FUTURE DISCREPANCY KNOWLEDGE DISTILLATION FOR DIFFUSION MODEL COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion model capacity distillation trains a compact Student to imitate a high-capacity Teacher, reducing the computation required for each network evaluation during generation. Classic knowledge distillation (KD) typically matches their current predictions at states independently constructed by noising real data. During generation, however, the current state is produced recursively through preceding model predictions and state updates. Different Teacher and Student sampling histories can therefore yield different states for local supervision. Under a common local objective, we compare Teacher- and Student-rollout states and find that their relative performance depends on the Teacher–Student configuration. Even with the training-state source fixed, local prediction matching measures only the immediate discrepancy without accounting for its evolution during subsequent sampling. Local errors of similar magnitude can produce different future state deviations under shared reference dynamics. To address this gap, we introduce *Future Discrepancy* (FD). Starting from a common state, the Teacher and Student each perform one state update. We then propagate the two next states with the same frozen Teacher and use their discrepancy at a selected future position to supervise the current Student state update. FD applies to either Teacher- or Student-rollout states. We use *Temporal Sampling* to reduce Teacher continuation queries during training and combine FD with complementary classic KD on independently noised data states to form *Future-Discrepancy Knowledge Distillation* (FD-KD). Experiments across Student construction strategies, Teacher–Student configurations, and image generation tasks demonstrate the gains of FD over local supervision using the same state source and the effectiveness of the complete FD-KD method.

1 INTRODUCTION

Generative models based on diffusion or flow matching have achieved substantial progress in image synthesis (Ho et al., 2020; Rombach et al., 2022; Lipman et al., 2023). However, their inference typically requires repeated network evaluations over multiple sampling steps (Salimans & Ho, 2022; Song et al., 2023), while the large capacity of networks further increase computational cost and latency, limiting their application in practical settings (Rombach et al., 2022; Ma et al., 2024). In this work, we focus on compressing model capacity through knowledge distillation by transferring knowledge from a high-performance pretrained Teacher with large model capacity to a lightweight Student (Hinton et al., 2015; Kim et al., 2024; Han et al., 2026; Fang et al., 2025). Our goal is to substantially reduce the computational cost of each inference step while retaining the Teacher's generation quality. We refer to this task as *diffusion model capacity distillation*.

Diffusion model capacity distillation has attracted increasing attention in both class-conditional and text-conditional image generation with U-Net and DiT backbones (Yang et al., 2023; Rombach et al., 2022; Kim et al., 2024; Fang et al., 2025). Most methods follow conventional knowledge distillation (Hinton et al., 2015) by independently adding noise to real samples to construct training states. Given the same state, timestep, and condition, the Student is trained to match the Teacher's noise or velocity predictions or intermediate features (Kim et al., 2024; Han et al., 2026).

During inference, however, model predictions are embedded in recursive sampling trajectories (Ho et al., 2020). At each timestep, preceding state updates determine the current state; the model

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.

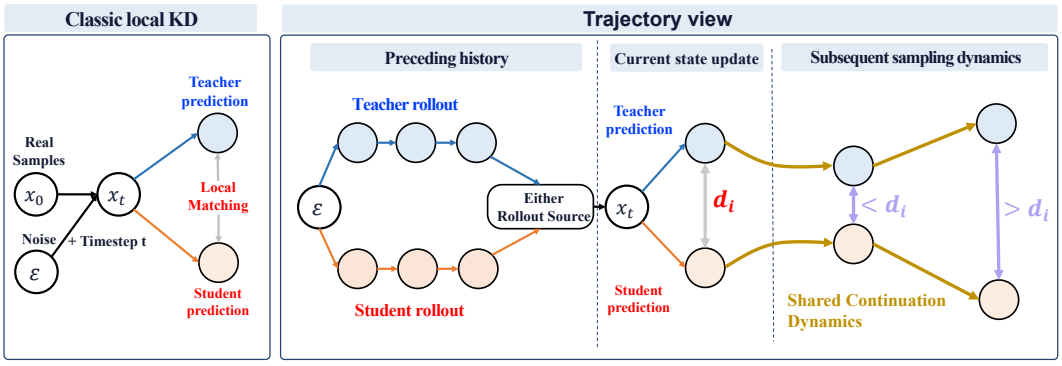


Figure 1: Overview of classic KD and the trajectory perspective. (a) Classic KD matches local predictions on independently noised data states. (b) Preceding Teacher or Student sampling history determines the current state. Teacher and Student updates from that state introduce one local difference d_i. Changing state separation schematically illustrates possible attenuation and amplification under shared subsequent reference dynamics. FD evaluates the discrepancy at a future position.

predicts from that state, and the sampler uses its prediction to produce the next state. Because a capacity-limited Student cannot reproduce the Teacher at every state, the discrepancies between its next states and the Teacher's can affect the states visited later in its trajectory. Consequently, independently noised real-data states may not adequately represent the states encountered by the Student during sampling (Ning et al., 2023; 2024). This raises a central question for diffusion model capacity distillation: *On which state distribution should the Student be supervised?*

From the perspective of generation trajectories, a natural way to make the Student approximate the Teacher during recursive sampling is to compare their trajectory distributions. For stochastic Markov sampling with a shared initial noise distribution, trajectory KL decomposes into step-wise conditional KL terms. The forward divergence, $D_{\mathrm{KL}}(P_{\mathcal{T}}\|P_{\mathcal{S}})$, averages these terms over Teacher-visited states, whereas the reverse divergence, $D_{\mathrm{KL}}(P_{\mathcal{S}}\|P_{\mathcal{T}})$, averages them over Student-visited states. These two state sources motivate *Teacher-Policy Distillation* (TPD) and *On-Policy Distillation* (OPD) (Agarwal et al., 2024; Li et al., 2026; Fang et al., 2026). Under the same local objective, TPD and OPD differ in whether the current state comes from a Teacher or Student trajectory. Teacher states provide a Teacher-defined reference but may miss regions reached after Student deviations; Student states better reflect the Student's deployment distribution, although their coverage changes during training. For a capacity-limited Student, these sources emphasize different regions of state space, so TPD and OPD need not have a fixed performance ordering across Teacher–Student gaps, initializations, and training configurations.

