue, respectively.

Proposition 4.1 (Marginal-equivalent ***Reverse Dynamics***). *Let $\gamma : [0, T] \to \mathbb{R}_{\geq 0}$ be bounded and measurable. Under existence and uniqueness of the associated density evolution, the process initialized at $x_T \sim p_T$*

$$\mathrm{d}x_t = \left[f(t)x_t - \frac{1+\gamma(t)}{2} g^2(t) s_t^\star(x_t) \right] \mathrm{d}t + \sqrt{\gamma(t)} g(t)\, \mathrm{d}\bar{w}_t, \qquad \blacktriangleright \textbf{ Reverse Dynamics} \tag{5}$$

has marginal density p_t when integrated from $t = T$ to $t = 0$. The choices $\gamma = 0$ and $\gamma = 1$ recover the PF-ODE and Reverse SDE in Eq. (3), respectively.

On a grid $T = t_0 > t_1 > \cdots > t_n = 0$, the numerical state x_{t_i} has law q_{t_i}, which may differ from p_{t_i}. To isolate how the dynamics act on this inherited mismatch, start Eq. (5) from q_{t_i} with score s_θ, and denote its continuous-time density on $[t_{i+1}, t_i]$ by q_t. Define the exact PF drift and score error as

$$b_t(x) = f(t)x - \frac{g^2(t)}{2} s_t^\star(x), \qquad e_t(x) = s_\theta(x, t) - s_t^\star(x). \tag{6}$$

A relative-entropy decomposition (Schaeffer et al., 2025; Cao et al., 2023) separates transport, distribution matching, and score error.

Theorem 4.2 (Distribution matching along the reverse process). *Suppose p_t, q_t are positive and sufficiently smooth, with the boundary decay needed for integration by parts. Let $r_t(x) = \nabla_x \log(q_t(x)/p_t(x))$. Suppressing the spatial argument,*

$$-\partial_t q_t = \nabla_x \cdot (b_t q_t) + \underbrace{\frac{\gamma(t) g^2(t)}{2} \nabla_x \cdot (q_t r_t)}_{\textit{distribution matching}} - \underbrace{\frac{(1+\gamma(t)) g^2(t)}{2} \nabla_x \cdot (q_t e_t)}_{\textit{score error}}. \tag{7}$$

The first term transports q_t with the PF field of p_t. The second combines additional score drift and diffusion, acts on the relative score r_t, and vanishes when $q_t = p_t$. The third reflects score error. For $\mathrm{KL}(q_t \| p_t) = \int q_t \log(q_t/p_t)\, \mathrm{d}x$, we obtain:

Corollary 4.3 (Derivative of $\mathrm{KL}(q_t \| p_t)$). *Under the assumptions of Theorem 4.2,*

$$-\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{KL}(q_t \| p_t) = \frac{g^2(t)}{2} \left[(1+\gamma(t)) \int q_t e_t^\top r_t \, \mathrm{d}x - \gamma(t) \int q_t \|r_t\|_2^2 \, \mathrm{d}x \right]. \tag{8}$$

In particular, an exact score gives

$$e_t = 0 \quad \Longrightarrow \quad -\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(q_t\|p_t) = -\frac{\gamma(t)g^2(t)}{2}\int q_t\|r_t\|_2^2\,\mathrm{d}x \le 0. \tag{9}$$

More generally, write $\|v\|_{q_t}^2 = \int q_t(x)\|v(x)\|_2^2\,\mathrm{d}x$. *For* $g(t) > 0$ *and* $\gamma(t) > 0$, *a sufficient condition for strict decrease is*

$$(1+\gamma(t))\|e_t\|_{q_t} < \gamma(t)\|r_t\|_{q_t} \quad \Longrightarrow \quad -\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(q_t\|p_t) < 0. \tag{10}$$

Effect of stochasticity. Here $-\mathrm{d}/\mathrm{d}t$ follows the sampling direction $T \to 0$. With an exact score, PF transport ($\gamma = 0$) preserves inherited KL mismatch, even under exact integration. Positive γ reduces it when $g(t) > 0$ and $\|r_t\|_{q_t} > 0$, whether it arose from earlier score or discretization errors.

Learning the correction strength. Score error can help or oppose dissipation, and its coefficient also grows with γ. The balance in Eq. (10) varies with time and preceding steps. It does not specify an optimal schedule: q_t depends on the dynamics, and the required densities are unknown. Rather than fix $\gamma = 0$ or 1, we learn one coefficient per interval:

$$\gamma(t) = \gamma_i \quad \text{for } t \in (t_{i+1}, t_i], \qquad \gamma_i \ge 0, \quad i = 0, \dots, n-1. \tag{11}$$

These continuous-time statements do not guarantee that a coarse numerical step reduces KL. Changing γ_i changes both the drift integral and the states at which the score is evaluated. We therefore learn its numerical update together with the dynamics, rather than choose γ_i independently of the discretization. Proofs and the sufficient-condition derivation are in Appendices A.1–A.3.

4.2 A LEARNABLE SOLVER FOR THE SELECTED DYNAMICS

Before time calibration, a step from t_i to $t_{i+1} > 0$ uses

$$\Delta t_i = t_i - t_{i+1} > 0, \qquad v_i = \int_{t_{i+1}}^{t_i} g^2(t)\,\mathrm{d}t, \qquad \xi_i \sim \mathcal{N}(0, I). \tag{12}$$

Integrating Eq. (5) with coefficient γ_i and frozen score network gives

$$x_{t_{i+1}} = x_{t_i} + \int_{t_{i+1}}^{t_i}\left[-f(t)x_t + \frac{1+\gamma_i}{2}g^2(t)s_\theta(x_t, t)\right]\mathrm{d}t + \sqrt{\gamma_i v_i}\,\xi_i. \tag{13}$$

The shared γ_i pairs the score drift with noise covariance $\gamma_i v_i I$, producing the matching term in Eq. (7). We approximate the full drift integral, including $-f(t)x_t$, with a multistep predictor–corrector that reuses score evaluations (Zhao et al., 2023). For each $m < i$, we cache the predicted state and score, then recompute its drift with the current γ_i:

$$u_{i,m}^\theta = -f(t_{m+1})\widetilde{x}_{t_{m+1}} + \frac{1+\gamma_i}{2}g^2(t_{m+1})s_\theta(\widetilde{x}_{t_{m+1}}, t_{m+1}), \qquad m < i. \tag{14}$$

Changing γ_i thus requires no extra network evaluation at past states: we reuse their scores, not drift values tied to previous coefficients. The predictor uses k_i cached evaluations. One fresh score evaluation at the predicted state gives $u_{i,i}^\theta$ by setting $m = i$ in Eq. (14); the corrector combines it with the cached values:

$$\begin{cases} \textbf{RD-predictor:} & \widetilde{x}_{t_{i+1}} = x_{t_i} + \Delta t_i \sum_{j=1}^{k_i} a_{i,j} u_{i,i-j}^\theta + \sqrt{\gamma_i v_i}\,\xi_i, \\ \textbf{RD-corrector:} & x_{t_{i+1}} = x_{t_i} + \Delta t_i \left[b_{i,0} u_{i,i}^\theta + \sum_{j=1}^{k_i} b_{i,j} u_{i,i-j}^\theta \right] + \sqrt{\gamma_i v_i}\,\xi_i. \end{cases} \tag{15}$$

Both stages start from x_{t_i} and use the same noise sample ξ_i. The corrector replaces the predictor's drift estimate without drawing additional noise. We jointly learn $a_{i,j}$, $b_{i,j}$, and γ_i to adapt the weights to each step's dynamics. Following numerical solvers (Atkinson et al., 2009), each coefficient group sums to one. We relax the remaining analytical constraints and normalize learned real-valued weights:

$$a_{i,j} = \frac{\widetilde{a}_{i,j}}{\sum_{\ell=1}^{k_i}\widetilde{a}_{i,\ell}}, \quad j = 1, \dots, k_i, \qquad b_{i,j} = \frac{\widetilde{b}_{i,j}}{\sum_{\ell=0}^{k_i}\widetilde{b}_{i,\ell}}, \quad j = 0, \dots, k_i. \tag{16}$$

4.3 TIME CALIBRATION

After a numerical update, the sample's noise level may differ from its scheduled time (Zhao et al., 2026). We estimate that level by variance matching, following EDM (Karras et al., 2022), and use it as the starting time of the next step. Thus, instead of updating from t_i to t_{i+1}, we update from the calibrated time t_i^{c} to t_{i+1}. The step size becomes $\Delta t_i = t_i^{\mathrm{c}} - t_{i+1}$, and the noise variance is computed over this same interval.

The predictor and corrector each produce a sample, with calibrated times t_{i+1}^{p} and t_{i+1}^{c}, respectively. We evaluate the network at the predictor's time and start the next step from the corrector's time.

Theorem 4.4 (Time calibration). *With time measured as noise standard deviation, assume the noise model in Appendix A.4 and set $\kappa_i = 1$. For $t_i^{\mathrm{c}} > t_{i+1} > 0$, set $\Delta t_i = t_i^{\mathrm{c}} - t_{i+1}$, $h_i = -(1+\gamma_i)\Delta t_i$, and $v_i = (t_i^{\mathrm{c}})^2 - t_{i+1}^2$. The calibrated times are (all sums run over $j = 1, \dots, k_i$)*

$$t_{i+1}^{\mathrm{p}} = \left[\left(t_i^{\mathrm{c}} + h_i \sum_j a_{i,j} \right)^2 + \gamma_i v_i \right]^{1/2}, \tag{17}$$

$$t_{i+1}^{\mathrm{c}} = \left\{ \left[t_i^{\mathrm{c}} + h_i \left(\sum_j b_{i,j} + b_{i,0} \frac{t_i^{\mathrm{c}} + h_i \sum_j a_{i,j}}{t_{i+1}^{\mathrm{p}}} \right) \right]^2 + \gamma_i v_i \left(1 + \frac{h_i b_{i,0}}{t_{i+1}^{\mathrm{p}}} \right)^2 \right\}^{1/2}. \tag{18}$$

Starting with $t_0^{\mathrm{c}} = t_0$, we repeat this calculation after each update. The formulas use only times and solver parameters; they do not require the clean sample. We also learn $\kappa_i > 0$ to adjust the network's time input to $\kappa_i t_{i+1}^{\mathrm{p}}$. This additional adjustment is separate from variance matching. When $\gamma_i = 0$, both calibrated times reduce to t_{i+1}.

4.4 DISTRIBUTION-MATCHING UPPER BOUND LOSS

At fixed NFE, we match the student's output law q_ϕ to q_{tea}, generated by a high-accuracy teacher with the same pretrained model:

$$\min_\phi W_2^2(q_\phi, q_{\mathrm{tea}}), \qquad \phi = \{\gamma_i, a_{i,j}, b_{i,j}, \kappa_i\}_{i=0}^{n-1}, \tag{19}$$

Evaluating W_2 requires optimizing over couplings. We instead construct a computable upper bound by sharing the initial noise:

$$x_0^{\mathrm{tea}} = \mathrm{solve}_{\mathrm{tea}}(x_{t_0}), \qquad x_0^{\phi} = \mathrm{solve}_{\phi}(x_{t_0}, \xi), \qquad x_{t_0} \sim p_T, \tag{20}$$

Here $\xi = (\xi_0, \dots, \xi_{n-1})$ contains independent standard Gaussian innovations, independent of x_{t_0}. The paired outputs define a coupling of q_ϕ and q_{tea}.

Theorem 4.5 (Distribution-matching upper bound). *If q_ϕ and q_{tea} have finite second moments, the endpoints in Eq. (20) satisfy*

$$W_2^2(q_\phi, q_{\mathrm{tea}}) \le \mathbb{E}_{x_{t_0}, \xi} \| x_0^\phi - x_0^{\mathrm{tea}} \|_2^2. \tag{21}$$

For latent models, we minimize this upper bound (the pixel models use Inception loss; its feature-space bound is given in Appendix A.5):

$$\min_\phi \mathcal{L}_2(\phi), \qquad \mathcal{L}_2(\phi) = \mathbb{E}_{x_{t_0}, \xi} \| x_0^\phi - x_0^{\mathrm{tea}} \|_2^2. \tag{22}$$

This loss supervises only the final output, leaving intermediate states free to depart from the teacher trajectory. The shared-noise coupling need not be optimal, so minimizing $\mathcal{L}_2$ is a surrogate for, rather than an exact evaluation of, the Wasserstein objective. The proof and the bound relative to p_{data} are in Appendix A.5.

Table 1: Image generation FID↓ on six benchmarks. Panels (a)–(c) and (e) use pixel-space models; (d) and (f) use latent-space models. NFE counts both conditional and unconditional evaluations for Stable Diffusion v1.5. For LD3, we report its best result across base solvers. Best results are **bold**; second-best results are underlined.

(a) CIFAR-10 32×32 (Krizhevsky & Hinton, 2009)

	Method	NFE			
		3	5	7	9
Handcrafted	iPNDM	24.55	7.77	4.04	2.83
	DPM-Solver++(3M)	55.76	9.94	4.29	2.99
	UniPC	109.60	23.98	5.83	3.21
Learnable	AMED-Solver	18.49	7.59	4.36	3.67
	AMED-Plugin	10.81	6.61	3.65	2.63
	EPD-Solver	10.40	4.33	2.82	2.49
	EPD-Plugin	10.54	4.47	3.27	2.42
	LD3	16.52	6.39	2.98	2.51
	DLMS	–	3.23	2.53	2.37
	S4S-Alt	16.95	3.73	2.52	2.31
	AdaSDE	12.62	4.18	2.88	2.56
	BézierFlow	22.20	5.26	3.29	2.65
	INDIS	9.26	3.31	2.60	2.28
	DyWeight	8.16	3.02	2.40	**2.13**
	Ours	**6.87**	**2.71**	**2.31**	2.20

(b) FFHQ 64×64 (Karras et al., 2019)

	Method	NFE			
		3	5	7	9
Handcrafted	iPNDM	27.72	13.80	7.16	4.98
	DPM-Solver++(3M)	66.07	13.47	6.20	4.77
	UniPC	86.43	21.40	7.44	4.47
Learnable	AMED-Solver	47.31	14.80	8.82	6.31
	AMED-Plugin	26.87	12.49	6.64	4.24
	EPD-Solver	21.74	7.84	4.81	3.82
	EPD-Plugin	19.02	7.97	5.09	3.53
	LD3	23.86	10.36	4.38	2.94
	DLMS	–	6.85	5.16	4.23
	S4S-Alt	19.86	6.25	3.45	3.00
	AdaSDE	23.80	8.05	5.11	4.19
	BézierFlow	33.72	13.71	4.28	3.68
	INDIS	17.72	6.91	3.90	3.25
	DyWeight	16.78	5.85	3.39	2.77
	Ours	**14.47**	**4.94**	**3.12**	**2.70**

(c) ImageNet 64×64 (Russakovsky et al., 2015)

	Method	NFE			
		3	5	7	9
Handcrafted	iPNDM	34.81	15.54	8.64	5.64
	DPM-Solver++(3M)	65.19	16.87	8.68	6.25
	UniPC	91.38	24.36	9.57	6.34
Learnable	AMED-Solver	38.10	10.74	6.66	5.44
	AMED-Plugin	28.06	13.83	7.81	5.60
	EPD-Solver	18.28	6.35	5.26	4.27
	EPD-Plugin	19.89	8.17	4.81	4.02
	LD3	27.82	11.55	5.63	4.71
	DLMS	–	7.16	6.31	4.57
	AdaSDE	18.51	6.90	5.26	4.59
	INDIS	18.96	7.28	4.94	3.17
	DyWeight	17.37	6.30	4.55	3.82
	Ours	**14.94**	**4.62**	**3.14**	**2.91**

(d) LSUN-Bedroom 256×256 (Yu et al., 2015)

	Method	NFE			
		3	5	7	9
Handcrafted	iPNDM	43.31	18.46	11.03	7.89
	DPM-Solver++(3M)	111.90	18.44	5.18	3.77
	UniPC	112.30	13.76	4.52	3.72
Learnable	AMED-Solver	58.21	13.20	7.10	5.65
	AMED-Plugin	101.5	25.68	8.63	7.82
	EPD-Solver	13.21	7.52	5.97	5.01
	EPD-Plugin	14.12	8.26	5.24	4.51
	LD3	14.62	5.93	4.16	3.98
	DLMS	–	5.44	3.99	3.70
	S4S-Alt	37.65	13.03	10.03	–
	AdaSDE	18.03	6.96	5.69	5.16
	INDIS	12.44	4.99	3.81	3.42
	DyWeight	9.82	4.97	3.87	3.45
	Ours	**9.44**	**4.08**	**3.31**	**3.18**

(e) AFHQv2 64×64 (Choi et al., 2020)

	Method	NFE			
		3	5	7	9
Handcrafted	iPNDM	15.53	5.58	3.19	2.48
	DPM-Solver++(3M)	35.05	10.63	4.47	3.44
	UniPC	60.89	13.01	5.07	4.46
Learnable	LD3	17.94	6.09	2.97	2.25
	S4S-Alt	14.71	3.89	2.56	2.18
	BézierFlow	26.31	7.00	3.25	2.57
	INDIS	10.15	3.48	**2.37**	2.26
	DyWeight	9.16	3.20	2.42	**2.13**
	Ours	**8.75**	**3.15**	2.40	2.19

(f) MS-COCO 512×512 (Lin et al., 2014)

	Method	NFE (1 step = 2 NFE)			
		8	12	16	20
Handcrafted	iPNDM(2M)	19.65	14.38	14.03	13.75
	DPM-Solver++(2M)	21.33	15.15	12.33	12.04
	UniPC(2M)	32.01	15.81	14.44	14.36
Learnable	AMED-Plugin	18.92	14.84	13.96	13.24
	EPD-Plugin	16.46	13.14	12.52	12.17
	AdaSDE	30.89	13.99	13.39	12.68
	DyWeight	14.92	11.82	11.75	11.54
	Ours	**14.52**	**11.44**	**11.00**	**10.96**

5 Experiments

We compare sample quality at fixed NFE for unconditional, class-conditional, and text-to-image generation with pretrained pixel- and latent-space models.

5.1 Setup

Models and datasets. Pixel-space tasks include unconditional CIFAR-10 (32×32) (Krizhevsky & Hinton, 2009), FFHQ (64×64) (Karras et al., 2019), and AFHQv2 (64×64) (Choi et al.,

Table 2: FLUX.1-dev results on MS-COCO (512×512). FID (↓) and CLIP (↑) are reported at 3, 5, and 7 NFEs; the best results are in **bold**.