However, replacing independently noised data states with Teacher or Student trajectory states for local distillation only changes the state distribution over which the local loss is averaged. Existing local prediction losses measure the immediate discrepancy when an error occurs, but do not explicitly account for the interaction between its direction and the subsequent sampling dynamics, as illustrated in Figure 1. Consequently, two local errors with similar magnitudes can be amplified, attenuated, or redirected to different degrees under the same subsequent reference dynamics, eventually producing substantially different future state deviations.

To this end, we propose *Future Discrepancy* (FD), which extends the evaluation of current Teacher and Student state updates from an immediate local discrepancy to a future state deviation under shared reference dynamics. Given a state from either a Teacher or Student trajectory, the two models each perform one state update at the same timestep and condition. We propagate the resulting next states with the same frozen Teacher to a selected future position and define their state difference there as FD. FD therefore isolates how this single discrepancy between next states evolves under the specified reference dynamics and uses the future deviation to supervise the current Student state update. In state space, evaluating the immediate next states recovers their local discrepancy, while later positions capture its evolution under subsequent dynamics.

Applying FD along a multistep trajectory creates many candidate pairs of current and future evaluation positions, so dense supervision requires repeated frozen-Teacher continuations and adds training cost. At fixed future intervals, neighboring FD signals show local temporal redundancy, motivating

Temporal Sampling to select sparse positions while balancing signal coverage and Teacher evaluations. Together, temporally sampled trajectory-state FD and complementary conventional KD form *Future-Discrepancy Knowledge Distillation* (FD-KD). Teacher and Student trajectory states give the two configurations FD-KD($\mathcal{T}$) and FD-KD($\mathcal{S}$), respectively. Teacher continuation and Temporal Sampling are used only during training; inference retains the compact Student's original multistep sampling process without additional parameters or network evaluations. Our main contributions are summarized below:

- We compare independently noised data states, Teacher trajectory states, and Student trajectory states under a common local objective. The comparison reveals distinct fitting and generation characteristics and configuration-dependent relative performance for the two trajectory sources.
- We propose FD to evaluate current Teacher and Student state updates through the future state deviation produced by a shared frozen-Teacher continuation. This target distinguishes errors with similar local magnitudes by their different future deviations under the specified reference dynamics.
- We design Temporal Sampling to reduce Teacher continuation queries and combine sampled FD with conventional KD to form FD-KD. We evaluate the method across Teacher–Student configurations on ImageNet-1K and MS-COCO.

2 Related Work

Knowledge Distillation for Diffusion Model Compression Existing approaches reduce parameter count and per-step cost through lightweight architectures, structured pruning (Fang et al., 2023), block removal (Kim et al., 2024; Men et al., 2025), or depth reduction (Fang et al., 2025), then use Teacher supervision to recover generation quality while preserving iterative sampling. One line of work studies which Teacher knowledge to match at a noisy state, including predictions, intermediate representations, and sensitivity to local perturbations (Yang et al., 2023; Kim et al., 2024; Lee et al., 2024; Zheng & Shlizerman, 2026). A second line changes the distillation data, condition coverage, or fitting procedure to improve training under limited data or large Teacher–Student capacity gaps (Kim et al., 2025; Xiang et al., 2025; Han et al., 2026). These methods primarily evaluate local consistency at the current noisy state.

Training-State Sources under Recursive Sampling In iterative generation, independently constructed noisy training inputs may not fully cover recursively visited states, while step-wise errors also alter subsequent inputs (Ning et al., 2023; 2024). DAgger (Ross et al., 2011) addresses analogous distribution shift in imitation learning by querying the expert on learner-visited states and iteratively aggregating the resulting supervision. This principle was later adopted in autoregressive knowledge distillation: ImitKD (Lin et al., 2020), MiniLLM (Gu et al., 2024), and GKD (Agarwal et al., 2024) obtain Teacher feedback on Student-generated sequences or contexts, with different choices of training distribution and matching objective. On-policy distillation has also been used to integrate multiple capabilities in large language models (Yang et al., 2025; Ma et al., 2026). DiffusionOPD (Li et al., 2026), Flow-OPD (Fang et al., 2026), and DanceOPD (Zhou et al., 2026) pursue related capability-integration goals by aggregating tasks, rewards, or generative capabilities along Student-generated trajectories. D-OPSD (Jiang et al., 2026) instead targets continual fine-tuning of step-distilled diffusion models. Their state sources vary jointly with tasks, Teachers, or optimization designs. We instead examine diffusion-model capacity distillation through recursive sampling. We first compare training-state sources under a common local loss, then introduce Future Discrepancy to evaluate how Teacher–Student state differences evolve under shared frozen-Teacher continuation.

Temporal Supervision in Diffusion Distillation Sampling-step distillation compresses the Teacher's multi-step denoising into Student state updates over longer time intervals, whereas consistency modeling constrains predictions or trajectory mappings across diffusion times. These methods primarily reduce network evaluations and enable one- or few-step generation (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2025; Zhou et al., 2024). While retaining the compact Student's original multi-step sampling process, we evaluate the current discrepancy between Teacher and Student next states through its evolution during subsequent sampling. Some methods in sequence prediction and imitation learning formulate structured prediction as a series of interdependent decisions and use sequence-level rollout outcomes to supervise current choices (Chang et al., 2015). Extending DAgger's supervision on learner-visited states (Ross et al., 2011), AggreVaTe (Ross & Bagnell, 2014)

evaluates candidate actions by estimating the cumulative task cost of taking each action and then following the expert. Future Discrepancy follows this principle, using future state differences from subsequent sampling with the same frozen Teacher rather than task-specific cumulative costs.

3 Future-Discrepancy Knowledge Distillation

The input state is determined by preceding sampling history, the current Teacher and Student state updates create a local split, and subsequent dynamics transform this split. We place this local comparison within sampling trajectories and define Future Discrepancy (FD) under shared reference dynamics. Finally, we select temporal targets along the trajectory and combine trajectory-state FD with classic KD on independently noised data states to form FD-KD.

3.1 Knowledge Distillation in Diffusion and Flow Models

Let $\mathcal{T}$ be a frozen pretrained Teacher and $\mathcal{S}_\theta$ a compact Student. At sampling time t, both models receive a state x_t and condition c, and produce a native prediction $f_M(x_t, t, c)$ for $M \in \{\mathcal{T}, \mathcal{S}\}$.

Specifically, for diffusion models, f_M typically predicts noise or velocity and parameterizes a reverse transition distribution from x_t to x_{t-1} (Ho et al., 2020; Song et al., 2021).

For flow matching, a velocity field $v_M(x, t, c)$ generates the probability path p_t^M (Lipman et al., 2023). For adjacent sampler times t and t', an Euler step is $x_{t'}^M = x_t + (t' - t)v_M(x_t, t, c)$. See Appendix A.1 for the transition distributions and their conditional KL relations.

For distillation, a training state is independently constructed from a data sample as $x_t = a(t)x_{\text{data}} + b(t)\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $a(t), b(t)$ follow the diffusion noise schedule or flow matching probability path. Teacher and Student receive the same x_t, t, and c, yielding

$$\ell_t^{\text{KD}} = \|f_{\mathcal{S}}(x_t, t, c) - \text{sg}[f_{\mathcal{T}}(x_t, t, c)]\|_2^2, \qquad \mathcal{L}_{\text{KD}} = \mathbb{E}_{x_{\text{data}}, \epsilon, t}\left[\ell_t^{\text{KD}}\right]. \tag{1}$$

We optimize this unweighted prediction MSE for both parameterizations (Ho et al., 2020; Li et al., 2026; Fang et al., 2026). Similarly, feature distillation follows the same form with f_M replaced by the selected intermediate representation.

3.2 Sampling History and the Current Teacher–Student State Update

We index positions along the generation direction by $i \in \{0, \ldots, N-1\}$ and absorb the corresponding sampling time into the update at each position. Let $\mathcal{F}_i^M$ denote the state update of model $M \in \{\mathcal{T}, \mathcal{S}\}$, so a complete rollout satisfies $x_{i+1}^M = \mathcal{F}_i^M(x_i^M)$. Teacher and Student rollouts induce state visitation distributions $q_i^{\mathcal{T}}$ and $q_i^{\mathcal{S}}$ even when they start from the same noise distribution. See Appendix A.1 for the trajectory KL decomposition.

Teacher-Policy Distillation (TPD) and On-Policy Distillation (OPD) accordingly reuse the per-state loss in Eq. (1) on Teacher and Student rollout states (Li et al., 2026; Fang et al., 2026). Writing this loss at position i as ℓ_i^{KD} gives

$$\mathcal{L}_{\text{TPD}} = \mathbb{E}_{i,\, x_i \sim q_i^{\mathcal{T}}}[\ell_i^{\text{KD}}], \qquad \mathcal{L}_{\text{OPD}} = \mathbb{E}_{i,\, x_i \sim q_i^{\mathcal{S}}}[\ell_i^{\text{KD}}]. \tag{2}$$

Given a detached common state x_i, the Teacher and Student execute the current update with the same time, condition, and coupled sampling randomness when applicable:

$$z_i^M = \mathcal{F}_i^M(x_i), \quad M \in \{\mathcal{T}, \mathcal{S}\}, \qquad d_i = z_i^{\mathcal{S}} - z_i^{\mathcal{T}}. \tag{3}$$

Here $z_i^{\mathcal{T}}$ and $z_i^{\mathcal{S}}$ are the two next states, and d_i is the state-space perturbation introduced by the current Teacher–Student split. For an Euler update, d_i is the velocity difference scaled by the step size. For a shared-covariance diffusion transition distribution, its mean difference is likewise a linear function of the noise or velocity prediction difference.

3.3 Future Discrepancy under Shared Reference Dynamics

The states $z_i^{\mathcal{S}}$ and $z_i^{\mathcal{T}}$ form a controlled split from the same input state, time, and condition. We propagate both states with the same frozen Teacher to evaluate how this split evolves. For $i < j$, the

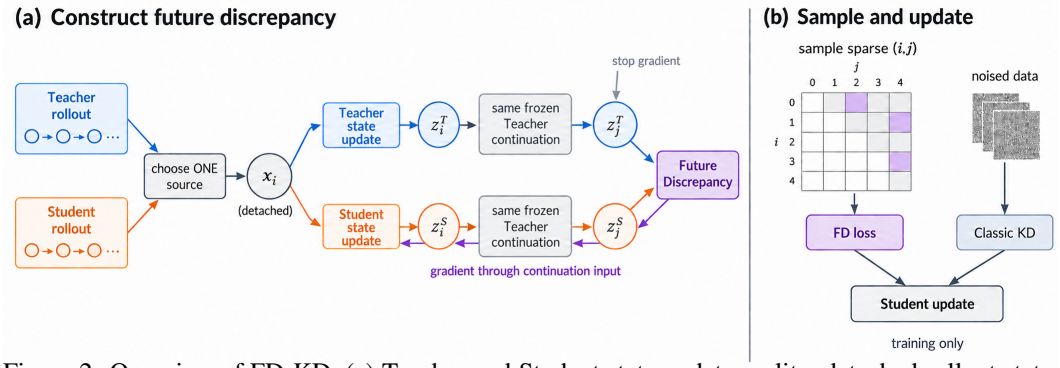


Figure 2: Overview of FD-KD. (a) Teacher and Student state updates split a detached rollout state. Shared frozen-Teacher continuation produces endpoints $z^M_{i\to j}$, abbreviated as z^M_j in the schematic. (b) The $N = 5$ schematic shows valid update-index pairs $i < j \le N-1$ and sparse selection from this support; the temporal-sampling experiments fix $j = i+1$ ($h = 1$). Selected FD terms combine with classic KD on independently noised data for one Student update.