	FID (↓)		
Method / NFE	**3**	**5**	**7**
DPM-Solver++(2M)	63.64	21.81	22.18
iPNDM(2M)	48.46	25.03	22.22
INDIS	44.30	24.83	22.69
DyWeight	42.69	21.02	20.69
Ours	**33.36**	**20.73**	**20.11**
	CLIP (↑)		
Method / NFE	**3**	**5**	**7**
DPM-Solver++(2M)	24.40	26.38	26.20
iPNDM(2M)	24.83	26.37	26.17
INDIS	25.69	26.35	26.12
DyWeight	25.43	26.73	26.49
Ours	**27.31**	**29.90**	**30.55**

Table 3: Component ablation on CIFAR-10. FID (↓) is reported at 3, 5, 7, and 9 NFEs; the best result in each column is in **bold**. We examine reverse dynamics, score timing, and predictor–corrector updates. The coefficient γ controls the matched score drift and noise variance, while κ_i scales the score-query time as $\kappa_i t_{i+1}^{\mathrm{p}}$.

	FID (↓)			
Config. / NFE	**3**	**5**	**7**	**9**
w/o γ	17.63	3.69	2.50	2.29
w/o time calibration	41.69	6.10	2.74	2.36
w/o κ_i	12.37	4.26	2.91	2.53
w/o predictor	13.71	5.64	3.08	2.50
w/o corrector	13.85	4.84	3.19	2.72
Ours	**6.87**	**2.71**	**2.31**	**2.20**



Figure 3: **Qualitative comparison at 5 NFEs.** Each dataset panel shows selected samples from DPM-Solver++(2M), iPNDM(2M), AMED-Solver, and RD-Solver. Within each column, all methods share the same initial noise and, for ImageNet-64, the same class label. The models generate 64×64 images on ImageNet-64, AFHQv2, and FFHQ, and 256×256 images on LSUN-Bedroom.

2020), and class-conditional ImageNet (64×64) (Russakovsky et al., 2015). Latent-space tasks include unconditional LSUN-Bedroom (256×256) (Yu et al., 2015) and MS-COCO text-to-image generation (512×512) (Lin et al., 2014) with Stable Diffusion v1.5 (Rombach et al., 2022) and FLUX.1-dev (Labs, 2024).

Baselines. Handcrafted baselines are iPNDM (Zhang & Chen, 2023; Liu et al., 2022), DPM-Solver++ (Lu et al., 2025), and UniPC (Zhao et al., 2023). Learned baselines include AMED-Solver/Plugin (Zhou et al., 2024), EPD-Solver/Plugin (Zhu et al., 2025), LD3 (Tong et al., 2025), DLMS (Liang et al., 2025), S4S-Alt (Frankel et al., 2025), AdaSDE (Wang et al., 2025a), DyWeight (Zhao et al., 2026), INDIS (Yuan et al., 2026), and BézierFlow (Min et al., 2026). INDIS learns instance-dependent timesteps; BézierFlow learns stochastic interpolant schedules using Bézier curves. We report LD3's best FID across base multistep solvers. FLUX baselines are DPM-Solver++(2M), iPNDM(2M), INDIS, and DyWeight. Sources and configurations are in Appendix D.

Optimization and evaluation. We fit one solver per NFE with a frozen network and shared teacher–student noise and conditioning. Cached pairs number 20k per pixel-space dataset and LSUN-Bedroom, and 5k per text-to-image model. Adam (Kingma & Ba, 2015) (AdamW (Loshchilov & Hutter, 2019) for FLUX) uses cosine decay. FID (Heusel et al., 2017) uses 50k samples for pixel models and LSUN-Bedroom, and 30k for MS-COCO; FLUX also reports CLIP (Radford et al., 2021). Teacher and training settings are in Appendix B.

5.2 Main results

RD-Solver achieves the lowest FID in 21 of 24 settings in Table 1. At 5 NFEs, FID is 26.7% lower than DyWeight's on ImageNet and 17.9% lower on LSUN-Bedroom. On ImageNet, it reaches 3.14 FID at 7 NFEs versus DyWeight's 3.82 at 9 NFEs. DyWeight leads on CIFAR-10 and AFHQv2 at 9 NFEs, and INDIS on AFHQv2 at 7 NFEs. At 5 NFEs, CIFAR-10 FID is 2.71 (INDIS: 3.31; BézierFlow: 5.26).

On FLUX.1-dev (Table 2), RD-Solver attains the lowest FID and highest CLIP at every tested budget. At 3 NFEs, it improves DyWeight's FID from 42.69 to 33.36 and CLIP from 25.43 to 27.31; FID gains narrow at 5 and 7 NFEs. Figure 1 uses the 7-NFE weights trained at 512×512 at multiple native aspect ratios; Figure 3 compares four solvers at 5 NFEs on four datasets. More samples are in Appendix E.

5.3 Ablation study

On CIFAR-10, the full solver achieves the lowest FID at 3, 5, 7, and 9 NFEs (Table 3).

Learning reverse dynamics. Removing γ raises FID from 6.87 to 17.63 at 3 NFEs and from 2.71 to 3.69 at 5 NFEs. The gaps narrow at 7 NFEs (2.31 to 2.50) and 9 NFEs (2.20 to 2.29), showing greater benefit at smaller NFEs.

Score timing and predictor–corrector updates. At 3 NFEs, removing calibration or κ_i raises FID from 6.87 to 41.69 or 12.37. At 5 NFEs, removing the predictor or corrector raises it to 5.64 or 4.84. Appendix C.1 tests initialization, time schedules, pair count, teacher choice, and history reuse.

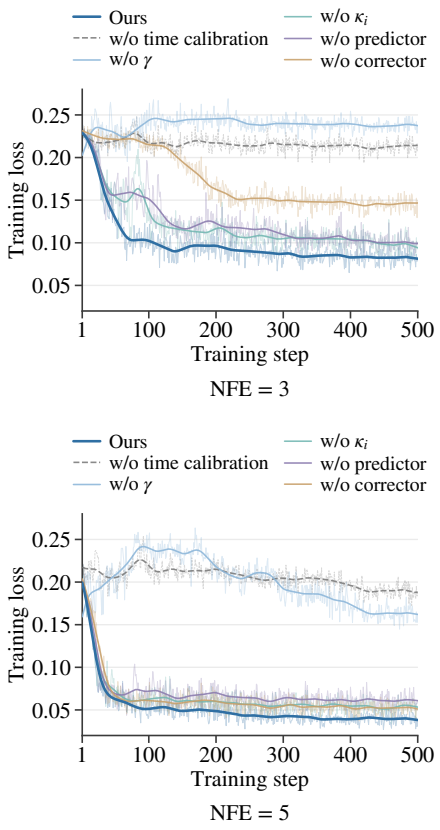

Figure 4: **Component ablation on CIFAR-10.** Top: 3 NFEs; bottom: 5 NFEs. Two-seed mean training loss over 500 steps, with raw (faint) and Gaussian-smoothed (dark) curves.

Training dynamics. Figure 4 compares six variants over 500 steps at 3 and 5 NFEs; the full solver reaches the lowest mean loss. In these curve experiments, the predictor and corrector ablations fix the respective weights at analytical initialization and train the remaining parameters.

6 Conclusion

We introduced **RD-Solver** for few-step diffusion sampling. The balance between distribution matching and score error motivates jointly learning reverse dynamics, multistep updates, and score-evaluation times. Experiments across diffusion and flow models show improved quality at low NFEs, supporting the choice to learn which reverse process to simulate alongside its discretization.

AI USE STATEMENT

Generative AI tools assisted with writing and polishing the manuscript and with retrieving relevant literature. This assistance also included revising mathematical statements and proofs and checking algebraic steps. The authors are responsible for the final text, references, mathematical arguments, and empirical claims.

REPRODUCIBILITY STATEMENT

The assumptions and proofs are given in Appendix A. Appendix B provides training and sampling algorithms, teacher configurations, losses, time schedules, and optimization settings. Appendix D distinguishes published baseline results from our evaluations and lists the reference implementations. Additional ablations appear in Appendix C.1. The training times marked with a dagger in Appendix C.2 are approximate wall-clock measurements.

REFERENCES

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. URL http://jmlr.org/papers/v26/23-1605.html.

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Kendall Atkinson, Weimin Han, and David E Stewart. *Numerical solution of ordinary differential equations*. John Wiley & Sons, 2009.

Yu Cao, Jingrun Chen, Yixin Luo, and Xiang Zhou. Exploring the optimal choice for generative processes in diffusion models: Ordinary vs stochastic differential equations. *Advances in Neural Information Processing Systems*, 36:33420–33468, 2023.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8185–8194. IEEE, 2020.

Arnak S Dalalyan and Avetik Karagulyan. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *Stochastic Processes and their Applications*, 129(12):5278–5311, 2019.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. GENIE: Higher-order denoising diffusion solvers. 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=LKEYuYNOqx.

Eric Frankel, Sitan Chen, Jerry Li, Pang Wei Koh, Lillian J. Ratliff, and Sewoong Oh. S4S: Solving for a Fast Diffusion Model Solver. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 17542–17574, 2025. URL https://mlanthology.org/icml/2025/frankel2025icml-s4s/.

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, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Jaivardhan Kapoor, Auguste Schulz, Julius Vetter, Felix Pei, Richard Gao, and Jakob H Macke. Latent diffusion for neural spiking data. *Advances in Neural Information Processing Systems*, 37: 118119–118154, 2024.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4396–4405. IEEE, 2019.

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

Sanghwan Kim, Hao Tang, and Fisher Yu. Distilling ode solvers of diffusion models into smaller steps. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9410–9419. IEEE, 2024.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL https://mlanthology.org/iclr/2015/kingma2015iclr-adam/.

Rainer Kress. *Numerical analysis*. Springer Science & Business Media, 2012.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Yuchen Liang, Xiangzhong Fang, Hanting Chen, and Yunhe Wang. Linear multistep solver distillation for fast sampling of diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vkOFOUDLTn.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 300–309. IEEE, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PlKWVd2yBkY.

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*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

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, 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. *Advances in neural information processing systems*, 35:5775–5787, 2022.

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, 2025.

Yunhong Min, Juil Koo, Seungwoo Yoo, and Minhyuk Sung. BézierFlow: Learning Bézier stochastic interpolant schedulers for few-step generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PCuDI32xhQ.

Aleksandr Oganov, Ilya Bykov, Eva Neudachina, Mishan Aliev, Alexander Tolmachev, Alexander Sidorov, Zuev Aleksandr, Andrey Okhotin, Denis Rakitin, and Aibek Alanov. GAS: Improving discretization of diffusion ODEs via generalized adversarial solver. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=R1l46h8kyM.

Soochul Park and Yeon Ju Lee. Dual-solver: A generalized ODE solver for diffusion models with dual prediction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=121IDiLVMt.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FjNys5c7VyY.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

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.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=nBGBzV4It3.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Bernardo P Schaeffer, Ricardo Rosa, and Glauco Valle. The effect of stochasticity in score-based diffusion sampling: a kl divergence analysis. *arXiv preprint arXiv:2506.11378*, 2025.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Neta Shaul, Juan Perez, Ricky T. Q. Chen, Ali Thabet, Albert Pumarola, and Yaron Lipman. Bespoke solvers for generative flow models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1PXEY7ofFX.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

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

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*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *International Conference on Machine Learning*, volume 202, pp. 32211–32252, 2023. URL https://mlanthology.org/icml/2023/song2023icml-consistency/.

Vinh Tong, Trung-Dung Hoang, Anji Liu, Guy Van den Broeck, and Mathias Niepert. Learning to discretize denoising diffusion odes. In *International Conference on Learning Representations*, volume 2025, pp. 47244–47282, 2025.

Ruoyu Wang, Beier Zhu, Junzhi Li, Liangyu Yuan, and Chi Zhang. Adaptive stochastic coefficients for accelerating diffusion sampling. In *Proc. NeurIPS*, 2025a.

Ruoyu Wang, Ziyu Li, Beier Zhu, Liangyu Yuan, Hanwang Zhang, Xun Yang, Xiaojun Chang, and Chi Zhang. Parallel diffusion solver via residual dirichlet policy optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Shuai Wang, Zexian Li, Qipeng zhang, Tianhui Song, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Differentiable solver search for fast diffusion sampling. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=oUApObDC1e.

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. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *BioRxiv*, pp. 2022–12, 2022.

Yilun Xu, Mingyang Deng, Xiang Cheng, Yonglong Tian, Ziming Liu, and Tommi Jaakkola. Restart sampling for improving generative processes. *Advances in Neural Information Processing Systems*, 36:76806–76838, 2023.

Shuchen Xue, Mingyang Yi, Weijian Luo, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhi-Ming Ma. Sa-solver: Stochastic adams solver for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:77632–77674, 2023.

Zilyu Ye, Zhiyang Chen, Tiancheng Li, Zemin Huang, Weijian Luo, and Guo-Jun Qi. Schedule on the fly: Diffusion time prediction for faster and better image generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23412–23422. IEEE, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623. IEEE, 2024b.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Liangyu Yuan, Ruoyu Wang, Tong Zhao, Dingwen Fu, Mingkun Lei, Beier Zhu, and Chi Zhang. Few-step diffusion sampling through instance-aware discretizations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35882–35892, June 2026.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

Tong Zhao, Mingkun Lei, Liangyu Yuan, Yanming Yang, Chenxi Song, Yang Wang, Beier Zhu, and Chi Zhang. Dyweight: Dynamic gradient weighting for few-step diffusion sampling. *arXiv preprint arXiv:2603.11607*, 2026.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36: 55502–55542, 2023.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7777–7786. IEEE, 2024.

Beier Zhu, Ruoyu Wang, Tong Zhao, Hanwang Zhang, and Chi Zhang. Distilling parallel gradients for fast ode solvers of diffusion models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19557–19566. IEEE, 2025.

APPENDIX

Appendix A collects the theoretical analysis. Appendix B gives teacher configurations, time schedules, losses, algorithms, and implementation details. Additional ablations and comparisons with distillation methods are collected in Appendix C; baseline details and visualizations are in Appendix D and Appendix E.

Contents

A **Theoretical Analysis** 16

A.1 Proof of Proposition 4.1 16

A.2 Proof of Theorem 4.2 16

A.3 Proof of Corollary 4.3 17

A.4 Proof of Theorem 4.4 19

A.5 Proof of Theorem 4.5 20

A.6 More Theoretical Results 21

A.6.1 A Gaussian example with an interior optimum 21

A.6.2 One-step contraction and local error 23

A.6.3 Endpoint KL under conditional invertibility 27

B **Experimental Details** 31

B.1 Complete Algorithms 31

B.1.1 Training 31

B.1.2 Sampling 32

B.2 Implementation Details 32

C **More Experimental Results** 35

C.1 More ablation 35

C.2 Comparison with Distillation Methods 37

D **Baseline Details** 39

E **Additional Visualizations** 40

E.1 Pixel-Space Diffusion Models 40

E.2 Latent-Space Diffusion Models 44

E.3 Text-to-Image Generation 45

A Theoretical Analysis

Density calculations are made at positive noise levels where the SDEs and their Fokker–Planck equations are well posed. The densities are positive and smooth; the displayed integrals are finite; differentiation under the integral is valid; and the boundary terms in integration by parts vanish. Statements at $t = 0$ hold when the corresponding limits exist. In particular, q_t inside a reverse interval denotes a continuous continuation; we do not assert that an arbitrary coarse numerical solver satisfies an exact Fokker–Planck equation.

A.1 Proof of Proposition 4.1

We verify that the reverse process has the same marginal density as the forward diffusion. The argument uses the score identity and uniqueness of the density evolution, rather than equality of individual sample paths.

Proof. The forward Fokker–Planck equation is

$$\partial_t p_t = -\nabla_x \cdot (f(t)x p_t) + \frac{g^2(t)}{2}\Delta_x p_t. \tag{A.1}$$

The exact score satisfies

$$p_t s_t^\star = \nabla_x p_t, \tag{A.2}$$

$$\nabla_x \cdot (p_t s_t^\star) = \Delta_x p_t. \tag{A.3}$$

For a candidate density ρ_t, the reverse dynamics in Eq. (5) give

$$-\partial_t \rho_t = \nabla_x \cdot \left(\left[f(t)x - \frac{1+\gamma(t)}{2} g^2(t) s_t^\star \right] \rho_t \right) + \frac{\gamma(t) g^2(t)}{2} \Delta_x \rho_t. \tag{A.4}$$

The positive diffusion term corresponds to evolution toward decreasing t. Substituting $\rho_t = p_t$ into its right-hand side gives

$$\begin{aligned}
&\nabla_x \cdot \left(\left[f(t)x - \frac{1+\gamma(t)}{2} g^2(t) s_t^\star \right] p_t \right) + \frac{\gamma(t) g^2(t)}{2} \Delta_x p_t \\
&= \nabla_x \cdot (f(t)x p_t) - \frac{1+\gamma(t)}{2} g^2(t) \nabla_x \cdot (p_t s_t^\star) + \frac{\gamma(t) g^2(t)}{2} \Delta_x p_t \\
&= \nabla_x \cdot (f(t)x p_t) - \frac{1+\gamma(t)-\gamma(t)}{2} g^2(t) \Delta_x p_t \\
&= \nabla_x \cdot (f(t)x p_t) - \frac{g^2(t)}{2} \Delta_x p_t \\
&= -\partial_t p_t.
\end{aligned} \tag{A.5}$$

Thus p_t solves the same reverse density equation as ρ_t. The terminal conditions agree, $\rho_T = p_T$. Uniqueness therefore implies $\rho_t = p_t$.

At $\gamma = 0$, the reverse process is

$$\mathrm{d}x_t = [f(t)x_t - \tfrac{1}{2} g^2(t) s_t^\star(x_t)]\, \mathrm{d}t. \tag{A.6}$$

At $\gamma = 1$, it is

$$\mathrm{d}x_t = [f(t)x_t - g^2(t) s_t^\star(x_t)]\, \mathrm{d}t + g(t)\, \mathrm{d}\bar{w}_t. \tag{A.7}$$

These are the PF-ODE and reverse SDE in Eq. (3). □

A.2 Proof of Theorem 4.2

The density q_t may start from any entrance law at t_i. This permits the analysis to begin after earlier numerical updates have already accumulated error, without identifying the subsequent continuous density with a discrete interpolation.

Proof. Recall the two fields defined in Eq. (6):

$$b_t(x) = f(t)x - \frac{g^2(t)}{2} s_t^\star(x), \tag{A.8}$$

$$e_t(x) = s_\theta(x,t) - s_t^\star(x). \tag{A.9}$$

Replacing the score in the reverse dynamics by s_θ yields

$$-\partial_t q_t = \nabla_x \cdot (f(t)x q_t) - \frac{1+\gamma(t)}{2} g^2(t) \nabla_x \cdot (q_t s_\theta) + \frac{\gamma(t) g^2(t)}{2} \Delta_x q_t. \tag{A.10}$$

Substitute $s_\theta = s_t^\star + e_t$ and separate the PF coefficient:

$$\begin{aligned} -\partial_t q_t &= \nabla_x \cdot (f(t)x q_t) - \frac{g^2(t)}{2} \nabla_x \cdot (q_t s_t^\star) - \frac{\gamma(t) g^2(t)}{2} \nabla_x \cdot (q_t s_t^\star) \\ &\quad - \frac{(1+\gamma(t)) g^2(t)}{2} \nabla_x \cdot (q_t e_t) + \frac{\gamma(t) g^2(t)}{2} \Delta_x q_t \\ &= \nabla_x \cdot (b_t q_t) + \frac{\gamma(t) g^2(t)}{2} [\Delta_x q_t - \nabla_x \cdot (q_t s_t^\star)] - \frac{(1+\gamma(t)) g^2(t)}{2} \nabla_x \cdot (q_t e_t). \end{aligned} \tag{A.11}$$

The expression in brackets is the relative-score flux:

$$\begin{aligned} \Delta_x q_t - \nabla_x \cdot (q_t s_t^\star) &= \nabla_x \cdot (\nabla_x q_t - q_t \nabla_x \log p_t) \\ &= \nabla_x \cdot \{ q_t [\nabla_x \log q_t - \nabla_x \log p_t] \} \\ &= \nabla_x \cdot \left(q_t \nabla_x \log \frac{q_t}{p_t} \right) \\ &= \nabla_x \cdot (q_t r_t). \end{aligned} \tag{A.12}$$

Consequently,

$$-\partial_t q_t = \nabla_x \cdot (b_t q_t) + \frac{\gamma(t) g^2(t)}{2} \nabla_x \cdot (q_t r_t) - \frac{(1+\gamma(t)) g^2(t)}{2} \nabla_x \cdot (q_t e_t), \tag{A.13}$$

which is Eq. (7). □

A.3 Proof of Corollary 4.3

We differentiate $\mathrm{KL}(q_t \| p_t)$ in the sampling direction and show explicitly that the common PF transport cancels. We follow the continuous-time relative-entropy analysis of Schaeffer et al. (2025).

Proof. Mass conservation and the relative-score definition give

$$\int q_t(x)\, \mathrm{d}x = 1, \tag{A.14}$$

$$\int (-\partial_t q_t(x))\, \mathrm{d}x = 0, \tag{A.15}$$

$$\nabla_x \left(\frac{q_t}{p_t} \right) = \frac{q_t}{p_t} r_t. \tag{A.16}$$

Differentiation under the integral yields

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{KL}(q_t \| p_t) &= \int (-\partial_t q_t) \left(1 + \log \frac{q_t}{p_t} \right) \mathrm{d}x - \int \frac{q_t}{p_t} (-\partial_t p_t)\, \mathrm{d}x \\ &= \int (-\partial_t q_t) \log \frac{q_t}{p_t}\, \mathrm{d}x - \int \frac{q_t}{p_t} (-\partial_t p_t)\, \mathrm{d}x. \end{aligned} \tag{A.17}$$

The paired density equations are

$$-\partial_t p_t = \nabla_x \cdot (b_t p_t), \tag{A.18}$$

$$-\partial_t q_t = \nabla_x \cdot (b_t q_t) + \frac{\gamma(t) g^2(t)}{2} \nabla_x \cdot (q_t r_t) - \frac{(1+\gamma(t)) g^2(t)}{2} \nabla_x \cdot (q_t e_t). \tag{A.19}$$

We evaluate their three contributions separately.

Common transport. Integration by parts gives

$$\int \nabla_x \cdot (b_t q_t) \log \frac{q_t}{p_t} \, \mathrm{d}x = -\int q_t b_t^\top r_t \, \mathrm{d}x, \tag{A.20}$$

$$-\int \frac{q_t}{p_t} \nabla_x \cdot (b_t p_t) \, \mathrm{d}x = \int p_t b_t^\top \nabla_x \left(\frac{q_t}{p_t} \right) \mathrm{d}x = \int q_t b_t^\top r_t \, \mathrm{d}x. \tag{A.21}$$

Hence the sum of the transport terms is zero:

$$\int \nabla_x \cdot (b_t q_t) \log \frac{q_t}{p_t} \, \mathrm{d}x - \int \frac{q_t}{p_t} \nabla_x \cdot (b_t p_t) \, \mathrm{d}x = 0. \tag{A.22}$$

Distribution matching. The relative-score term contributes

$$\int \nabla_x \cdot (q_t r_t) \log \frac{q_t}{p_t} \, \mathrm{d}x = -\int q_t r_t^\top \nabla_x \log \frac{q_t}{p_t} \, \mathrm{d}x = -\int q_t \|r_t\|_2^2 \, \mathrm{d}x. \tag{A.23}$$

Score error. The remaining term is

$$-\int \nabla_x \cdot (q_t e_t) \log \frac{q_t}{p_t} \, \mathrm{d}x = \int q_t e_t^\top \nabla_x \log \frac{q_t}{p_t} \, \mathrm{d}x = \int q_t e_t^\top r_t \, \mathrm{d}x. \tag{A.24}$$

Substituting these contributions into Eq. (A.17) gives

$$-\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{KL}(q_t \| p_t) = \frac{(1+\gamma(t))g^2(t)}{2} \int q_t e_t^\top r_t \, \mathrm{d}x - \frac{\gamma(t) g^2(t)}{2} \int q_t \|r_t\|_2^2 \, \mathrm{d}x. \tag{A.25}$$

This proves Eq. (8). When $e_t = 0$, it reduces to

$$-\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{KL}(q_t \| p_t) = -\frac{\gamma(t) g^2(t)}{2} \|r_t\|_{q_t}^2 \le 0. \tag{A.26}$$

A sufficient condition with imperfect scores. Pointwise Cauchy–Schwarz and then its integral version imply

$$\left| \int q_t e_t^\top r_t \, \mathrm{d}x \right| \le \int q_t \|e_t\|_2 \|r_t\|_2 \, \mathrm{d}x \le \left(\int q_t \|e_t\|_2^2 \, \mathrm{d}x \right)^{1/2} \left(\int q_t \|r_t\|_2^2 \, \mathrm{d}x \right)^{1/2} = \|e_t\|_{q_t} \|r_t\|_{q_t}. \tag{A.27}$$

Consequently,

$$-\frac{\mathrm{d}}{\mathrm{d}t} \mathrm{KL}(q_t \| p_t) \le \frac{g^2(t)}{2} \left[(1+\gamma(t)) \|e_t\|_{q_t} \|r_t\|_{q_t} - \gamma(t) \|r_t\|_{q_t}^2 \right] = \frac{g^2(t)}{2} \|r_t\|_{q_t} \left[(1+\gamma(t)) \|e_t\|_{q_t} - \gamma(t) \|r_t\|_{q_t} \right]. \tag{A.28}$$

The strict inequality in Eq. (10) implies $\gamma(t) > 0$ and $\|r_t\|_{q_t} > 0$. With $g(t) > 0$, the last display is strictly negative, proving the stated sufficient condition. □

The mismatch inherited by a reverse interval. For the continuous continuation from q_{t_i}, integration gives

$$\begin{aligned}
&\mathrm{KL}(q_{t_{i+1}}\|p_{t_{i+1}}) - \mathrm{KL}(q_{t_i}\|p_{t_i}) \\
&= \int_{t_{i+1}}^{t_i} \left[-\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{KL}(q_t\|p_t)\right] \mathrm{d}t \\
&= -\frac{\gamma_i}{2}\int_{t_{i+1}}^{t_i} g^2(t)\|r_t\|_{q_t}^2 \,\mathrm{d}t + \frac{1+\gamma_i}{2}\int_{t_{i+1}}^{t_i} g^2(t)\left(\int q_t e_t^\top r_t \,\mathrm{d}x\right)\mathrm{d}t.
\end{aligned} \tag{A.29}$$

For an exact score on this interval,

$$\mathrm{KL}(q_{t_{i+1}}\|p_{t_{i+1}}) = \mathrm{KL}(q_{t_i}\|p_{t_i}) - \frac{\gamma_i}{2}\int_{t_{i+1}}^{t_i} g^2(t)\|r_t\|_{q_t}^2 \,\mathrm{d}t. \tag{A.30}$$

Thus the ideal PF flow preserves the entrance KL mismatch, whereas stochastic reverse dynamics dissipate it whenever the last integral is positive. The endpoint in this identity is that of the continuous continuation, not an arbitrary numerical corrector. The imperfect-score condition is sufficient, not necessary, and does not determine a globally optimal schedule.

A.4 Proof of Theorem 4.4

Time represents the noise standard deviation. Following the variance normalization in EDM (Karras et al., 2022), Appendix B.6, we choose the calibrated time so that dividing the noise by that time gives unit variance. For a fixed clean sample x_0, this means

$$\mathrm{Var}\left(\frac{x - x_0}{\hat{t}}\,\middle|\, x_0\right) = 1 \quad\Longleftrightarrow\quad \hat{t}^2 = \mathrm{Var}(x - x_0 \mid x_0). \tag{A.31}$$

Variance is taken per coordinate. We apply this calculation to the predictor and then to the corrector.

For a step from t_i^{c} to t_{i+1}, where $t_i^{\mathrm{c}} > t_{i+1} > 0$, let

$$\Delta t_i = t_i^{\mathrm{c}} - t_{i+1}, \qquad h_i = -(1+\gamma_i)\Delta t_i, \qquad v_i = (t_i^{\mathrm{c}})^2 - t_{i+1}^2. \tag{A.32}$$

Here $f(t) = 0$ and $g^2(t) = 2t$, as in the EDM noise coordinate. For this calculation, assume that the stored noise predictions all equal the current sample's noise ϵ:

$$x_{t_i} - x_0 = t_i^{\mathrm{c}}\epsilon, \qquad \Delta t_i u_{i,i-j}^{\theta} = h_i\epsilon, \quad j = 1, \dots, k_i. \tag{A.33}$$

Assume also that the fresh denoising prediction returns x_0 and that $\kappa_i = 1$. Its drift increment is

$$\Delta t_i u_{i,i}^{\theta} = h_i \frac{\widetilde{x}_{t_{i+1}} - x_0}{t_{i+1}^{\mathrm{p}}}. \tag{A.34}$$

Conditional on x_0, ϵ and the added noise ξ_i are independent standard Gaussians. All scalar coefficients are fixed in this calculation.

Proof. **Predictor.** Substituting the assumed drift increments into Eq. (15) gives

$$\widetilde{x}_{t_{i+1}} - x_0 = \delta_i\epsilon + \sqrt{\gamma_i v_i}\,\xi_i, \qquad \delta_i = t_i^{\mathrm{c}} + h_i\sum_{j=1}^{k_i} a_{i,j}. \tag{A.35}$$

The noises are independent and have unit variance, so

$$1 = \mathrm{Var}\left(\frac{\widetilde{x}_{t_{i+1}} - x_0}{t_{i+1}^{\mathrm{p}}}\,\middle|\, x_0\right) = \frac{\delta_i^2 + \gamma_i v_i}{(t_{i+1}^{\mathrm{p}})^2}, \tag{A.36}$$

$$t_{i+1}^{\mathrm{p}} = \sqrt{\delta_i^2 + \gamma_i v_i}. \tag{A.37}$$

Substituting δ_i gives Eq. (17).

Corrector. The fresh drift uses the predictor above. The corrector also reuses the same added noise ξ_i, so we collect its two contributions before taking the variance:

$$\begin{aligned} x_{t_{i+1}} - x_0 &= t_i^{\mathrm{c}}\epsilon + h_i \sum_{j=1}^{k_i} b_{i,j}\epsilon + \frac{h_i b_{i,0}}{t_{i+1}^{\mathrm{p}}}\left(\delta_i\epsilon + \sqrt{\gamma_i v_i}\,\xi_i\right) + \sqrt{\gamma_i v_i}\,\xi_i \\ &= \left[t_i^{\mathrm{c}} + h_i\left(\sum_{j=1}^{k_i} b_{i,j} + b_{i,0}\frac{\delta_i}{t_{i+1}^{\mathrm{p}}}\right)\right]\epsilon + \sqrt{\gamma_i v_i}\left(1 + \frac{h_i b_{i,0}}{t_{i+1}^{\mathrm{p}}}\right)\xi_i. \end{aligned} \tag{A.38}$$

Requiring unit variance again yields

$$1 = \mathrm{Var}\left(\frac{x_{t_{i+1}} - x_0}{t_{i+1}^{\mathrm{c}}}\,\middle|\, x_0\right), \tag{A.39}$$

$$(t_{i+1}^{\mathrm{c}})^2 = \left[t_i^{\mathrm{c}} + h_i\left(\sum_{j=1}^{k_i} b_{i,j} + b_{i,0}\frac{\delta_i}{t_{i+1}^{\mathrm{p}}}\right)\right]^2 + \gamma_i v_i\left(1 + \frac{h_i b_{i,0}}{t_{i+1}^{\mathrm{p}}}\right)^2. \tag{A.40}$$

Taking the positive square root gives Eq. (18). □

With the normalized weights in Eq. (16), $\delta_i = t_{i+1} - \gamma_i \Delta t_i$. The predictor formula simplifies to

$$t_{i+1}^{\mathrm{p}} = \sqrt{t_{i+1}^2 + \gamma_i(1+\gamma_i)(\Delta t_i)^2}.$$

It does not depend on the individual predictor weights. The corrector formula uses $\sum_{j\geq 1} b_{i,j} = 1 - b_{i,0}$ and depends only on $b_{i,0}$ among the current weights. If $\gamma_i = 0$, both calibrated times equal t_{i+1}.

To use the result, start with $t_0^{\mathrm{c}} = t_0$. After each corrector, replace the next starting time t_{i+1} by t_{i+1}^{c}; the next step size is therefore $t_{i+1}^{\mathrm{c}} - t_{i+2}$. For stored predictions in Eq. (14), evaluate f, g at t_{m+1}^{p} and use the saved network output $s_\theta(\widetilde{x}_{t_{m+1}}, \kappa_m t_{m+1}^{\mathrm{p}})$. The learned multiplier κ_m further adjusts the network's time input.

The clean sample x_0 is only used in the derivation, not in the formulas used for sampling. The calculation is exact under the one-step assumptions above. During sampling it is an approximation: stored predictions need not share the same ϵ, and denoising need not recover x_0.

A.5 Proof of Theorem 4.5

The distributional objective is defined over all couplings. The paired training rollouts supply one admissible coupling, which is sufficient for an upper bound without estimating either endpoint density.

Proof. Let p_ω be the joint law of the initial noise and independent innovations:

$$\omega = (x_{t_0}, \xi). \tag{A.41}$$

The teacher ignores ξ. The endpoint laws are

$$q_\phi = \mathrm{law}(x_0^\phi(\omega)), \tag{A.42}$$

$$q_{\mathrm{tea}} = \mathrm{law}(x_0^{\mathrm{tea}}(\omega)). \tag{A.43}$$

Construct their joint pushforward:

$$\pi = (x_0^\phi, x_0^{\mathrm{tea}})_{\#} p_\omega. \tag{A.44}$$

For any measurable set $a \subseteq \mathbb{R}^d$, its first marginal is

$$\begin{aligned} \pi(a \times \mathbb{R}^d) &= \Pr(x_0^\phi \in a) \\ &= q_\phi(a). \end{aligned} \tag{A.45}$$

Its second marginal is

$$\begin{aligned} \pi(\mathbb{R}^d \times a) &= \Pr(x_0^{\mathrm{tea}} \in a) \\ &= q_{\mathrm{tea}}(a). \end{aligned} \tag{A.46}$$

Hence $\pi \in \Pi(q_\phi, q_{\mathrm{tea}})$, the set of admissible couplings. The Wasserstein definition now gives

$$\begin{aligned} W_2^2(q_\phi, q_{\mathrm{tea}}) &= \inf_{\widetilde{\pi} \in \Pi(q_\phi, q_{\mathrm{tea}})} \int \|x - x'\|_2^2 \, \mathrm{d}\widetilde{\pi}(x, x') \\ &\le \int \|x - x'\|_2^2 \, \mathrm{d}\pi(x, x') \\ &= \int \|x_0^\phi(\omega) - x_0^{\mathrm{tea}}(\omega)\|_2^2 \, \mathrm{d}p_\omega(\omega) \\ &= \mathbb{E}_{x_{t_0}, \xi} \|x_0^\phi - x_0^{\mathrm{tea}}\|_2^2 \\ &= \mathcal{L}_2(\phi). \end{aligned} \tag{A.47}$$

This is Eq. (21). □

The representation used by the training loss. For a measurable map H whose output laws have finite second moments, the pushforward $(H, H)_\# \pi$ is a coupling of $H_\# q_\phi$ and $H_\# q_{\mathrm{tea}}$. Hence

$$W_2^2(H_\# q_\phi, H_\# q_{\mathrm{tea}}) \le \mathbb{E}\|H(x_0^\phi) - H(x_0^{\mathrm{tea}})\|_2^2 = \mathcal{L}_H(\phi).$$

This proves the feature-space claim in Section 4.4. For pixel-space models, $H = h_{\mathrm{inc}}$ is the frozen Inception feature extractor. For latent-space models, H is the identity on latent states and gives $\mathcal{L}_2$. A general feature map need not preserve distances or distinguish all states. A state-space bound would follow, for example, from an additional inverse-stability condition $\|x - x'\|_2 \le C\|H(x) - H(x')\|_2$ on the relevant supports, in which case $\mathcal{L}_2 \le C^2 \mathcal{L}_H$. We do not assume this for Inception features. The state-space bounds below retain $\mathcal{L}_2$ and do not substitute the Inception loss for it. For latent models these bounds concern latent laws; transferring them to decoded images requires assumptions on the decoder.