Teacher continuation is

$$\mathcal{R}^{\mathcal{T}}_{i\to j} = \mathcal{F}^{\mathcal{T}}_j \circ \cdots \circ \mathcal{F}^{\mathcal{T}}_{i+1}. \tag{4}$$

We set $\mathcal{R}^{\mathcal{T}}_{i\to i} = \mathrm{Id}$. For either branch $M \in \{\mathcal{S}, \mathcal{T}\}$, its endpoint at position j is $z^M_{i\to j} = \mathcal{R}^{\mathcal{T}}_{i\to j}(z^M_i)$. The two branches share the Teacher state updates, conditions, and coupled sampling randomness, with each update evaluated at its branch state. The state discrepancy and supervision target at future position j are

$$G_{i\to j} = z^{\mathcal{S}}_{i\to j} - z^{\mathcal{T}}_{i\to j}, \qquad \ell^{\mathrm{FD}}_{i\to j} = D\big(z^{\mathcal{S}}_{i\to j}, \mathrm{sg}[z^{\mathcal{T}}_{i\to j}]\big), \tag{5}$$

where D is a state-space discrepancy, implemented as mean squared error, and sg stops gradients through the reference branch. Since $z^{\mathcal{S}}_i = z^{\mathcal{T}}_i + d_i$, FD admits the exact form

$$G_{i\to j} = \mathcal{R}^{\mathcal{T}}_{i\to j}(z^{\mathcal{T}}_i + d_i) - \mathcal{R}^{\mathcal{T}}_{i\to j}(z^{\mathcal{T}}_i). \tag{6}$$

The shared nonlinear continuation can amplify, attenuate, or redirect the current perturbation as the two branch states provide different inputs to subsequent Teacher updates.

First-order view. Equation (6) is the finite nonlinear construction used by FD. To characterize its local geometry, we linearize the frozen-Teacher continuation around the Teacher split output. Let $J_{i\to j} = \nabla_z \mathcal{R}^{\mathcal{T}}_{i\to j}(z)\big|_{z=z^{\mathcal{T}}_i}$. With a locally Lipschitz Jacobian and small d_i,

$$G_{i\to j} = J_{i\to j} d_i + \mathcal{O}(\|d_i\|_2^2), \qquad \|G_{i\to j}\|_2^2 \approx d_i^\top J_{i\to j}^\top J_{i\to j} d_i. \tag{7}$$

Immediate state-space squared error applies the isotropic metric $\|d_i\|_2^2$, whereas the future discrepancy is shaped by the direction-dependent metric $J_{i\to j}^\top J_{i\to j}$. Local perturbations with similar magnitudes can therefore produce different future deviations. Equation (7) provides an analytical view of this direction dependence. Training evaluates the finite discrepancy in Eq. (5) through ordinary forward continuation; the Jacobian and its quadratic form are not instantiated in the optimization. When $j = i$, the continuation is the identity, giving $G_{i\to i} = d_i$ and $\ell^{\mathrm{FD}}_{i\to i} = D(z^{\mathcal{S}}_i, \mathrm{sg}[z^{\mathcal{T}}_i])$. Positive continuation lengths evaluate the same split at progressively later positions.

3.4 Trajectory-Wide Temporal Sampling

For candidate split positions $\mathcal{I} \subseteq \{0, \ldots, N-2\}$, define the valid support and its sampling distribution as

$$\Omega_{\mathrm{FD}} = \{(i,j) : i \in \mathcal{I},\ i < j \le N-1\}, \qquad (i,j) \sim \pi_{\mathrm{FD}}. \tag{8}$$

Indices i, j label updates, with $h = j - i$ additional Teacher updates after the split. At fixed $h \ge 1$, valid splits satisfy $i \in \mathcal{I}$ and $i \le N-1-h$; $h = 0$ lies outside Ω_{FD}. See Appendix B for sampling strategies. For rollout source $s \in \{\mathcal{T}, \mathcal{S}\}$, the resulting FD component is

$$\mathcal{L}^s_{\mathrm{FD}} = \mathbb{E}_{(i,j)\sim\pi_{\mathrm{FD}},\, x_i\sim q^s_i}\left[\ell^{\mathrm{FD}}_{i\to j}\right]. \tag{9}$$

3.5 FD-KD Objective and Training

FD provides the primary supervision on model-generated trajectory states, while classic KD supplements it with local Teacher targets on independently noised data states. The complete objective is

$$\mathcal{L}^{s}_{\text{FD-KD}} = \mathcal{L}^{s}_{\text{FD}} + \lambda_{\text{KD}}\mathcal{L}_{\text{KD}}, \tag{10}$$

Here $s \in \{\mathcal{T}, \mathcal{S}\}$ sets the rollout source, giving FD-KD($\mathcal{T}$) or FD-KD($\mathcal{S}$). Teacher parameters remain frozen; FD gradients pass through continuation inputs to the current Student state update. Student inference is unchanged. Training details are in Appendix B.