Distance to the data distribution. If p_{data} also has finite second moments, taking square roots and using the triangle inequality gives

$$W_2(q_\phi, q_{\mathrm{tea}}) \le \mathcal{L}_2(\phi)^{1/2}, \tag{A.48}$$

$$\begin{aligned} W_2(q_\phi, p_{\mathrm{data}}) &\le W_2(q_\phi, q_{\mathrm{tea}}) + W_2(q_{\mathrm{tea}}, p_{\mathrm{data}}) \\ &\le \mathcal{L}_2(\phi)^{1/2} + W_2(q_{\mathrm{tea}}, p_{\mathrm{data}}). \end{aligned} \tag{A.49}$$

In particular, if $W_2(q_{\mathrm{tea}}, p_{\mathrm{data}}) \le \varepsilon_{\mathrm{tea}}$, then

$$W_2(q_\phi, p_{\mathrm{data}}) \le \mathcal{L}_2(\phi)^{1/2} + \varepsilon_{\mathrm{tea}}. \tag{A.50}$$

The shared-noise coupling need not be optimal, so the loss is an upper-bound surrogate for the Wasserstein objective. The result imposes no intermediate trajectory agreement, and the teacher error is not a lower bound on the student's attainable error.

A.6 More Theoretical Results

The main analysis identifies how reverse dynamics act on an inherited density mismatch. The following results examine three aspects of that mechanism. A Gaussian example shows when an interior stochasticity can exactly correct an endpoint variance. A one-step analysis separates contraction of entrance error from newly introduced score and integration errors. Finally, a conditional-invertibility argument explains which additional regularity is needed to obtain an endpoint KL bound, beyond the Wasserstein bound in Theorem 4.5. These analyses are supplementary; they impose no additional assumptions on the algorithm and define no alternative training losses.

A.6.1 A Gaussian example with an interior optimum

The KL identity explains a competition between distribution correction and score error, but does not by itself imply that the best stochasticity is strictly between the PF-ODE and reverse SDE. The following Gaussian construction gives a setting in which an interior choice can exactly match the endpoint. It retains both an imperfect entrance law and a misscaled score.

Consider a one-dimensional VE diffusion. To distinguish total marginal variance from the perturbation variance in Eq. (2), write

$$p_0 = \mathcal{N}(0, \zeta_0^2), \qquad \zeta_0 > 0, \tag{A.51}$$

$$\zeta_t^2 = \zeta_0^2 + \int_0^t g^2(r)\,\mathrm{d}r, \tag{A.52}$$

$$p_t = \mathcal{N}(0, \zeta_t^2). \tag{A.53}$$

The sampler starts from $q_T = \mathcal{N}(0, \beta\zeta_T^2)$, with $\beta > 0$, and uses

$$s_\theta(x,t) = -\frac{x}{\chi\zeta_t^2}, \qquad 0 < \chi < 1. \tag{A.54}$$

Here χ describes score misscaling, not the time calibration κ_i.

Theorem A.1 (Gaussian endpoint variance)**.** *For constant $\gamma \in [0,1]$, the reverse dynamics with the score in Eq. (A.54) produce*

$$q_0^\gamma = \mathcal{N}(0, \varrho(\gamma)\zeta_0^2), \tag{A.55}$$

where

$$\varrho(\gamma) = \frac{\gamma\chi}{1+\gamma-\chi} + \left(\beta - \frac{\gamma\chi}{1+\gamma-\chi}\right)\left(\frac{\zeta_0^2}{\zeta_T^2}\right)^{(1+\gamma)/\chi - 1}. \tag{A.56}$$

If $(\varrho(0)-1)(\varrho(1)-1) < 0$, there exists $\gamma^\star \in (0,1)$ such that $q_0^{\gamma^\star} = p_0$.

Proof. The reverse SDE has linear drift and additive Gaussian noise, so it remains centered Gaussian. Denote its variance by $\widehat{\zeta}_t^2$. The reverse second-moment equation is

$$-\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\zeta}_t^2 = -\frac{(1+\gamma)g^2(t)}{\chi\zeta_t^2}\widehat{\zeta}_t^2 + \gamma g^2(t). \tag{A.57}$$

Set $c_\gamma = (1+\gamma)/\chi > 1$. Since $\mathrm{d}\zeta_t^2/\mathrm{d}t = g^2(t)$, the integrating factor gives

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left[\widehat{\zeta}_t^2(\zeta_t^2)^{-c_\gamma}\right] &= (\zeta_t^2)^{-c_\gamma}\left[\frac{\mathrm{d}\widehat{\zeta}_t^2}{\mathrm{d}t} - \frac{c_\gamma g^2(t)}{\zeta_t^2}\widehat{\zeta}_t^2\right] \\ &= -\gamma g^2(t)(\zeta_t^2)^{-c_\gamma}.\end{aligned} \tag{A.58}$$

Integrate from the entrance time T to t:

$$\begin{aligned}\widehat{\zeta}_t^2(\zeta_t^2)^{-c_\gamma} - \beta(\zeta_T^2)^{1-c_\gamma} &= -\gamma\int_T^t g^2(r)(\zeta_r^2)^{-c_\gamma}\,\mathrm{d}r \\ &= \frac{\gamma}{c_\gamma - 1}\left[(\zeta_t^2)^{1-c_\gamma} - (\zeta_T^2)^{1-c_\gamma}\right].\end{aligned} \tag{A.59}$$

Multiplying by $(\zeta_t^2)^{c_\gamma}$ gives

$$\widehat{\zeta}_t^2 = \frac{\gamma}{c_\gamma - 1}\zeta_t^2 + \left(\beta - \frac{\gamma}{c_\gamma - 1}\right)(\zeta_T^2)^{1-c_\gamma}(\zeta_t^2)^{c_\gamma}. \tag{A.60}$$

At $t = 0$, divide by ζ_0^2 and use

$$\frac{\gamma}{c_\gamma - 1} = \frac{\gamma\chi}{1+\gamma-\chi}. \tag{A.61}$$

This yields Eq. (A.56).

The denominator $1+\gamma-\chi$ is positive on $[0,1]$, so ϱ is continuous there. Under the strict sign condition, the intermediate value theorem gives $\varrho(\gamma^\star) = 1$ for some $\gamma^\star \in (0,1)$. Both endpoint laws are centered Gaussian; equality of variance therefore gives $q_0^{\gamma^\star} = p_0$. □

Interpretation. For this example, the endpoint KL is explicit:

$$\mathrm{KL}(q_0^\gamma \| p_0) = \frac{1}{2}\left[\varrho(\gamma) - 1 - \log \varrho(\gamma)\right]. \tag{A.62}$$

When the theorem's sign condition holds, both standard endpoint choices have positive KL, whereas the interior choice has zero KL. This is a conditional existence example, not a statement that every score error admits such an interior optimum. It concerns exact continuous integration; it does not establish the same optimum for an arbitrary finite-step discretization.

A.6.2 One-step contraction and local error

We bound the one-step Wasserstein error of the frozen-score reverse dynamics by separating propagated entrance error from local score and discretization errors (Dalalyan & Karagulyan, 2019; Xu et al., 2023). The bound quantifies how stochasticity affects both error propagation and the error introduced within a step.

Notation and assumptions. Use the interval $[t, t+\Delta t]$, with $t = t_{i+1} > 0$ and $t + \Delta t = t_i$. In the VE coordinate, $f(r) = 0$ and $g^2(r) = 2r$. Freezing the score at the entrance and integrating scalar coefficients gives

$$\bar{x}_t = x_{t+\Delta t} + \frac{1+\gamma}{2}\big((t+\Delta t)^2 - t^2\big) s_\theta(x_{t+\Delta t}, t+\Delta t) + \sqrt{\gamma\big((t+\Delta t)^2 - t^2\big)}\,\epsilon, \tag{A.63}$$

where $\epsilon \sim \mathcal{N}(0, I_d)$ is independent of the entrance and $\gamma \in [0,1]$ is the main-text diffusion-strength coefficient. Let $p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma}$ be the numerical entrance law and $p_t^{\mathrm{RD}_\theta,\gamma}$ its output law. Let $\widehat{p}_t$ be the law of the same update initialized from the true entrance $p_{t+\Delta t}$. Exact reverse dynamics from $p_{t+\Delta t}$ give p_t. Assume well-posed exact dynamics and finite first moments.

For all relevant x, y and $r, s \in [t, t+\Delta t]$, assume

$$\|r\nabla \log p_r(x) - r s_\theta(x, r)\|_2 \le \epsilon_t, \tag{A.64}$$
$$\|r s_\theta(x, r) - r s_\theta(x, s)\|_2 \le L_0 |r - s|, \tag{A.65}$$
$$\|r s_\theta(x, s)\|_2 \le L_1, \tag{A.66}$$
$$\|r s_\theta(x, s) - r s_\theta(y, s)\|_2 \le L_2 \|x - y\|_2. \tag{A.67}$$

The constants $L_0, L_1, L_2, \epsilon_t$ are nonnegative, and ϵ_t is uniform on this interval. The two-time form of the last two conditions accounts for the varying scalar r and frozen score time in Eq. (A.63). The spatial norm is Euclidean.

Local numerical error. A synchronous coupling bounds the local error of Eq. (A.63), including the within-step Brownian displacement.

Lemma A.2 (Local numerical error from the exact entrance law)**.** *Under Eqs. (A.64) to (A.67),*

$$W_1(\widehat{p}_t, p_t) \le e^{(1+\gamma)L_2\Delta t}(1+\gamma)\Big[\Delta t\big((1+\gamma)L_2 L_1 + L_0\big) + L_2\sqrt{2d\gamma(t+\Delta t)\Delta t} + \epsilon_t\Big]\Delta t. \tag{A.68}$$

Proof. **Step 1: time-reversed processes.** Write $x_s^{\leftarrow} = x_{t+\Delta t-s}$ and $\bar{x}_s^{\leftarrow} = \bar{x}_{t+\Delta t-s}$ for $s \in [0, \Delta t]$. Couple the exact and frozen-score paths with the same entrance $x_0^{\leftarrow} = \bar{x}_0^{\leftarrow} \sim p_{t+\Delta t}$ and Brownian motion w_s:

$$x_s^{\leftarrow} = x_0^{\leftarrow} + (1+\gamma)\int_0^s (t+\Delta t-u)\nabla \log p_{t+\Delta t-u}(x_u^{\leftarrow})\,\mathrm{d}u + \sqrt{2\gamma}\int_0^s \sqrt{t+\Delta t-u}\,\mathrm{d}w_u, \tag{A.69}$$

$$\bar{x}_s^{\leftarrow} = x_0^{\leftarrow} + (1+\gamma)\int_0^s (t+\Delta t-u) s_\theta(x_0^{\leftarrow}, t+\Delta t)\,\mathrm{d}u + \sqrt{2\gamma}\int_0^s \sqrt{t+\Delta t-u}\,\mathrm{d}w_u. \tag{A.70}$$

At $s = \Delta t$ these have laws p_t and $\widehat{p}_t$. The second path has exactly the law in Eq. (A.63), since $\int_0^{\Delta t}(t + \Delta t - u)\,\mathrm{d}u = ((t + \Delta t)^2 - t^2)/2$.

Step 2: error dynamics. Use the error notation $e(s) = \mathbb{E}\|x_s^{\leftarrow} - \bar{x}_s^{\leftarrow}\|_2$. For $r = t + \Delta t - u$, split the drift discrepancy into

$$\begin{aligned} & r\nabla \log p_r(x_u^{\leftarrow}) - rs_\theta(x_0^{\leftarrow}, t + \Delta t) \\ &= \big[r\nabla \log p_r(x_u^{\leftarrow}) - rs_\theta(x_u^{\leftarrow}, r)\big] \\ &\quad + \big[rs_\theta(x_u^{\leftarrow}, r) - rs_\theta(\bar{x}_u^{\leftarrow}, r)\big] \\ &\quad + \big[rs_\theta(\bar{x}_u^{\leftarrow}, r) - rs_\theta(x_0^{\leftarrow}, t + \Delta t)\big]. \end{aligned} \tag{A.71}$$

The first term is bounded by ϵ_t, and the expected norm of the second by $L_2 e(u)$. The third is the within-step discretization term.

Step 3: temporal discretization and displacement. Insert $rs_\theta(x_0^{\leftarrow}, r)$ in the last difference:

$$\begin{aligned} & \mathbb{E}\|rs_\theta(\bar{x}_u^{\leftarrow}, r) - rs_\theta(x_0^{\leftarrow}, t + \Delta t)\|_2 \\ &\quad \le L_2 \mathbb{E}\|\bar{x}_u^{\leftarrow} - x_0^{\leftarrow}\|_2 + L_0 \Delta t. \end{aligned} \tag{A.72}$$

The drift displacement is at most $(1 + \gamma)L_1 \Delta t$. Jensen's inequality and Itô isometry bound the stochastic displacement:

$$\begin{aligned} \mathbb{E}\left\| \sqrt{2\gamma} \int_0^u \sqrt{t + \Delta t - v}\,\mathrm{d}w_v \right\|_2 &\le \left(2d\gamma \int_0^u (t + \Delta t - v)\,\mathrm{d}v \right)^{1/2} \\ &\le \sqrt{2d\gamma(t + \Delta t)\Delta t}. \end{aligned} \tag{A.73}$$

Thus

$$\mathbb{E}\|\bar{x}_u^{\leftarrow} - x_0^{\leftarrow}\|_2 \le (1 + \gamma)L_1 \Delta t + \sqrt{2d\gamma(t + \Delta t)\Delta t}. \tag{A.74}$$

Step 4: composite error inequality. The shared Brownian terms cancel in the path difference. Hence

$$\begin{aligned} e(s) \le (1 + \gamma) \int_0^s \Big[& L_2 e(u) + \epsilon_t + L_0 \Delta t + (1 + \gamma)L_2 L_1 \Delta t \\ & + L_2 \sqrt{2d\gamma(t + \Delta t)\Delta t} \Big]\,\mathrm{d}u. \end{aligned} \tag{A.75}$$

Step 5: Grönwall's inequality. Since $e(0) = 0$, the integral form of Grönwall's inequality gives

$$\begin{aligned} e(s) \le e^{(1+\gamma)L_2 s}(1 + \gamma) \Big[& \Delta t\big((1 + \gamma)L_2 L_1 + L_0\big) \\ & + L_2 \sqrt{2d\gamma(t + \Delta t)\Delta t} + \epsilon_t \Big] s. \end{aligned} \tag{A.76}$$

This also holds at $L_2 = 0$, without differentiating a norm at zero. At $s = \Delta t$, the synchronized endpoint pair is an admissible coupling, so $W_1(\widehat{p}_t, p_t) \le e(\Delta t)$. □

Gaussian overlap and propagated entrance error. Gaussian overlap controls propagation of entrance error (Xu et al., 2023). Let $Q(r) = \Pr(a \ge r)$ for $a \sim \mathcal{N}(0, 1)$ and $\mathrm{TV}(\mu, \nu) = \sup_A |\mu(A) - \nu(A)|$. For fixed $t, \Delta t$ and $B > 0$, define

$$\lambda(\gamma) = \begin{cases} 2Q\left(\dfrac{B\left[1 + \dfrac{(1 + \gamma)((t + \Delta t)^2 - t^2)L_2}{2(t + \Delta t)}\right]}{2\sqrt{\gamma((t + \Delta t)^2 - t^2)}} \right), & \gamma > 0, \\ 0, & \gamma = 0. \end{cases} \tag{A.77}$$

This is a lower bound on the coupling's merging probability; its argument uses the mean difference and noise variance of Eq. (A.63).

Lemma A.3 (TV distance between Gaussian perturbations). *For $x, y \in \mathbb{R}^d$ and $\sigma > 0$,*

$$\mathrm{TV}\big(\mathcal{N}(x, \sigma^2 I_d), \mathcal{N}(y, \sigma^2 I_d)\big) = 1 - 2Q\left(\frac{\|x - y\|_2}{2\sigma} \right). \tag{A.78}$$

Proof. Translate and rotate so the means differ only in the first coordinate, by $\|x-y\|_2$. Integrating out the other coordinates reduces the comparison to two one-dimensional Gaussian densities. They cross at their midpoint; each contributes mass $Q(\|x-y\|_2/(2\sigma))$ to their overlap. Total variation is one minus the overlap, proving the identity. A maximal coupling therefore merges with probability $2Q(\|x-y\|_2/(2\sigma))$. □

Lemma A.4 (Propagation of entrance error)**.** *Under Eq. (A.67),*

$$\begin{aligned}\mathrm{TV}(p_t^{\mathrm{RD}_\theta,\gamma},\widehat{p}_t) \le (1-\lambda(\gamma))\mathrm{TV}(p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma},p_{t+\Delta t}) \\ + \lambda(\gamma)\left[p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma}(\|x\|_2 > B/2) + p_{t+\Delta t}(\|x\|_2 > B/2)\right]. \end{aligned} \tag{A.79}$$

If both entrance laws satisfy $\|x\|_2 \le B/2$, the second line vanishes.

Proof. For entrance points with $\|x\|_2, \|y\|_2 \le B/2$, the conditional mean difference satisfies

$$\begin{aligned}&\left\| x - y + \frac{1+\gamma}{2}\big((t+\Delta t)^2 - t^2\big)[s_\theta(x,t+\Delta t) - s_\theta(y,t+\Delta t)]\right\|_2 \\ &\le \left[1 + \frac{(1+\gamma)((t+\Delta t)^2-t^2)L_2}{2(t+\Delta t)}\right]\|x-y\|_2 \\ &\le B\left[1 + \frac{(1+\gamma)((t+\Delta t)^2-t^2)L_2}{2(t+\Delta t)}\right]. \end{aligned} \tag{A.80}$$

For $\gamma > 0$, the kernels have covariance $\gamma((t+\Delta t)^2 - t^2)I_d$. By Lemma A.3, their maximal-coupling mismatch probability is at most $1-\lambda(\gamma)$.

Take a maximal entrance coupling. For identical inputs use identical noise; otherwise maximally couple the conditional Gaussian kernels. When both inputs lie in the radius-$B/2$ ball, mismatch contracts by $1-\lambda(\gamma)$; outside this event use the bound one. The union bound on the two entrance tail probabilities gives Eq. (A.79). At $\gamma = 0$, data processing gives the same statement with $\lambda(0) = 0$. □

Combining the two error terms. On the ball $\|x\|_2 \le B/2$, the bounded-diameter inequality relates TV to W_1. For the untruncated Gaussian step, the corresponding conversion also requires the entrance tail probabilities and output tail integrals below.

Proposition A.5 (One-step reverse-dynamics error bound)**.** *Under the stated regularity and moment assumptions, for every $B > 0$ and $\gamma \in (0,1]$,*

$$\begin{aligned}W_1(p_t^{\mathrm{RD}_\theta,\gamma},p_t) \le B\left[1 - 2Q\left(\frac{B\left[1+\frac{(1+\gamma)((t+\Delta t)^2-t^2)L_2}{2(t+\Delta t)}\right]}{2\sqrt{\gamma((t+\Delta t)^2-t^2)}}\right)\right] \\ \cdot \mathrm{TV}(p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma},p_{t+\Delta t}) \\ + e^{(1+\gamma)L_2\Delta t}(1+\gamma)\Big[\Delta t\big((1+\gamma)L_2L_1 + L_0\big) \\ + L_2\sqrt{2d\gamma(t+\Delta t)\Delta t} + \epsilon_t\Big]\Delta t \\ + 2BQ\left(\frac{B\left[1+\frac{(1+\gamma)((t+\Delta t)^2-t^2)L_2}{2(t+\Delta t)}\right]}{2\sqrt{\gamma((t+\Delta t)^2-t^2)}}\right) \\ \cdot \left[p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma}(\|x\|_2 > B/2) + p_{t+\Delta t}(\|x\|_2 > B/2)\right] \\ + \int_{\mathbb{R}^d}(\|x\|_2 - B/2)_+ \,\mathrm{d}p_t^{\mathrm{RD}_\theta,\gamma}(x) + \int_{\mathbb{R}^d}(\|x\|_2 - B/2)_+ \,\mathrm{d}\widehat{p}_t(x). \end{aligned} \tag{A.81}$$

The first two terms are propagated entrance error and local numerical error. The remaining terms account for entrance and output tails. At $\gamma = 0$, interpret both displayed Q terms by their limit zero.

Proof. **Step 1: split the total error.** Introduce the numerical output from the true entrance and apply the triangle inequality:

$$W_1(p_t^{\mathrm{RD}_\theta,\gamma}, p_t) \le W_1(p_t^{\mathrm{RD}_\theta,\gamma}, \widehat{p}_t) + W_1(\widehat{p}_t, p_t). \tag{A.82}$$

Step 2: bound propagated entrance error. For any finite-first-moment laws μ, ν, a maximal coupling (x, y) satisfies $\Pr(x \neq y) = \mathrm{TV}(\mu, \nu)$. Since $\|x-y\|_2 \le B + (\|x\|_2 - B/2)_+ + (\|y\|_2 - B/2)_+$,

$$\begin{aligned} W_1(\mu,\nu) &\le \mathbb{E}[\|x-y\|_2 \mathbf{1}_{\{x\neq y\}}] \\ &\le B\,\mathrm{TV}(\mu,\nu) + \int_{\mathbb{R}^d} (\|x\|_2 - B/2)_+ \,\mathrm{d}\mu(x) + \int_{\mathbb{R}^d} (\|x\|_2 - B/2)_+ \,\mathrm{d}\nu(x). \end{aligned} \tag{A.83}$$

This reduces to $W_1 \le B\,\mathrm{TV}$ when both laws are supported in the radius-$B/2$ ball. Applying it to the two numerical outputs, then using Lemma A.4, gives

$$\begin{aligned} W_1(p_t^{\mathrm{RD}_\theta,\gamma}, \widehat{p}_t) \le\ & B(1-\lambda(\gamma))\,\mathrm{TV}(p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma}, p_{t+\Delta t}) \\ & + B\lambda(\gamma)\left[p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma}(\|x\|_2 > B/2) + p_{t+\Delta t}(\|x\|_2 > B/2)\right] \\ & + \int_{\mathbb{R}^d} (\|x\|_2 - B/2)_+ \,\mathrm{d}p_t^{\mathrm{RD}_\theta,\gamma}(x) + \int_{\mathbb{R}^d} (\|x\|_2 - B/2)_+ \,\mathrm{d}\widehat{p}_t(x). \end{aligned} \tag{A.84}$$

Step 3: add local numerical error. The second term in Eq. (A.82) is bounded by Lemma A.2:

$$\begin{aligned} W_1(\widehat{p}_t, p_t) \le e^{(1+\gamma)L_2\Delta t}(1+\gamma)\Big[\Delta t\big((1+\gamma)L_2L_1 + L_0\big) \\ + L_2\sqrt{2d\gamma(t+\Delta t)\Delta t} + \epsilon_t\Big]\Delta t. \end{aligned} \tag{A.85}$$

Combining these two estimates and inserting the expression for $\lambda(\gamma)$ from Eq. (A.77) proves Eq. (A.81). □

Two-term form under the bounded-state hypothesis. If the two entrance laws and the two compared numerical output laws satisfy $\|x\|_2 \le B/2$, all tail terms vanish and the bound becomes

$$\begin{aligned} W_1(p_t^{\mathrm{RD}_\theta,\gamma}, p_t) \le\ & \underbrace{B\left[1 - 2Q\left(\frac{B\left[1 + \frac{(1+\gamma)((t+\Delta t)^2 - t^2)L_2}{2(t+\Delta t)}\right]}{2\sqrt{\gamma((t+\Delta t)^2 - t^2)}}\right)\right] \cdot \mathrm{TV}(p_{t+\Delta t}^{\mathrm{RD}_\theta,\gamma}, p_{t+\Delta t})}_{\text{propagated entrance error bound}} \\ & + \underbrace{e^{(1+\gamma)L_2\Delta t}(1+\gamma)\Big[\Delta t\big((1+\gamma)L_2L_1 + L_0\big) + L_2\sqrt{2d\gamma(t+\Delta t)\Delta t} + \epsilon_t\Big]\Delta t}_{\text{local score and discretization error bound}}. \end{aligned} \tag{A.86}$$

This conditional form requires the stated boundedness assumption; untruncated Gaussian outputs at $\gamma > 0$ do not satisfy it. For those outputs the applicable statement is Eq. (A.81). Clipping or truncating the sampler would change its kernel and is not assumed here.

The two standard endpoints. At $\gamma = 0$, $\lambda(0) = 0$ and the local bound reduces to

$$W_1(\widehat{p}_t, p_t) \le e^{L_2\Delta t}\big[\Delta t(L_2L_1 + L_0) + \epsilon_t\big]\Delta t. \tag{A.87}$$

At $\gamma = 1$, it becomes

$$W_1(\widehat{p}_t, p_t) \le 2e^{2L_2\Delta t}\Big[\Delta t(2L_2L_1 + L_0) + L_2\sqrt{2d(t+\Delta t)\Delta t} + \epsilon_t\Big]\Delta t. \tag{A.88}$$

In these two displays, $\widehat{p}_t$ is the output of the numerical step at the specified γ, always initialized from $p_{t+\Delta t}$.

Interpretation. For the reverse-dynamics step, stochasticity creates Gaussian overlap but also scales the drift error by $1 + \gamma$ and introduces the displayed Brownian displacement. The coefficients follow from this step's dynamics. Comparing these upper bounds alone does not order the actual sampler errors or determine an optimal γ.

A.6.3 Endpoint KL under conditional invertibility

The main-text coupling bound needs only finite moments. A KL bound requires more control: small endpoint displacement alone does not control the local volume change of a transport map. The following result makes those extra conditions explicit. Its direction is $\mathrm{KL}(q_{\mathrm{tea}} \| q_\phi)$, which is different from the continuous $\mathrm{KL}(q_t \| p_t)$ studied above. Here $\mathcal{L}_2$ denotes the state-space endpoint discrepancy in Eq. (21); it is not the Inception-feature loss used for pixel-space training.

Endpoint maps and additional assumptions. For this result, assume the common initialization is exactly Gaussian:

$$p_T = \mathcal{N}(0, \sigma_{\mathrm{init}}^2 I), \qquad \sigma_{\mathrm{init}} > 0. \tag{A.89}$$

This is an additional hypothesis for the result, not an assertion that every forward marginal at finite terminal time is exactly Gaussian. For fixed student innovations ξ, denote the two endpoint maps by

$$h_{\mathrm{tea}}(x_{t_0}) = x_0^{\mathrm{tea}}(x_{t_0}), \tag{A.90}$$

$$h_{\phi,\xi}(x_{t_0}) = x_0^{\phi}(x_{t_0}, \xi). \tag{A.91}$$

Assume the following conditions.

Assumption A.6 (Conditional endpoint invertibility)**.** The teacher map and, for almost every ξ, the conditional student map are orientation-preserving C^1 diffeomorphisms of $\mathbb{R}^d$.

Assumption A.7 (Stable student inverse)**.** There is $\ell_{\mathrm{inv}} < \infty$, uniform over ξ, such that

$$\|h_{\phi,\xi}^{-1}(x) - h_{\phi,\xi}^{-1}(x')\|_2 \le \ell_{\mathrm{inv}} \|x - x'\|_2. \tag{A.92}$$

Assumption A.8 (Small relative distortion)**.** Define the relative map, its displacement, and its Jacobian deviation by

$$a_\xi = h_{\phi,\xi}^{-1} \circ h_{\mathrm{tea}}, \tag{A.93}$$

$$d_\xi(x) = a_\xi(x) - x, \tag{A.94}$$

$$m_\xi(x) = \partial_x d_\xi(x). \tag{A.95}$$

For some $0 \le \delta < 1$, assume $\|m_\xi(x)\|_{\mathrm{op}} \le \delta$ almost everywhere.

Assumption A.9 (Differentiability and integrability)**.** All displayed expectations and derivatives are finite. The displacement d_ξ admits Gaussian integration by parts, and $x_0^\phi(x_{t_0}, \cdot)$ belongs to the Gaussian Sobolev space $W^{1,2}$ for almost every x_{t_0}.

Define the continuous scalar function

$$\psi(\delta) = \begin{cases} \dfrac{-\log(1-\delta) - \delta}{\delta^2}, & 0 < \delta < 1, \\ \dfrac{1}{2}, & \delta = 0. \end{cases} \tag{A.96}$$

The conditional mean endpoint is

$$\bar{x}_0^\phi(x_{t_0}) = \mathbb{E}_\xi[x_0^\phi(x_{t_0}, \xi)]. \tag{A.97}$$

Theorem A.10 (Conditional endpoint KL bound)**.** *Under the preceding assumptions,*

$$\mathrm{KL}(q_{\mathrm{tea}}\|q_\phi) \le \frac{\ell_{\mathrm{inv}}^2}{2\sigma_{\mathrm{init}}^2}\mathcal{L}_2(\phi) + \psi(\delta)\mathbb{E}_{x_{t_0},\xi}\|m_\xi(x_{t_0})\|_{\mathrm{F}}^2. \tag{A.98}$$

More explicitly, the same KL is bounded by

$$\begin{aligned}\mathrm{KL}(q_{\mathrm{tea}}\|q_\phi) \le{}& \frac{\ell_{\mathrm{inv}}^2}{2\sigma_{\mathrm{init}}^2}\mathbb{E}_{x_{t_0}}\|\bar{x}_0^\phi(x_{t_0}) - x_0^{\mathrm{tea}}(x_{t_0})\|_2^2 \\ &+ \frac{\ell_{\mathrm{inv}}^2}{2\sigma_{\mathrm{init}}^2}\mathbb{E}_{x_{t_0},\xi}\sum_{i=0}^{n-1}\|\partial_{\xi_i}x_0^\phi(x_{t_0},\xi)\|_{\mathrm{F}}^2 \\ &+ \psi(\delta)\mathbb{E}_{x_{t_0},\xi}\|m_\xi(x_{t_0})\|_{\mathrm{F}}^2.\end{aligned} \tag{A.99}$$

Proof. **Step 1: condition on the injected noise.** Let $q_{\phi|\xi}$ be the conditional student endpoint law:

$$q_{\phi|\xi} = (h_{\phi,\xi})_{\#}p_T. \tag{A.100}$$

Marginalizing the independent innovations gives

$$q_\phi(x) = \mathbb{E}_\xi[q_{\phi|\xi}(x)]. \tag{A.101}$$

Since $-\log$ is convex,

$$\begin{aligned}-\log q_\phi(x) &= -\log \mathbb{E}_\xi[q_{\phi|\xi}(x)] \\ &\le \mathbb{E}_\xi[-\log q_{\phi|\xi}(x)].\end{aligned} \tag{A.102}$$

Integrating against q_{tea} yields

$$\begin{aligned}\mathrm{KL}(q_{\mathrm{tea}}\|q_\phi) &= \int q_{\mathrm{tea}}(x)[\log q_{\mathrm{tea}}(x) - \log q_\phi(x)]\,\mathrm{d}x \\ &\le \mathbb{E}_\xi \int q_{\mathrm{tea}}(x)[\log q_{\mathrm{tea}}(x) - \log q_{\phi|\xi}(x)]\,\mathrm{d}x \\ &= \mathbb{E}_\xi \mathrm{KL}(q_{\mathrm{tea}}\|q_{\phi|\xi}).\end{aligned} \tag{A.103}$$

Step 2: pull back to the initial Gaussian. Apply $h_{\phi,\xi}^{-1}$ to both conditional laws. The student law becomes p_T, while the teacher law becomes $(a_\xi)_{\#}p_T$. Invariance of KL under a common diffeomorphism gives

$$\mathrm{KL}(q_{\mathrm{tea}}\|q_{\phi|\xi}) = \mathrm{KL}((a_\xi)_{\#}p_T\|p_T). \tag{A.104}$$

To verify the invariance, for a diffeomorphism h and densities μ, ν,

$$\begin{aligned}\mathrm{KL}(h_{\#}\mu\|h_{\#}\nu) &= \int \mu(x)\log\frac{\mu(x)|\det\partial_x h(x)|^{-1}}{\nu(x)|\det\partial_x h(x)|^{-1}}\,\mathrm{d}x \\ &= \mathrm{KL}(\mu\|\nu).\end{aligned} \tag{A.105}$$

The Gaussian prior has density

$$p_T(x) = (2\pi\sigma_{\mathrm{init}}^2)^{-d/2}\exp\left(-\frac{\|x\|_2^2}{2\sigma_{\mathrm{init}}^2}\right). \tag{A.106}$$

The transported density at $a_\xi(x)$ is

$$[(a_\xi)_{\#}p_T](a_\xi(x)) = \frac{p_T(x)}{\det(I + m_\xi(x))}. \tag{A.107}$$

The determinant is positive by the orientation assumption. Hence

$$\begin{aligned}\mathrm{KL}((a_\xi)_{\#}p_T\|p_T) &= \mathbb{E}_{x\sim p_T}\left[\log p_T(x) - \log p_T(a_\xi(x)) - \log\det(I + m_\xi(x))\right] \\ &= \frac{1}{2\sigma_{\mathrm{init}}^2}\mathbb{E}_{x\sim p_T}[\|a_\xi(x)\|_2^2 - \|x\|_2^2] - \mathbb{E}_{x\sim p_T}\log\det(I + m_\xi(x)).\end{aligned} \tag{A.108}$$

Expanding $a_\xi(x) = x + d_\xi(x)$ gives

$$\|a_\xi(x)\|_2^2 - \|x\|_2^2 = 2x^\top d_\xi(x) + \|d_\xi(x)\|_2^2. \tag{A.109}$$

Gaussian integration by parts, coordinate by coordinate, yields

$$\mathbb{E}[x_j d_{\xi,j}(x)] = \sigma_{\text{init}}^2 \mathbb{E}[\partial_{x_j} d_{\xi,j}(x)], \tag{A.110}$$

$$\mathbb{E}[x^\top d_\xi(x)] = \sigma_{\text{init}}^2 \mathbb{E}\,\text{tr}\, m_\xi(x). \tag{A.111}$$

Therefore the conditional KL has the exact representation

$$\begin{aligned}\text{KL}((a_\xi)_\# p_T \| p_T) = \frac{1}{2\sigma_{\text{init}}^2}\mathbb{E}\|d_\xi(x)\|_2^2 \\ + \mathbb{E}[\text{tr}\, m_\xi(x) - \log\det(I + m_\xi(x))].\end{aligned} \tag{A.112}$$

Step 3: control displacement and volume distortion. For a real matrix m with $\|m\|_{\text{op}} \le \delta < 1$, the logarithm series converges, and for $k \ge 2$,

$$\begin{aligned}|\,\text{tr}(m^k)| &\le \|m\|_{\text{F}} \|m^{k-1}\|_{\text{F}} \\ &\le \|m\|_{\text{F}}^2 \|m\|_{\text{op}}^{k-2} \\ &\le \delta^{k-2}\|m\|_{\text{F}}^2.\end{aligned} \tag{A.113}$$

Consequently,

$$\begin{aligned}\text{tr}\, m - \log\det(I+m) &= \sum_{k=2}^{\infty} \frac{(-1)^k}{k}\,\text{tr}(m^k) \\ &\le \|m\|_{\text{F}}^2 \sum_{k=2}^{\infty} \frac{\delta^{k-2}}{k} \\ &= \psi(\delta)\|m\|_{\text{F}}^2.\end{aligned} \tag{A.114}$$

This is an upper bound; it does not require m to be symmetric. By inverse stability,

$$\begin{aligned}\|d_\xi(x)\|_2 &= \|h_{\phi,\xi}^{-1}(h_{\text{tea}}(x)) - h_{\phi,\xi}^{-1}(h_{\phi,\xi}(x))\|_2 \\ &\le \ell_{\text{inv}} \|h_{\text{tea}}(x) - h_{\phi,\xi}(x)\|_2.\end{aligned} \tag{A.115}$$

Combining the last two bounds with Eq. (A.112),

$$\begin{aligned}\text{KL}(q_{\text{tea}} \| q_{\phi|\xi}) \le \frac{\ell_{\text{inv}}^2}{2\sigma_{\text{init}}^2}\mathbb{E}_{x_{t_0}} \|x_0^\phi(x_{t_0}, \xi) - x_0^{\text{tea}}(x_{t_0})\|_2^2 \\ + \psi(\delta)\mathbb{E}_{x_{t_0}} \|m_\xi(x_{t_0})\|_{\text{F}}^2.\end{aligned} \tag{A.116}$$

Average over ξ and apply Eq. (A.103). This proves Eq. (A.98).

Step 4: separate mean transport from stochastic sensitivity. For fixed x_{t_0}, expand the squared endpoint discrepancy around its conditional mean. Since

$$\mathbb{E}_\xi[x_0^\phi - \bar{x}_0^\phi] = 0, \tag{A.117}$$

the cross term vanishes and

$$\begin{aligned}\mathbb{E}_\xi \|x_0^\phi - x_0^{\text{tea}}\|_2^2 &= \mathbb{E}_\xi \|(\bar{x}_0^\phi - x_0^{\text{tea}}) + (x_0^\phi - \bar{x}_0^\phi)\|_2^2 \\ &= \|\bar{x}_0^\phi - x_0^{\text{tea}}\|_2^2 + \mathbb{E}_\xi \|x_0^\phi - \bar{x}_0^\phi\|_2^2.\end{aligned} \tag{A.118}$$

The Gaussian Poincaré inequality bounds the conditional variance. We give the calculation to make the stochastic-sensitivity term explicit.