4 Experiments

4.1 Experimental Setup

We use class-conditional ImageNet-1K (Deng et al., 2009) as the main analysis setting and evaluate complete FD-KD on text-conditional MS-COCO (Lin et al., 2014). All results in ImageNet use 50K generated images and report FID, Inception Score (IS), Precision, and Recall (Heusel et al., 2017; Salimans et al., 2016; Kynkäänniemi et al., 2019). MS-COCO evaluation uses 30K generated images and reports FID, IS, and CLIP score (Hessel et al., 2021).

The ablations use SiT-B (Ma et al., 2024) as Teacher and two Students: depth-pruned SiT-B-D6 and independently pretrained SiT-S. Their constructions follow TinyFusion's depth-pruning setting (Fang et al., 2025) and GKD's separately pretrained setting (Agarwal et al., 2024), respectively. Unless otherwise specified, experiments with OPD, TPD, and their FD variants use 10-step sampling, batch size 64, and 25K training iterations. We use 50-step Heun sampling for inference and CFG 1.5 for all test results. We report means over three runs. Cost denotes seconds per training iteration, including rollout generation, target and continuation evaluations, loss computation, backpropagation, and the optimizer update. More implementation details are given in Appendix B.

4.2 Method Analysis and Ablation Studies

4.2.1 Training States from Sampling History

We first compare training-state sources under the same local prediction matching objective. Table 1 reports TPD and OPD for Student SiT-B-D6 and SiT-S. Classic KD on independently noised data states provides a conventional reference. Under local supervision, OPD achieves lower FID than TPD for both Students. IS and Precision also increase, while Recall decreases. Thus, the state source changes the observed quality–coverage balance.

Figure 3a traces state deviations during recursive sampling. It plots latent-state RMS gaps to paired Teacher states. A one-step reset of the initial Student remains close to the Teacher, whereas its recursive trajectory accumulates a larger gap. This pattern shows that training states carry deviations from preceding updates. In this diagnostic, the TPD and OPD curves remain below the classic-KD curve. Thus, TPD and OPD maintain closer alignment with the Teacher trajectory than classic KD during recursive sampling, as measured by latent-state RMS deviation.

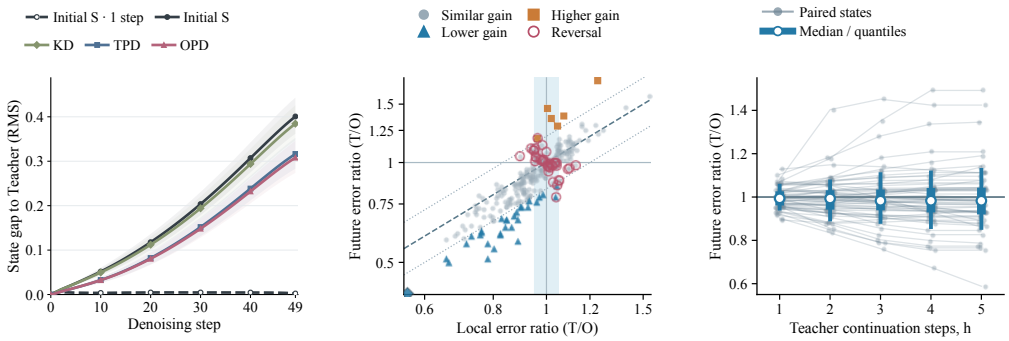
Table 1: Training-state sources and supervision targets for two compact Students on ImageNet. FD rows use shared Teacher continuation with $h = 1$ and all valid candidate positions.

			Pruned: SiT-B → SiT-B-D6				Pretrained: SiT-B → SiT-S			
Method	State source	Target	FID↓	IS↑	Prec.↑	Rec.↑	FID↓	IS↑	Prec.↑	Rec.↑
Student	–	–	207.62	2.18	0.02	0.00	31.88	50.22	0.55	0.55
Classic KD	Real Data	Local	24.29	66.14	0.61	0.52	26.30	60.15	0.60	0.53
TPD	Teacher	Local	16.09	109.49	0.72	0.36	14.12	114.31	0.75	0.37
OPD	Student	Local	15.64	119.07	0.74	0.31	13.34	126.31	0.77	0.33
FD($\mathcal{T}$)	Teacher	FD	15.07	112.94	0.74	0.35	13.56	116.47	0.76	0.37
FD($\mathcal{S}$)	Student	FD	14.29	125.57	0.76	0.30	12.58	129.87	0.78	0.34

4.2.2 LOCAL AND FUTURE SUPERVISION TARGETS

We next fix the trajectory-state source and compare local prediction matching with FD supervision. Table 1 compares TPD with FD($\mathcal{T}$) on Teacher states and OPD with FD($\mathcal{S}$) on Student states. Across both Student constructions and trajectory sources, FD consistently outperforms the corresponding local baseline in FID, supporting the benefit of future-state supervision.

For SiT-B $\rightarrow$ SiT-B-D6, we evaluate TPD and OPD checkpoints trained for 25K updates on common Teacher-rollout states. At each common state, we measure the Teacher–Student prediction error and apply each model's state update. We then continue both next states with the same frozen Teacher to measure their future discrepancy. Figure 3b compares the TPD/OPD ratios of local prediction errors and future discrepancies. Deviations from the diagonal and ranking reversals show that local and future errors can favor different checkpoints. For Figure 3c, we select comparisons with similar TPD and OPD local prediction errors. We then extend the shared Teacher continuation and track each selected pair's future error ratio. Although the median remains near equality, individual ratios become more dispersed at longer horizons. Thus, similar local errors can lead to different sample-level future discrepancies, which FD explicitly evaluates through shared Teacher dynamics.



(a) State gap over 50 steps (b) Local versus future error (c) Matched-local evolution

Figure 3: Sampling-history and future-discrepancy analysis for SiT-B $\rightarrow$ SiT-B-D6 with 25K-trained checkpoints. (a) RMS state gaps to paired Teacher states for recursive sampling. (b) TPD/OPD local versus future error ratios under shared Teacher continuation. (c) Future error ratios of matched pairs across continuation steps.