Let $z, z' \sim \mathcal{N}(0, I_m)$ be independent, and let φ be a centered smooth scalar function. Define the Ornstein–Uhlenbeck semigroup

$$(p_\rho \varphi)(u) = \mathbb{E}_{z'} \varphi(e^{-\rho} u + \sqrt{1 - e^{-2\rho}} z'), \qquad \rho \ge 0. \tag{A.119}$$

Here p_ρ is an auxiliary operator, not a diffusion marginal density. Its generator and gradient relation are

$$\mathsf{a}\varphi = \Delta\varphi - u^\top \nabla\varphi, \tag{A.120}$$

$$\nabla p_\rho \varphi = e^{-\rho} p_\rho(\nabla\varphi). \tag{A.121}$$

Gaussian integration by parts gives

$$\begin{aligned}\mathbb{E}[h(z)\mathsf{a}h(z)] &= \sum_{j=1}^{m} \mathbb{E}[h(z)\partial_{jj}h(z) - z_j h(z)\partial_j h(z)] \\ &= -\mathbb{E}\|\nabla h(z)\|_2^2. \end{aligned} \tag{A.122}$$

Thus

$$\frac{\mathrm{d}}{\mathrm{d}\rho}\mathbb{E}[(p_\rho\varphi(z))^2] = -2\mathbb{E}\|\nabla p_\rho\varphi(z)\|_2^2. \tag{A.123}$$

Because $p_\rho\varphi \to 0$ in Gaussian L^2, integration gives

$$\begin{aligned}\mathrm{var}[\varphi(z)] &= -\int_0^\infty \frac{\mathrm{d}}{\mathrm{d}\rho}\mathbb{E}[(p_\rho\varphi(z))^2]\,\mathrm{d}\rho \\ &= 2\int_0^\infty \mathbb{E}\|\nabla p_\rho\varphi(z)\|_2^2\,\mathrm{d}\rho \\ &\le 2\int_0^\infty e^{-2\rho}\mathbb{E}\|p_\rho(\nabla\varphi)(z)\|_2^2\,\mathrm{d}\rho \\ &\le 2\int_0^\infty e^{-2\rho}\mathbb{E}\|\nabla\varphi(z)\|_2^2\,\mathrm{d}\rho \\ &= \mathbb{E}\|\nabla\varphi(z)\|_2^2. \end{aligned} \tag{A.124}$$

The penultimate line uses Jensen's inequality and Gaussian invariance of the semigroup. Approximation extends the inequality to $W^{1,2}$.

Apply it to each output coordinate of $x_0^\phi(x_{t_0}, \xi)$, conditionally on x_{t_0}, and sum:

$$\begin{aligned}\mathbb{E}_\xi\|x_0^\phi - \bar{x}_0^\phi\|_2^2 &\le \mathbb{E}_\xi\|\partial_\xi x_0^\phi\|_{\mathrm{F}}^2 \\ &= \mathbb{E}_\xi \sum_{i=0}^{n-1} \|\partial_{\xi_i} x_0^\phi\|_{\mathrm{F}}^2. \end{aligned} \tag{A.125}$$

The equality partitions the Jacobian columns by independent interval innovations. Substitute this bound into Eq. (A.118) and then into Eq. (A.98) to obtain Eq. (A.99). □

Interpretation. The three terms measure mean endpoint error, noise sensitivity, and relative Jacobian distortion. The inverse-stability and near-identity hypotheses are additional restrictions that the solver parameterization does not guarantee. This conditional KL result does not replace the main Wasserstein training surrogate or introduce a new regularizer.

B Experimental Details

B.1 Complete Algorithms

B.1.1 Training

Algorithm 1 RD-Solver training with cached teacher endpoints

Input: Frozen score model s_θ; teacher $\mathrm{solve}_{\mathrm{tea}}$; prior p_T; conditioning distribution $p(c)$; cache size P; number of optimizer updates R; effective batch size m; optimizer Opt; endpoint discrepancy d_{end} (Eq. (B.2)); student grid and base-solver rules.
Output: Learned parameters $\phi = \{\gamma_i, a_{i,j}, b_{i,j}, \kappa_i\}_{i=0}^{n-1}$.

$\mathcal{D} \leftarrow \varnothing$
for $\ell = 1, \dots, P$ **do**
  Sample $z_\ell \sim p_T$ and $c_\ell \sim p(c)$
  $y_\ell \leftarrow \mathrm{stopgrad}(\mathrm{solve}_{\mathrm{tea}}(z_\ell, c_\ell))$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(z_\ell, c_\ell, y_\ell)\}$
end for
Initialize a_i, b_i from the base predictor–corrector coefficients
$\gamma_i \leftarrow 10^{-4}$, $\kappa_i \leftarrow 1$ for $i = 0, \dots, n-1$
for $r = 1, \dots, R$ **do**
  Draw a batch $\{(z_\ell, c_\ell, y_\ell)\}_{\ell=1}^{m}$ from $\mathcal{D}$
  Sample $\xi_i^{(\ell)} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, I)$ for all i, ℓ
  $x_\ell \leftarrow \mathrm{RDSample}(z_\ell, c_\ell, \xi^{(\ell)}; \phi)$ for all ℓ ▷ Algorithm 2
  $\widehat{\mathcal{L}} \leftarrow m^{-1} \sum_{\ell=1}^{m} d_{\mathrm{end}}(x_\ell, y_\ell)$
  $\phi \leftarrow \mathrm{Opt}\left(\phi, \nabla_\phi \widehat{\mathcal{L}}\right)$
end for
return ϕ

B.1.2 SAMPLING

Algorithm 2 RD-Solver sampling: $\mathrm{RDSample}(x_{t_0}, c, \xi; \phi)$

Input: Initial state x_{t_0}; conditioning c; innovations $\xi = (\xi_i)_{i=0}^{n-1}$; frozen s_θ; learned ϕ; target times $t_0 > \cdots > t_n = 0$; history cap k; base startup and terminal routines.
Output: Final sample x_{t_n}.

$t_0^{\mathrm{c}} \leftarrow t_0$; $\mathcal{H} \leftarrow \varnothing$
for $i = 0, \ldots, n-1$ **do**
  if $i = n-1$ **then**
    $x_{t_n} \leftarrow \mathrm{Terminal}_\phi(x_{t_i}, t_i^{\mathrm{c}}, c, \mathcal{H}, \xi_i)$ ▷ counted in NFE
    return x_{t_n}
  else if $\mathcal{H} = \varnothing$ **then**
    $(x_{t_{i+1}}, t_{i+1}^{\mathrm{c}}, \mathcal{H}) \leftarrow \mathrm{Startup}_\phi(i, x_{t_i}, t_i^{\mathrm{c}}, c, \xi_i)$ ▷ counted in NFE
  else
    $k_i \leftarrow \min(k, |\mathcal{H}|)$
    $\Delta t_i \leftarrow t_i^{\mathrm{c}} - t_{i+1}$; $v_i \leftarrow (t_i^{\mathrm{c}})^2 - t_{i+1}^2$
    $h_i \leftarrow -(1+\gamma_i)\Delta t_i$; $\nu_i \leftarrow \sqrt{\gamma_i v_i}$
    for $j = 1, \ldots, k_i$ **do**
      $(\widetilde{x}_{t_{i-j+1}}, t_{i-j+1}^{\mathrm{p}}, s_{i-j}^\theta) \leftarrow \mathcal{H}[j]$
      $u_{i,i-j}^\theta \leftarrow (1+\gamma_i) t_{i-j+1}^{\mathrm{p}} s_{i-j}^\theta$
    end for
    $\widetilde{x}_{t_{i+1}} \leftarrow x_{t_i} + \Delta t_i \sum_{j=1}^{k_i} a_{i,j} u_{i,i-j}^\theta + \nu_i \xi_i$
    $\delta_i \leftarrow t_i^{\mathrm{c}} + h_i$; $t_{i+1}^{\mathrm{p}} \leftarrow \sqrt{t_{i+1}^2 + \gamma_i(1+\gamma_i)(\Delta t_i)^2}$
    $s_i^\theta \leftarrow s_\theta(\widetilde{x}_{t_{i+1}}, \kappa_i t_{i+1}^{\mathrm{p}}; c)$
    $u_{i,i}^\theta \leftarrow (1+\gamma_i) t_{i+1}^{\mathrm{p}} s_i^\theta$
    $x_{t_{i+1}} \leftarrow x_{t_i} + \Delta t_i \left(b_{i,0} u_{i,i}^\theta + \sum_{j=1}^{k_i} b_{i,j} u_{i,i-j}^\theta \right) + \nu_i \xi_i$
    $A_i \leftarrow \delta_i + h_i b_{i,0} (\delta_i / t_{i+1}^{\mathrm{p}} - 1)$
    $B_i \leftarrow 1 + h_i b_{i,0} / t_{i+1}^{\mathrm{p}}$; $t_{i+1}^{\mathrm{c}} \leftarrow \sqrt{A_i^2 + \gamma_i v_i B_i^2}$
    Prepend $(\widetilde{x}_{t_{i+1}}, t_{i+1}^{\mathrm{p}}, s_i^\theta)$ to $\mathcal{H}$
    Keep the k most recent records in $\mathcal{H}$ ▷ skipped for "All"
  end if
end for

B.2 IMPLEMENTATION DETAILS

Teacher endpoints are cached with their initial noise and conditioning and reused across student budgets with the same teacher. We jointly optimize $\phi = \{\gamma_i, a_{i,j}, b_{i,j}, \kappa_i\}$ with a single optimizer; the pretrained model is frozen. Initialization uses

$$\sqrt{\gamma_i} = 0.01, \qquad \kappa_i = 1, \quad i = 0, \ldots, n-1, \tag{B.1}$$

and analytical Adams–Bashforth predictor–corrector weights (Atkinson et al., 2009). A predictor history of k uses at most k cached evaluations; the corrector adds the fresh evaluation. Startup uses the available history, and "All" removes the history cap.

Endpoint supervision. Teacher and student share initial noise and conditioning; only the student receives the innovations ξ. We optimize $\mathcal{L}_H$ in the supervision representation, as defined in Appendix A.5. The discrepancy in Algorithm 1 is

$$d_{\mathrm{end}}(x, x') = \begin{cases} \|h_{\mathrm{inc}}(x) - h_{\mathrm{inc}}(x')\|_2^2, & \text{pixel-space models,} \\ \|x - x'\|_2^2, & \text{latent-space models,} \end{cases} \tag{B.2}$$

where h_{inc} is a frozen Inception feature extractor and latent outputs are measured before decoding. Gradients through the student input are retained. The Inception extension of Theorem 4.5 in Appendix A.5 bounds the squared Wasserstein distance in feature space; a pixel-space bound requires additional assumptions on the feature map. The displayed loss sums over coordinates.

When an implementation averages over the d_H representation coordinates, its expected MSE equals $\mathcal{L}_H/d_H$; multiplying by d_H recovers the stated upper bound. This fixed scaling leaves the minimizer unchanged.

Teacher configurations. The dataset- and budget-specific settings are listed in Tables B.1 to B.3.

Table B.1: Teacher settings for pixel-space models and LSUN-Bedroom. Polynomial uses $\rho = 7$.

Dataset	Student NFE	Teacher solver	Order	Step	Time schedule
CIFAR-10	3, 5, 7, 9	iPNDM	4	35	Polynomial
FFHQ	3, 5, 7, 9	iPNDM	4	35	Polynomial
AFHQv2	3, 5, 7, 9	iPNDM	4	35	Polynomial
ImageNet-64	3, 5, 7, 9	iPNDM	4	35	Polynomial
LSUN-Bedroom	3, 7	iPNDM	3	35	Discrete
	5, 9	iPNDM	4	35	Discrete

Table B.2: Teacher settings for Stable Diffusion.

Student NFE	8	12	16	20
Teacher solver	iPNDM	iPNDM	iPNDM	Heun
Order	2	2	2	2
Steps	50	50	75	50
Time schedule	Discrete	Discrete	Discrete	Polynomial
Schedule parameter ρ	1	1	1	7

Table B.3: FLUX.1-dev teacher settings.

Student NFE	3	5	7
Teacher solver	Euler	Euler	Euler
Order	1	1	1
Teacher steps	50	50	50
Time schedule	Polynomial	Polynomial	Polynomial

B.2.1 Pixel-space models

We use official EDM checkpoints (Karras et al., 2022): unconditional VP-trained models for CIFAR-10, FFHQ, and AFHQv2, and the class-conditional ADM-architecture model for ImageNet-64. Resolutions are 32×32 for CIFAR-10 and 64×64 for the other datasets. ImageNet labels are sampled uniformly and shared by teacher and student. The polynomial noise grid uses $\rho = 7$ and endpoints 0.002 and 80.

Each dataset uses 20,000 cached pairs, predictor history $k = 3$, global batch size 16, and 10,000 backward passes per NFE budget. Adam uses learning rate 10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$, and $\epsilon = 10^{-8}$. Linear warmup increases the rate from 10% to its specified value during the first 10% of training; cosine decay then reduces it to 1%. Gradient norms are clipped at 1.0. Before Inception feature extraction, images are rescaled from $[-1, 1]$ to $[0, 255]$ and clamped. FID uses 50,000 generated images and EDM reference statistics.

B.2.2 LSUN-Bedroom

The unconditional LDM-vq-4 (Rombach et al., 2022) generates 256×256 images from $3 \times 64 \times 64$ latents. Guidance is 1.0. The discrete schedule uniformly spaces the wrapper's native time coordinate ($\rho = 1$), then maps it to noise levels by Eq. (B.6). Training uses 20,000 cached pairs,

latent endpoint squared L_2 loss, history $k = 3$, global batch size 4, and 10,000 backward passes per NFE. Adam settings, warmup, decay, and gradient clipping match the pixel-space configuration. FID uses 50,000 samples and LSUN-Bedroom reference statistics.

B.2.3 STABLE DIFFUSION V1.5

Stable Diffusion v1.5 (Rombach et al., 2022) generates 512×512 images from $4 \times 64 \times 64$ latents. Teacher and student share MS-COCO prompts; classifier-free guidance is 7.5 with an empty unconditional prompt. Training uses 5,000 cached pairs and final latent MSE averaged over the batch and latent coordinates. Predictor history is $k = 2, \text{All}, 3, 2$ at 8, 12, 16, and 20 NFEs, respectively.

The global batch size is 2, and Adam uses learning rate 10^{-5} with the same momentum parameters, warmup, decay, and clipping as the pixel-space configuration. Each budget uses 2,000 backward passes. The student does not use AFS. NFE counts both conditional and unconditional evaluations. FID uses 30,000 MS-COCO samples.

B.2.4 FLUX.1-DEV

FLUX.1-dev (Labs, 2024) runs in bfloat16 with guidance 3.5. Training and quantitative evaluation use 512×512 images: initial $16 \times 64 \times 64$ latents are packed into 1024×64 tokens. Figure 1 uses direct sampling at multiple native aspect ratios with the same trained 7-NFE coefficients.

Training uses 5,000 cached pairs, history $k = 2$, and endpoint latent MSE computed after casting both latents to float32. AdamW uses learning rate 10^{-5}, weight decay 0.05, and cosine decay without warmup. Each device processes one sample per microbatch; two microbatches are accumulated per update. Training lasts 2,000 backward calls per NFE budget. FID uses 30,000 MS-COCO images. CLIP score is mean cosine similarity between normalized image and text features, multiplied by 100.

B.2.5 TIME-GRID DEFINITIONS

Let $r_j = j/m$ for $j = 0, \ldots, m$, with m intervals between valid endpoints $t_{\max} > t_{\min} > 0$. Final denoising to zero is handled separately by the solver.

Polynomial (EDM-time). The EDM-style rule interpolates linearly in the $1/\rho$ power of the sampling coordinate:

$$t_j = \left[(1 - r_j)t_{\max}^{1/\rho} + r_j t_{\min}^{1/\rho}\right]^{\rho}, \qquad \rho > 0. \tag{B.3}$$

In the EDM coordinate, t is the noise standard deviation.

Time LogSNR. Using the mean and noise schedules from Eq. (2), define half-logSNR by

$$\lambda(t) = \log \frac{\alpha_t}{\sigma_t} = \frac{1}{2} \log \frac{\alpha_t^2}{\sigma_t^2}. \tag{B.4}$$

When λ is invertible on the integration interval, uniform spacing in this coordinate gives

$$t_j = \lambda^{-1}((1 - r_j)\lambda(t_{\max}) + r_j \lambda(t_{\min}))\,. \tag{B.5}$$

Using full logSNR, or its negative, yields the same grid after the corresponding inverse map.

Time Discrete. The discrete schedule places grid points in the diffusion model's time coordinate τ, then converts them to noise levels. Let $\bar{\alpha}(\tau)$ denote the cumulative signal power, with $\log \bar{\alpha}$ linearly interpolated between pretrained timesteps. Choose $\tau_{\min}$ and $\tau_{\max}$ such that $\bar{\alpha}(\tau_{\min}) = (1 + t_{\min}^2)^{-1}$ and $\bar{\alpha}(\tau_{\max}) = (1 + t_{\max}^2)^{-1}$. The grid is

$$\tau_j = \left[(1 - r_j)\tau_{\max}^{1/\rho} + r_j \tau_{\min}^{1/\rho}\right]^{\rho}, \qquad t_j = \sqrt{\frac{1 - \bar{\alpha}(\tau_j)}{\bar{\alpha}(\tau_j)}}, \qquad \rho > 0. \tag{B.6}$$

These rules specify the target grid. Each step starts from the calibrated time carried by the preceding corrected state. The additional learned multiplier changes the score query to $\kappa_i t_{i+1}^{\mathrm{p}}$.

C More Experimental Results

C.1 More ablation

We complement the component study in Table 3 with five ablations of numerical and training choices. All experiments use CIFAR-10 and report FID at 3, 5, 7, and 9 NFEs.

C.1.1 Coefficient Initialization

We compare uniform, Euler, and analytical Adams–Bashforth (Kress, 2012) initializations of the predictor–corrector coefficients (Table C.1). Adams–Bashforth yields the lowest FID at every budget. At 3 NFEs, it achieves 6.87, compared with 7.77 for Euler and 8.47 for uniform initialization; at 9 NFEs, the corresponding values are 2.20, 2.30, and 2.44.

Table C.1: Ablation of predictor–corrector coefficient initialization on CIFAR-10. FID (↓) is reported at 3, 5, 7, and 9 NFEs; the best result in each column is in **bold**.