4.2.3 REFERENCE CONTINUATION AND FUTURE HORIZON

Table 2 compares shared Student, shared Teacher, and separate branch-specific continuation on Student-trajectory states at $h = 1$, holding all other settings fixed. Shared Teacher continuation achieves the lowest FID and highest IS, but costs more than shared Student continuation. We view the pretrained, frozen Teacher as a more reliable and stable reference for future dynamics; Student continuation changes as its parameters are updated during training. We therefore use shared Teacher continuation in subsequent experiments.

With shared Teacher continuation, Table 3 compares $h \in \{0, 1, 2, 3\}$ at all valid candidate positions for each horizon. The $h = 0$ row reuses OPD with matched training implementation and optimization settings. At 25K updates, $h = 1$ reduces FID from 15.64 to 14.29, with cost increasing from 1.36 to 1.79 s/iter. The $h = 2$ setting offers modest further gains at higher cost; $h = 3$ worsens quality. We use $h = 1$ for the remaining analyses as a quality–cost trade-off.

4.2.4 TEMPORAL SAMPLING AND TRAINING COST

In Table 4, Dense repeats Table 3's $h = 1$ result using all nine valid positions; sparse sampling selects R positions per Student trajectory. Global Random samples uniformly; Stratified samples one position from each of R consecutive temporal groups. Figure 4 shows cross-position cosine similarity and signal retention under sparse queries. At $R = 3$, Stratified achieves FID 14.89 versus 15.22 for Global Random at the same cost of 0.72 s/iter. Relative to Dense, it reduces iteration time from 1.79 to 0.72 s/iter (59.8%), with FID increasing from 14.29 to 14.89. We use Stratified sampling with $R = 3$ in subsequent FD experiments unless otherwise specified.

Table 2: FD continuation on Student-trajectory states ($h = 1$). Student/Teacher denote shared continuation; Separate uses each branch's model.

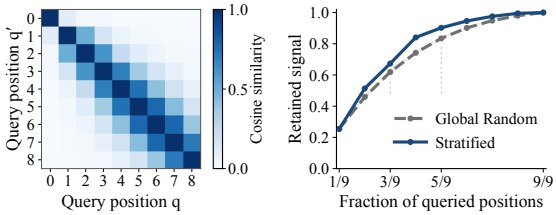
Continuation	FID↓	IS↑	Prec.↑	Rec.↑	Cost↓
Student	15.47	118.45	0.74	0.31	**1.54**
Teacher	**14.29**	**125.57**	**0.76**	0.30	1.79
Separate	21.00	88.41	0.67	**0.38**	1.62

Table 3: Continuation steps on Student-trajectory states at all valid positions. $h = 0$ reuses OPD.

Step h	FID↓	IS↑	Prec.↑	Rec.↑	Cost↓
0	15.64	119.07	0.74	**0.31**	**1.36**
1	14.29	125.57	0.76	0.30	1.79
2	**14.10**	**126.48**	**0.77**	0.30	2.22
3	15.42	118.47	0.74	0.31	2.61

Table 4: Temporal sampling for FD($\mathcal{S}$) on SiT-B → SiT-B-D6 ($h = 1$). Ret. measures the fraction of dense FD signal energy as shown in Figure 4.

Policy	R	Ret.↑	FID↓	IS↑	Prec.↑	Rec.↑	Cost↓
Dense	9	100.0%	**14.29**	**125.57**	**0.76**	**0.30**	1.79
Random	1	25.5%	18.77	105.12	0.72	0.28	**0.38**
Random	3	61.9%	15.22	122.31	0.76	0.30	0.72
Stratified	3	67.3%	14.89	124.92	0.76	0.29	0.72
Stratified	5	90.2%	14.56	124.41	0.76	0.30	1.06



(a) Query similarity (b) Retained signal

Figure 4: FD signal analysis at $h = 1$. (a) Cosine similarity across sampling positions. (b) Retained signal versus queried-position fraction.

4.2.5 Scaling Across Teacher–Student Configurations

After selecting the FD construction and temporal sampling strategy, we examine two configuration axes. Student scaling holds SiT-B fixed and compares depth-pruned SiT-B-D8, SiT-B-D6, and SiT-B-D4 Students. Teacher scaling holds independently pretrained SiT-S fixed and compares SiT-B, SiT-L, and SiT-XL Teachers. All FD results use temporal sampling. Figure 5 compares local and FD supervision in FID and IS within each trajectory source. Across all six configurations in Figure 5, FD($\mathcal{T}$) achieves lower FID and higher IS than TPD, while FD($\mathcal{S}$) improves both metrics over OPD. The magnitude of these gains and the ordering of the two FD sources vary across configurations. The preferred local source also changes with configuration. TPD has lower FID for SiT-B-D4 and under the SiT-XL Teacher, whereas OPD has lower FID in the remaining settings.

Figure 6 maps these configurations by initial trajectory gap and Student/Teacher parameter ratio, with circle colors indicating the lower-FID local method. The background and candidate boundaries use a historical measurement protocol. These preferences, visualized in Figure 6, indicate that the choice of local state source depends on the Teacher–Student combination. Appendix E provides qualitative comparisons in Figure 7.

4.2.6 Composition of the Full FD-KD Objective

The preceding analyses use pure FD supervision on trajectory states. We now add classic KD on independently noised data states to form the complete FD-KD objective. Table 5 shows that both configurations achieve lower FID than classic KD alone and their corresponding FD baselines. Adding classic KD also increases Recall but reduces Precision for both trajectory sources, while lowering IS for Student-trajectory states.

Table 5: Effect of combining FD with classic KD for SiT-B → SiT-B-D6.

Method	KD	FID↓	IS↑	Prec.↑	Rec.↑	Cost↓
Classic KD	✓	24.29	66.14	0.61	0.52	0.38
FD($\mathcal{T}$)		15.73	111.62	0.74	0.34	–
FD-KD($\mathcal{T}$)	✓	14.98	111.67	0.72	0.39	1.27
FD($\mathcal{S}$)		14.89	124.92	0.76	0.30	0.72
FD-KD($\mathcal{S}$)	✓	14.35	116.66	0.73	0.37	1.14

4.3 Benchmark Results and Generated Samples

Class-conditional ImageNet-1K. Table 6 compares compact models for DiT-XL/2 → TinyDiT-D14 at 500K parameter updates. With the same Student architecture and update budget, FD-KD($\mathcal{T}$) and FD-KD($\mathcal{S}$) achieve FID 2.61 and 2.59, respectively, compared with 2.86 for TinyFusion and 2.79 for LIFT + PLACE. Relative to TinyFusion, FD-KD($\mathcal{T}$) raises IS from 234.50 to 241.40 while

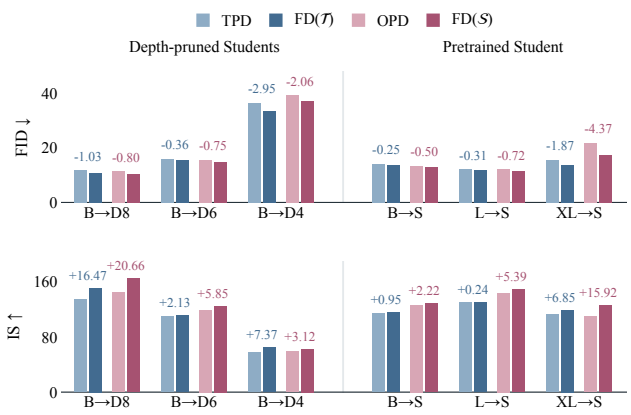


Figure 5: FID (top) and IS (bottom) under Student scaling (SiT-B fixed) and Teacher scaling (SiT-S fixed). Each configuration compares TPD with FD($\mathcal{T}$) and OPD with FD($\mathcal{S}$).

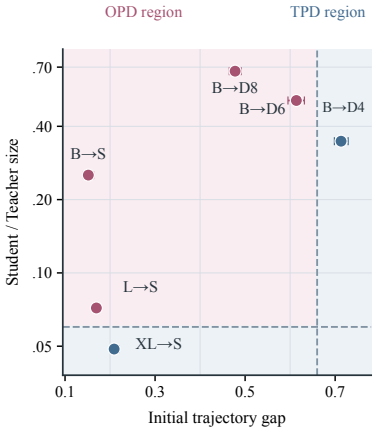


Figure 6: Initial trajectory gap versus Student/Teacher parameter ratio. Circle colors indicate OPD/TPD.

retaining Precision 0.82 and Recall 0.55. FD-KD($\mathcal{S}$) reaches IS 249.84 and Precision 0.83, with lower Recall of 0.53.

Text-to-image generation on MS-COCO. Table 7 compares methods within the BK-Base-v2 and BK-Tiny-v2 compact Student architectures, each trained for the listed 50K iterations. FD-KD($\mathcal{T}$) attains FID 14.63 and 13.79, respectively, below both BK-SDM and LIFT + PLACE within each block. It also has the highest IS and CLIP score in both blocks. These comparisons show improvements across the reported generation metrics for two text-to-image Student configurations. Additional visualizations are shown in Figure 8 of Appendix E.

Table 6: Class-conditional ImageNet-1K comparison for DiT-XL/2 → TinyDiT-D14. Iter. denotes parameter updates.

Model	Method	Iter.	FID↓	IS↑	Prec.↑	Rec.↑
DiT-XL/2	Teacher	–	2.27	278.20	0.83	0.57
DiT-D14	TinyFusion	500K	2.86	234.50	0.82	0.55
	LIFT+PLACE	500K	2.79	235.60	0.82	0.55
	FD-KD($\mathcal{T}$)	500K	2.61	241.40	0.82	0.55
	FD-KD($\mathcal{S}$)	500K	2.59	249.84	0.83	0.53

Table 7: MS-COCO text-to-image comparison within BK-Base-v2 and BK-Tiny-v2 Students. All metrics use 30K generated images.

Model	Method	Iter.	FID↓	IS↑	CLIP↑
SD2.1	Teacher	–	13.93	35.93	0.3075
BK-Base-v2	BK-SDM	50K	15.85	31.70	0.2868
	LIFT + PLACE	50K	16.72	31.82	0.2900
	FD-KD($\mathcal{T}$)	50K	14.63	32.01	0.2902
	FD-KD($\mathcal{S}$)	50K	14.56	32.01	0.2902
BK-Tiny-v2	BK-SDM	50K	15.68	31.64	0.2897
	LIFT + PLACE	50K	14.60	31.75	0.2916
	FD-KD($\mathcal{T}$)	50K	13.79	32.86	0.2937

5 Conclusion

We study diffusion model capacity distillation through the sampling history that determines training states and the future evolution of Teacher–Student discrepancies. Future Discrepancy (FD) evaluates a split from a common state at a future position under shared frozen-Teacher continuation and, with temporal sampling and classic KD, forms FD-KD. This construction distinguishes local errors of similar magnitude by how they evolve under the same subsequent dynamics. In the reported configurations, FD improves generation over matched local supervision. Temporal sampling reduces Teacher queries and training time, and full FD-KD is competitive in class-conditional and text-conditional generation. FD-KD requires selecting the rollout-state source for each Teacher–Student configuration, as we provide no general automatic selection rule. Frozen-Teacher continuation also adds training cost that temporal sampling reduces but does not eliminate, although Student inference remains unchanged. Together, these findings support using future state deviations to guide model capacity distillation within recursive sampling trajectories.