	FID (↓)			
Initialization / NFE	**3**	**5**	**7**	**9**
Uniform	8.47	2.94	2.53	2.44
Euler	7.77	2.89	2.42	2.30
Adams–Bashforth	**6.87**	**2.71**	**2.31**	**2.20**

C.1.2 Initial Time Schedule

Table C.2 compares the EDM polynomial grid ($\rho = 7$) with logSNR and time-uniform schedules. The polynomial grid achieves the lowest FID across all tested budgets. LogSNR is competitive at larger budgets, reaching 2.27 at 9 NFEs versus 2.20 for the polynomial grid, but has a larger gap at 3 NFEs (8.29 versus 6.87). Time-uniform is substantially worse throughout, with FID still at 16.29 at 9 NFEs.

Table C.2: Ablation of the initial time schedule on CIFAR-10. FID (↓) is reported at 3, 5, 7, and 9 NFEs; the best result in each column is in **bold**.

	FID (↓)			
Initial time schedule / NFE	**3**	**5**	**7**	**9**
Log-SNR	8.29	2.84	2.42	2.27
Time-uniform	41.15	27.62	21.94	16.29
EDM (polynomial, $\rho = 7$)	**6.87**	**2.71**	**2.31**	**2.20**

C.1.3 Number of Training Pairs

We vary the teacher–student training set between 10k, 20k, and 30k pairs (Table C.3). The 20k-pair setting gives the lowest FID at all four budgets. At 3 NFEs, its FID is 6.87, versus 7.75 with 10k pairs and 7.95 with 30k. The differences are smaller at 9 NFEs: 2.20, 2.30, and 2.29, respectively.

Table C.3: Ablation of the number of teacher–student pairs on CIFAR-10. FID (↓) is reported at 3, 5, 7, and 9 NFEs; the best result in each column is in **bold**.

	FID (↓)			
Teacher–student pairs / NFE	**3**	**5**	**7**	**9**
10k	7.75	2.88	2.41	2.30
30k	7.95	2.88	2.45	2.29
20k	**6.87**	**2.71**	**2.31**	**2.20**

C.1.4 TEACHER SOLVER

We compare the teacher configurations listed in Table C.4, varying the solver and its step count with iPNDM configured at second order. The 35-step, second-order iPNDM teacher gives the lowest student FID at all four budgets. At 5 NFEs, it yields 2.71, while the alternatives range from 2.91 to 2.94. Neither increasing iPNDM to 50 teacher steps nor switching to a 50-step Heun teacher improves the result.

Table C.4: Ablation of the teacher solver on CIFAR-10. FID (↓) is reported at 3, 5, 7, and 9 NFEs; the best result in each column is in **bold**. Parentheses specify teacher steps and, for iPNDM, solver order; the columns give student NFEs.

	FID (↓)			
Teacher solver / NFE	**3**	**5**	**7**	**9**
iPNDM (50 steps, order 2)	7.92	2.93	2.44	2.29
Heun (50 steps)	7.95	2.92	2.42	2.29
DPM-Solver-2 (35 steps)	7.85	2.94	2.43	2.31
iPNDM (18 steps, order 2)	7.89	2.91	2.42	2.29
iPNDM (35 steps, order 2)	**6.87**	**2.71**	**2.31**	**2.20**

C.1.5 HISTORY REUSE

Table C.5 varies the history-window configuration $K \in \{1, 2, 3, 4, N\}$, where $K = N$ uses all available history. The $K = 3$ configuration gives the lowest FID at every budget. At 5 NFEs, it reaches 2.71, compared with 3.13 for $K = 1$ and 2.78 for all-history reuse; at 9 NFEs, these values are 2.20, 2.29, and 2.24.

Table C.5: Ablation of history reuse in the predictor–corrector on CIFAR-10. FID (↓) is reported at 3, 5, 7, and 9 NFEs; the best result in each column is in **bold**. K controls the number of historical evaluations reused by the predictor–corrector; $K = N$ uses all available history.

	FID (↓)			
History size / NFE	**3**	**5**	**7**	**9**
$K = 1$	8.50	3.13	2.46	2.29
$K = 2$	8.03	2.98	2.44	2.33
$K = 4$	7.99	2.93	2.49	2.29
$K = N$	7.98	2.78	2.37	2.24
$K = 3$	**6.87**	**2.71**	**2.31**	**2.20**

C.2 Comparison with Distillation Methods

We compare the sampling cost of each method with the cost of adapting the pretrained model. Distillation updates a generator network to reduce its sampling steps, whereas RD-Solver keeps the pretrained network frozen and learns only solver parameters. We combine published baseline results with the RD-Solver results already reported in Table 1; we ran no distillation experiments ourselves.

Training-cost convention. Times marked † are approximate wall-clock times measured for RD-Solver training: about 6 minutes on 4× RTX 5090 for CIFAR-10, 10 minutes on 8× RTX 5090 for ImageNet-64, and 70 minutes on 8× RTX Pro 6000 for Stable Diffusion v1.5. Baseline times are taken from the cited papers. Because hardware and timing scopes differ, these times do not support a controlled wall-time comparison.

Consistency distillation, rectified flow, and progressive distillation. Table C.6 follows the three comparisons in LD3's Appendix C. RD-Solver reaches FID 2.71 with 5 NFEs, compared with 2.93 at 2 NFEs for consistency distillation and 3.00 at 4 NFEs for progressive distillation, so these operating points use three and one additional evaluations, respectively. At 7 NFEs, RD-Solver reaches FID 2.31, compared with 2.57 at 8 NFEs for progressive distillation. The baseline numbers come from other papers and use different backbones; we did not rerun them on a matched architecture.

Table C.6: Comparison with the three generation methods considered in LD3's Appendix C on CIFAR-10. Baseline FIDs, times, and devices are taken from that comparison (Tong et al., 2025); our FIDs are from Table 1. †: approximate measured wall-clock training time (see text).

Method	Backbone	NFE	FID↓	Wall time	Devices
CD (Song et al., 2023)	NCSN++	2	2.93	1 day	8× A100
RF (Liu et al., 2023)	DDPM++	127	2.58	1 day	8× A100
PD (Salimans & Ho, 2022)	iDDPM*	1	9.12	1 day	8× TPU
PD	iDDPM*	2	4.51	1 day	8× TPU
PD	iDDPM*	4	3.00	1 day	8× TPU
PD	iDDPM*	8	2.57	1 day	8× TPU
RD-Solver	EDM (VP)	3	6.87	≈ 6 min†	4× RTX 5090
RD-Solver	EDM (VP)	5	2.71	≈ 6 min†	4× RTX 5090
RD-Solver	EDM (VP)	7	2.31	≈ 6 min†	4× RTX 5090

*PD modifies the iDDPM architecture. The RF row is the 127-NFE configuration reported by LD3, not a one-step distilled model; its training time is LD3's estimate from the RF implementation, and the CD training time is also taken from LD3. The table lists backbone and hardware differences as reported.

Distribution matching distillation on pixel-space models. Table C.7 adds DMD (Yin et al., 2024b) and DMD2 (Yin et al., 2024a). On unconditional CIFAR-10, RD-Solver uses 20k cached pairs, compared with DMD's 500k, and obtains FID 2.71 at 5 NFEs versus DMD's 3.77 at 1 NFE. On ImageNet-64 the cached-pair counts are closer (20k versus 25k). DMD2 reaches much lower FID at 1 NFE, but its standard and extended runs take approximately two and five days on seven A100 GPUs. The RD-Solver times in the table are approximate measured wall-clock times for fitting the few-step sampler.

Table C.7: Pixel-space comparison with DMD and DMD2, using their published results (Yin et al., 2024b;a). "Pairs" counts cached teacher noise–image pairs, not real training images or repeated sample exposures. "BP" denotes backward passes in our implementation; "iter." denotes iterations reported by the respective distillation paper. NR means not reported for that configuration. †: approximate measured wall-clock time (see text).

Method	NFE	FID↓	Pairs	Optimization budget	Wall time	Devices
Unconditional CIFAR-10 32×32*; EDM (VP)*						
DMD	1	3.77	500k	300k iter.	NR	7 GPUs (type NR)
RD-Solver	5	2.71	20k	10k BP	$\approx$ 6 min†	4× RTX 5090
RD-Solver	7	2.31	20k	10k BP	$\approx$ 6 min†	4× RTX 5090
Class-conditional ImageNet 64×64*; EDM (ADM)*						
DMD	1	2.62	25k	350k iter.	NR	7 GPUs (type NR)
DMD2	1	1.51	None	200k iter.	$\approx$ 2 days	7× A100
DMD2 (long)	1	1.28	None	400k+150k iter.	$\approx$ 5 days	7× A100
RD-Solver	5	4.62	20k	10k BP	$\approx$ 10 min†	8× RTX 5090
RD-Solver	7	3.14	20k	10k BP	$\approx$ 10 min†	8× RTX 5090
RD-Solver	9	2.91	20k	10k BP	$\approx$ 10 min†	8× RTX 5090

DMD2 uses real ImageNet images for adversarial training rather than a teacher-pair cache; "None" does not mean data-free training. Optimization budgets use the original counting conventions and are not equal-cost units. Our 20k pairs are unique cached examples, not the total number of sample exposures during 10k backward passes.

Text-to-image generation. Table C.8 includes only Stable Diffusion v1.5 results and leaves out SDXL. DMD reports FID 11.49 after approximately 36 hours of training on 72 A100 GPUs. RD-Solver obtains a similar FID of 11.44 at 6 sampling steps (12 NFEs) with 5k cached teacher pairs, compared with DMD's 500k, and about 70 minutes of training on eight RTX Pro 6000 GPUs. DMD2 attains a lower FID of 8.35 in one step after a reported 26-hour training run on 64 A100 GPUs. It drops the paired-regression cache and instead uses 500k real images for its GAN objective together with LAION prompts, so its 500k images are not teacher–student pairs.

Table C.8: Reference comparison on MS-COCO 512×512 with the Stable Diffusion v1.5 backbone. DMD and DMD2 results and costs are taken from their original papers (Yin et al., 2024b;a); our FIDs are from Table 1. Training prompts, guidance, and FID protocols differ across papers. †: approximate measured wall-clock time (see text).

Method	Steps	NFE	FID↓	Teacher pairs	Wall time	Devices
DMD	1	1	11.49	500k	$\approx$ 36 h	72× A100
DMD2	1	1	8.35	None	$\approx$ 26 h	64× A100
RD-Solver	4	8	14.52	5k	$\approx$ 70 min†	8× RTX Pro 6000
RD-Solver	6	12	11.44	5k	$\approx$ 70 min†	8× RTX Pro 6000

DMD2 uses 500k real LAION images for its GAN objective and LAION prompts, not 500k cached teacher pairs. Its guidance is distilled into the generator. RD-Solver retains classifier-free guidance at inference, so one step costs two NFEs. Baseline timing scopes are kept as reported; we do not assume that preprocessing costs are identical across methods.

The text-to-image results use different training prompts, guidance, and FID protocols: DMD and DMD2 use LAION prompts, while Appendix B.2.3 specifies our configuration. The rows are therefore reference operating points and do not form a controlled comparison of quality or runtime. In our guided sampler, six steps require 12 NFEs because each step evaluates conditional and unconditional predictions. The small additive NFE gaps on CIFAR-10 therefore do not imply near-one-step inference cost on Stable Diffusion.

D BASELINE DETAILS

Baseline configurations. Table D.1 lists sources for the baseline methods. For the baselines other than INDIS and BézierFlow, numerical values follow the published comparisons in DyWeight and AdaSDE. We retain the dataset–budget conventions and mark unavailable entries with dashes. For LD3, we report the best FID across its base solvers, as in the source comparison. EDM/Heun also appears in the qualitative comparisons in Appendix E.

INDIS adapts timestep allocations to each input instance; BézierFlow optimizes continuous stochastic interpolant schedules using Bézier control points. In Table 1, the available INDIS results at 3, 5, and 7 NFEs are taken from its paper, while the 9-NFE results use our evaluations with the selected exponential moving average (EMA) checkpoint. BézierFlow's 3-NFE results are taken from its paper; at 5, 7, and 9 NFEs, we report the lower FID from our UniPC and iPNDM evaluations at each budget.

The two-stage AMED and EPD solver configurations raise an accounting issue. With an Analytical First Step (AFS) (Dockhorn et al., 2022), one network evaluation is replaced analytically; two-stage updates then give odd-NFE budgets in the reported comparisons (Zhao et al., 2026). Even-NFE runs obtained by disabling AFS change the initialization as well as the evaluation count and can therefore have different quality. We retain the reported odd-NFE configurations at 3, 5, 7, and 9 NFEs in the main comparison. This convention is separate from the classifier-free-guidance accounting used for Stable Diffusion v1.5, where the main table counts two NFEs per guided model evaluation.

Code repositories. Table D.1 collects reference implementations for the compared methods. AMED-Solver and AMED-Plugin share a repository, as do the two EPD variants. Repository links are provided for reference; they do not imply that every source-reported baseline was rerun in this work. For DLMS and S4S-Alt, we link the published method descriptions.

Table D.1: Reference code and result sources for the baselines. The iPNDM implementation is distributed with DEIS. An additional iPNDM implementation and AMED are available in the `diff-sampler` toolbox. For DLMS and S4S-Alt, the links point to the publications.

Method	GitHub repository / published source
iPNDM	https://github.com/qsh-zh/deis
DPM-Solver++	https://github.com/LuChengTHU/dpm-solver
UniPC	https://github.com/wl-zhao/UniPC
AMED-Solver / Plugin	https://github.com/zju-pi/diff-sampler
EPD-Solver / Plugin	https://github.com/BeierZhu/EPD
LD3	https://github.com/vinhsuhi/LD3
DLMS	Published results (OpenReview)
S4S-Alt	Published results (ICML 2025)
AdaSDE	https://github.com/Westlake-AGI-Lab/AdaSDE
DyWeight	https://github.com/Westlake-AGI-Lab/DyWeight
INDIS	https://github.com/851695e35/INDIS
BézierFlow	https://github.com/KAIST-Visual-AI-Group/BezierFlow
EDM / Heun	https://github.com/NVlabs/edm

Main-paper qualitative comparison. For Figure 3, each solver uses 5 NFEs and AFS. DPM-Solver++ uses its second-order data-prediction formulation with a logSNR grid; iPNDM uses order 2 with the polynomial grid ($\rho = 7$). RD-Solver uses its dataset-specific 5-NFE checkpoints. We train AMED-Solver predictors using the official implementation with a Heun teacher, 10,000 training samples, and the `amed` student; the grid is time-uniform for pixel-space models and discrete for LSUN-Bedroom. Within each displayed column, all methods share the initial noise and ImageNet class label. These qualitative reruns do not replace the quantitative results in Table 1.

E ADDITIONAL VISUALIZATIONS

E.1 PIXEL-SPACE DIFFUSION MODELS

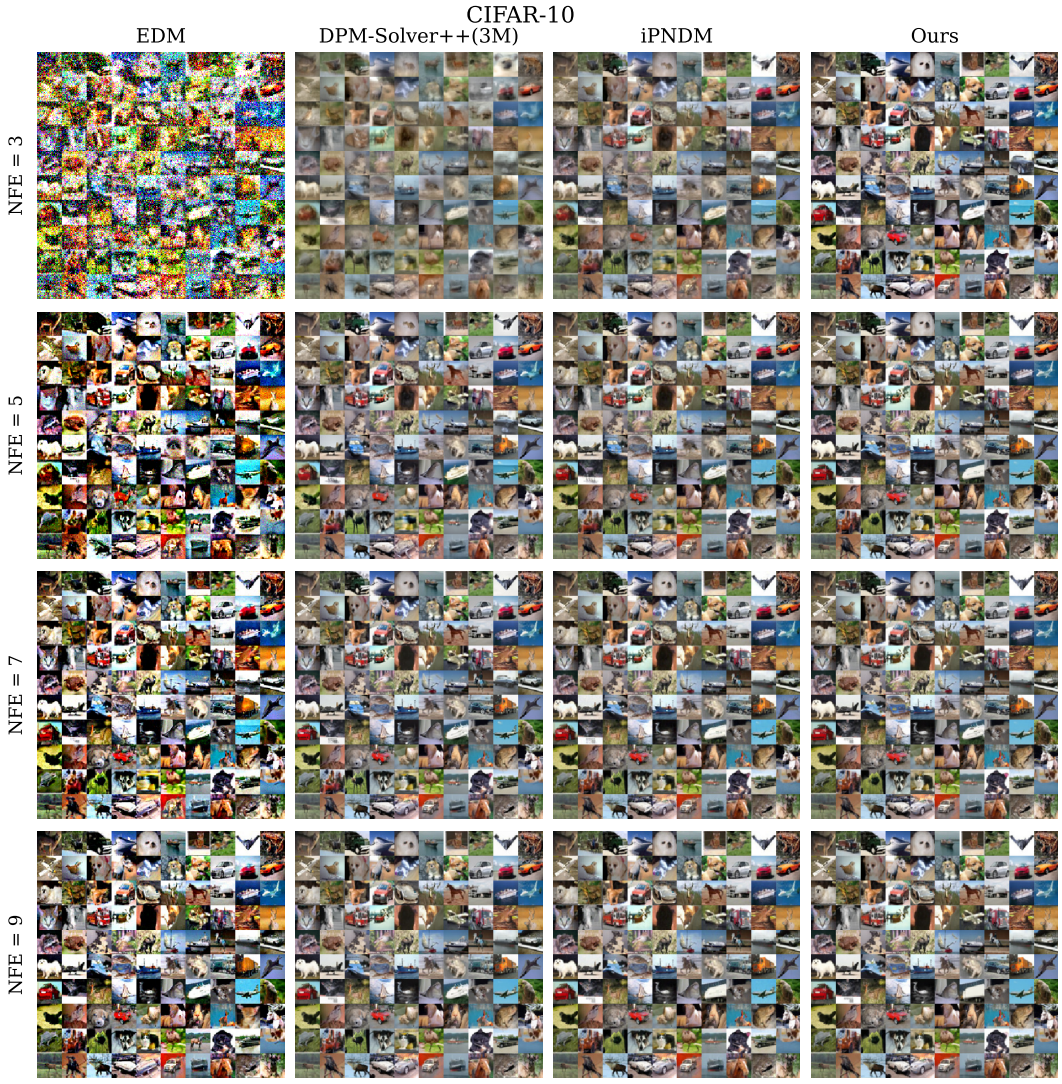

Figure 5: Qualitative comparison on CIFAR-10 at 3, 5, 7, and 9 NFEs. The columns compare EDM, DPM-Solver++(3M), iPNDM, and RD-Solver (Ours).



Figure 6: Qualitative comparison on FFHQ at 3, 5, 7, and 9 NFEs. The columns compare EDM, DPM-Solver++(3M), iPNDM, and RD-Solver (Ours).



Figure 7: Qualitative comparison on AFHQv2 at 3, 5, 7, and 9 NFEs. The columns compare EDM, DPM-Solver++(3M), iPNDM, and RD-Solver (Ours).



Figure 8: Qualitative comparison on class-conditional ImageNet-64 at 3, 5, 7, and 9 NFEs. The columns compare EDM, DPM-Solver++(3M), iPNDM, and RD-Solver (Ours).

E.2 LATENT-SPACE DIFFUSION MODELS

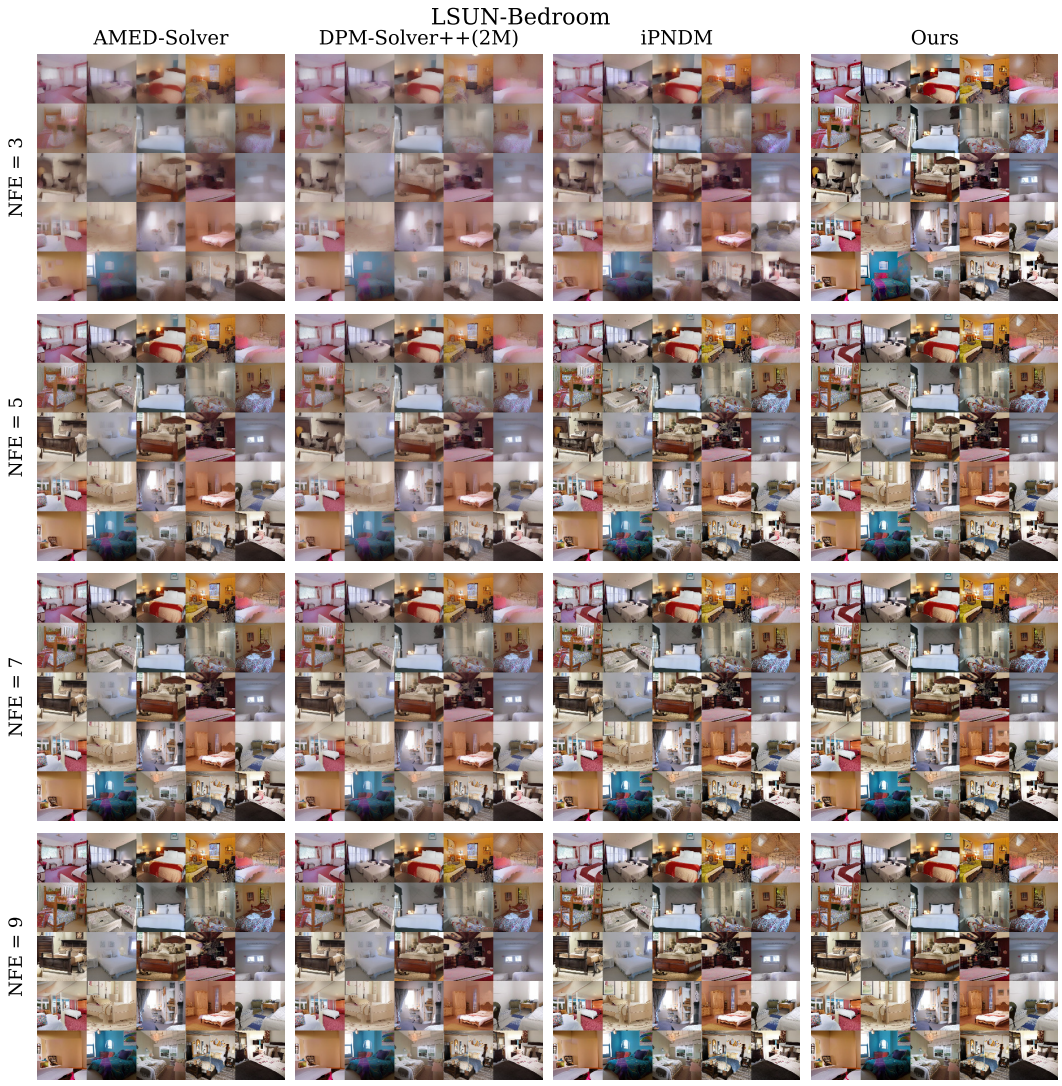

Figure 9: Qualitative comparison on LSUN-Bedroom at 3, 5, 7, and 9 NFEs. The columns compare AMED-Solver, DPM-Solver++(2M), iPNDM, and RD-Solver (Ours).

E.3 TEXT-TO-IMAGE GENERATION

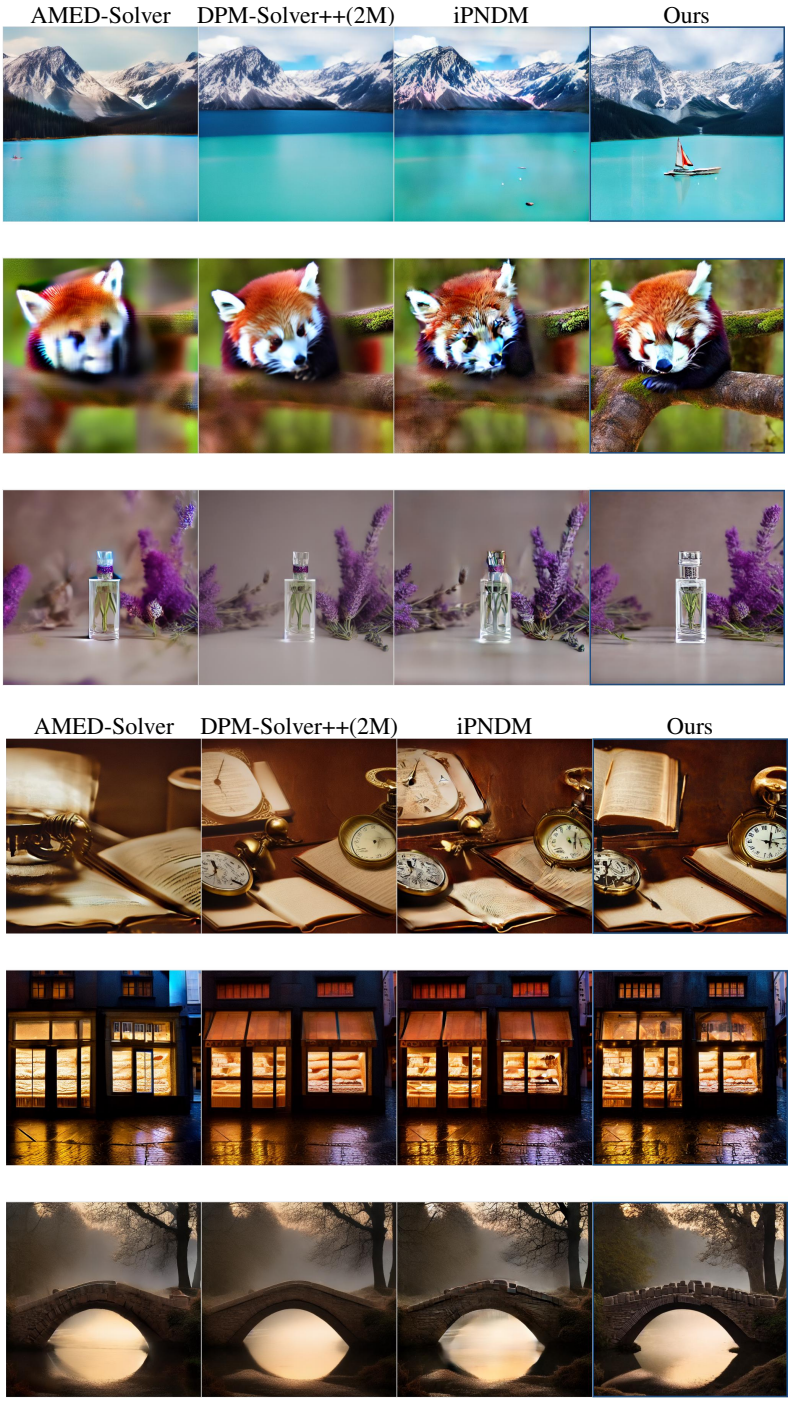

Figure 10: Qualitative comparison using Stable Diffusion v1.5. Top: 8 NFEs; bottom: 12 NFEs.

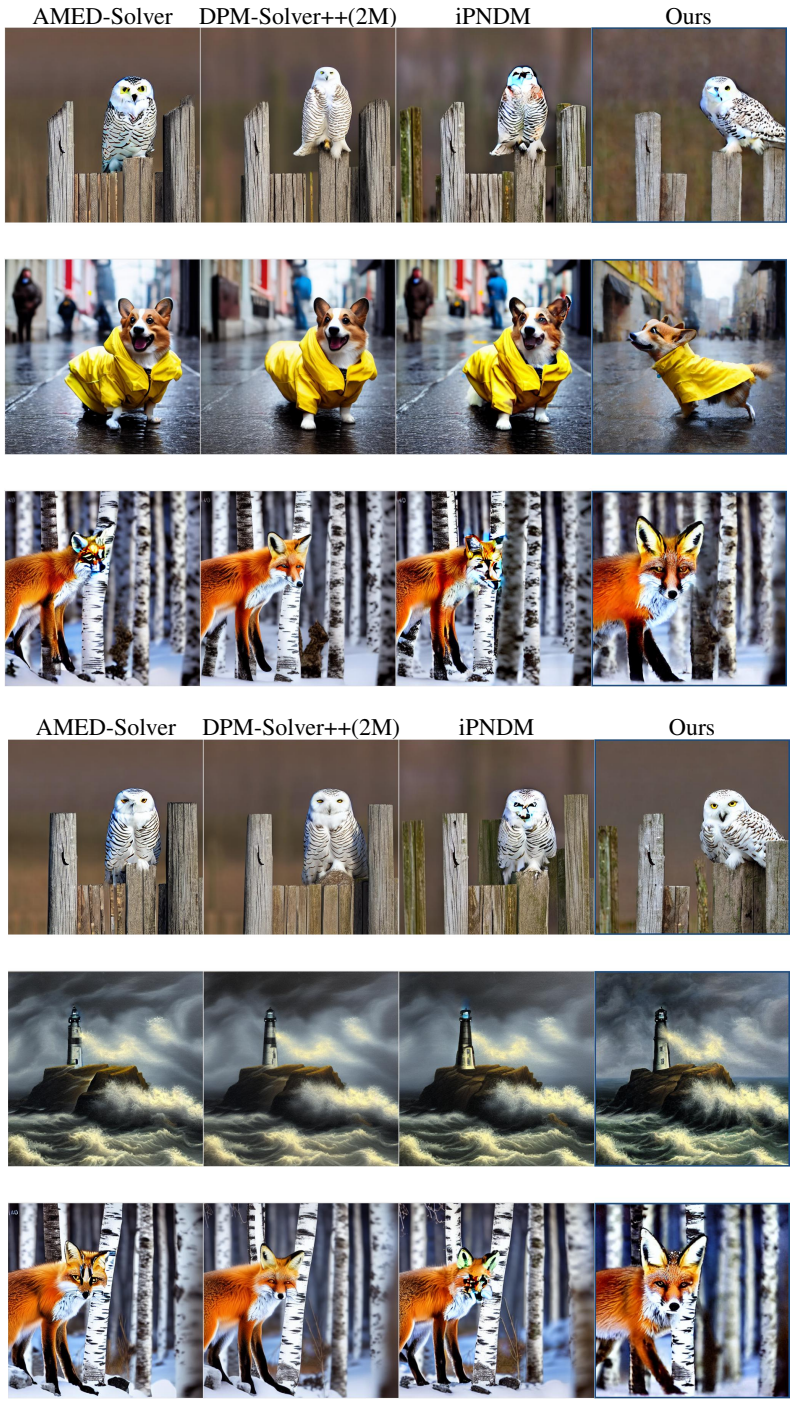

Figure 11: Qualitative comparison using Stable Diffusion v1.5. Top: 16 NFEs; bottom: 20 NFEs.

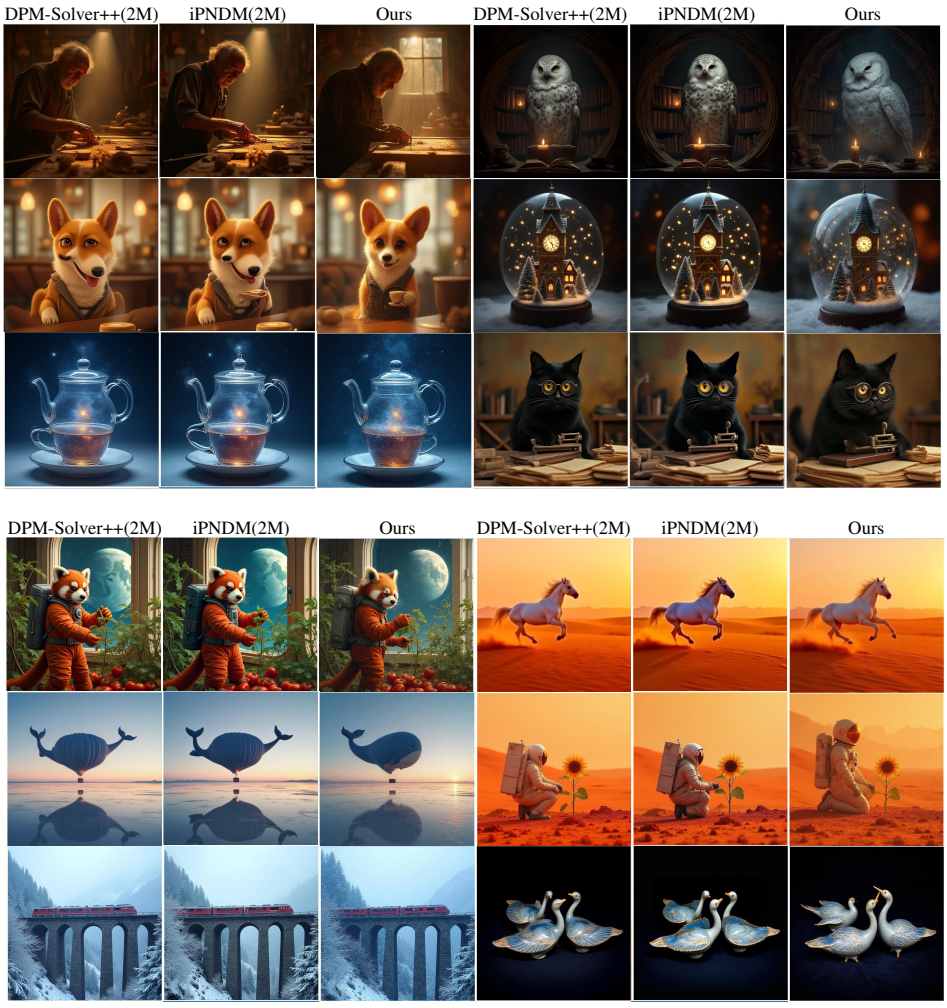

Figure 12: Qualitative comparison using FLUX.1-dev at 5 NFEs (top) and 7 NFEs (bottom).

Stable Diffusion v1.5 prompts

NFE = 8. (1) A small sailboat crossing a turquoise lake beneath snow-capped mountains, travel photograph. (2) A red panda sleeping on a mossy tree branch, wildlife photograph. (3) A glass perfume bottle surrounded by lavender flowers, elegant studio photograph.

NFE = 12. (1) A brass pocket watch resting on an open book beside a candle, cinematic still life. (2) A cozy bakery storefront glowing on a rainy cobblestone street at night. (3) A medieval stone bridge crossing a misty river at sunrise, cinematic landscape.

NFE = 16. (1) A snowy owl perched on a weathered fence post, detailed wildlife photograph. (2) A small corgi wearing a yellow raincoat on a wet city sidewalk, street photograph. (3) A fox standing in a snowy birch forest, detailed wildlife photograph.

NFE = 20. (1) A snowy owl perched on a weathered fence post, detailed wildlife photograph. (2) A lighthouse on a rocky island during a dramatic storm, seascape painting. (3) A fox standing in a snowy birch forest, detailed wildlife photograph.

FLUX.1-dev prompts

NFE = 5. (1) An elderly violin maker at work. (2) A corgi barista behind a cafe counter. (3) A glass teapot containing a tiny galaxy. (4) A snowy owl librarian in a candlelit library. (5) A clock-tower snow globe. (6) A black cat operating a printing press.

NFE = 7. (1) A red panda astronaut tending plants. (2) A whale-shaped hot-air balloon over the sea. (3) An alpine train crossing a snowy viaduct. (4) A stallion running through the desert. (5) An astronaut observing a sunflower on Mars. (6) Three porcelain cranes on a black background.

FLUX.1-dev teaser (Figure 1; NFE = 7)

The eleven panels are listed in reading order, from the upper left across the composition and then downward. The samples use guidance 3.5 and the supplied 7-NFE coefficients.

1. Crystal whale city. A giant crystal whale floating through a neon city at night, trailing a garden of glowing flowers, cinematic fantasy.

2. Cosmic fashion. A glamorous woman in a jeweled space helmet holding a miniature sun and moon beside her face, vivid fashion editorial.

3. Batman and robot. Batman and a small orange robot sharing noodles at a neon street stall, cinematic movie still.

4. Prismatic windmill. A monumental windmill with four transparent stained-glass sails casting rainbow light across endless tulip fields, dramatic panorama.

5. Giraffe flower market. A giraffe in an astronaut suit buying flowers at a lantern-lit night market, surreal photograph.

6. Cloud-koi city. A futuristic coastal city where enormous koi fish swim through the clouds above sky trains, anime panorama.

7. Umbrella and whales. An astronaut surfing down a colossal waterfall on a red umbrella while whales leap through the mist, surreal photograph.

8. Generative-model stage. A lively robot orchestra on a futuristic theater stage beneath a giant neon sign reading "GENERATIVE MODEL," colorful animated film.

9. Pastry city. Tiny chefs building a city from croissants, macarons and chocolate, whimsical food photography.

10. Red-panda wizard. A cheerful red panda wizard casting rainbow spells in a pixel-art forest, retro game illustration.

11. Origami dragon. A porcelain astronaut riding a pink origami dragon through paper clouds, handcrafted poster.

Figure 13: Prompts used for the Stable Diffusion v1.5 and FLUX.1-dev qualitative comparisons, including the opening figure (Figure 1).