Reproducibility Statement

Appendix A gives the trajectory-KL derivation and the first-order interpretation of FD. Appendix B records Student architectures and training, evaluation, and cost settings. Appendix C specifies the sampling-history, local-to-future, and temporal-signal diagnostics in Figures 3 and 4. Appendix D specifies repeated-run and sensitivity tests for the final FD-KD recipe.

Use of Large Language Models

In accordance with the ICLR policy, we report our use of large language models (LLMs) in preparing this manuscript. The use of LLMs was limited to writing refinement and language polishing, such as correcting grammar, improving wording, clarity, and readability. LLMs were not used for scientific ideation, hypothesis formulation, methodology or experimental design, method implementation, data processing, result analysis or interpretation, or drawing scientific conclusions. All scientific contributions are entirely the work of the authors. We have reviewed all LLM-assisted content and take full responsibility for the final manuscript.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Kai-Wei Chang, Akshay Krishnamurthy, Akshay Agarwal, Hal Daumé, III, and John Langford. Learning to search better than your teacher. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2058–2066. PMLR, 07–09 Jul 2015.

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

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 16716–16728. Curran Associates, Inc., 2023.

Gongfan Fang, Kunjun Li, Xinyin Ma, and Xinchao Wang. Tinyfusion: Diffusion transformers learned shallow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18144–18154, June 2025.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, et al. Flow-opd: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Hyunsoo Han, Sangyeop Yeo, and Jaejun Yoo. Lift and place: A simple, stable, and effective knowledge distillation framework for lightweight diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5564–5573, June 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, et al. D-opsd: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026.

Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. Bk-sdm: A lightweight, fast, and cheap version of stable diffusion. In *European Conference on Computer Vision*, pp. 381–399. Springer, 2024.

Dohyun Kim, Sehwan Park, Geonhee Han, Seung Wook Kim, and Paul Hongsuck Seo. Random conditioning for diffusion model compression with distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18607–18618, June 2025.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Youngwan Lee, Kwanyong Park, Yoorhim Cho, Yong-Ju Lee, and Sung Ju Hwang. Koala: Empirical lessons toward memory-efficient and fast diffusion models for text-to-image synthesis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 51597–51633. Curran Associates, Inc., 2024.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. Diffusionopd: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, 2020.

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 *International Conference on Learning Representations*, 2023.

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17719–17728, October 2025.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, 2025.

Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26245–26265. PMLR, 23–29 Jul 2023.

Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *International Conference on Learning Representations*, volume 2024, pp. 15167–15189, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

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

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 23–29 Jul 2023.

Qianlong Xiang, Miao Zhang, Yuzhang Shang, Jianlong Wu, Yan Yan, and Liqiang Nie. Dkdm: Data-free knowledge distillation for diffusion models with any architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2955–2965, June 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22552–22562. IEEE, 2023.

Caleb Zheng and Eli Shlizerman. 2ndmatch: Finetuning pruned diffusion models via second-order jacobian matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43385–43395, June 2026.

Wei Zhou, Xiongwei Zhu, Zelin Xu, Bo Dong, Lixue Gong, Yongyuan Liang, Meng Chu, Leigang Qu, Lingdong Kong, Wei Liu, et al. Danceopd: On-policy generative field distillation. *arXiv preprint arXiv:2606.27377*, 2026.

Zhenyu Zhou, Defang Chen, Can Wang, Chun Chen, and Siwei Lyu. Simple and fast distillation of diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 40831–40860. Curran Associates, Inc., 2024.

A Mathematical Interpretation

A.1 Trajectory-KL Decomposition

For a stochastic Markov sampler with a shared initial distribution and common support, model M induces a trajectory $\tau = (x_0, \ldots, x_N)$ with distribution

$$P_M(\tau \mid c) = p_{\text{init}}(x_0) \prod_{i=0}^{N-1} p_M(x_{i+1} \mid x_i, c). \tag{11}$$

For stochastic Markov samplers $A, B \in \{\mathcal{T}, \mathcal{S}\}$ in Eq. equation 11, fix the condition c and assume a shared initial distribution and common support. The initial factor cancels from the likelihood ratio, and conditioning on the state at each position gives

$$\begin{aligned} D_{\text{KL}}(P_A \| P_B) &= \mathbb{E}_{\tau \sim P_A}\left[\sum_{i=0}^{N-1} \log \frac{p_A(x_{i+1} \mid x_i, c)}{p_B(x_{i+1} \mid x_i, c)}\right] \\ &= \sum_{i=0}^{N-1} \mathbb{E}_{x_i \sim q_i^A}\left[D_{\text{KL}}(p_A(\cdot \mid x_i, c) \| p_B(\cdot \mid x_i, c))\right]. \end{aligned} \tag{12}$$

$$p_M(x_{t-1} \mid x_t, c) = \mathcal{N}\big(x_{t-1}; \mu_M(x_t, t, c), \sigma_t^2 I\big), \qquad M \in \{\mathcal{T}, \mathcal{S}\}. \tag{13}$$

At nondegenerate stochastic steps ($\sigma_t > 0$), Teacher and Student share covariance $\sigma_t^2 I$. Defining $\kappa_t(x_t) = D_{\text{KL}}(p_{\mathcal{T}}(\cdot \mid x_t, c) \| p_{\mathcal{S}}(\cdot \mid x_t, c))$, the shared-covariance Gaussian KL becomes

$$\kappa_t(x_t) = \frac{1}{2\sigma_t^2} \left\|\mu_{\mathcal{T}}(x_t, t, c) - \mu_{\mathcal{S}}(x_t, t, c)\right\|_2^2 \propto \left\|\epsilon_{\mathcal{T}}(x_t, t, c) - \epsilon_{\mathcal{S}}(x_t, t, c)\right\|_2^2. \tag{14}$$

The same relation holds for velocity parameterization after replacing ϵ_M with v_M. The proportionality factor depends on the schedule, variance, and parameterization.

A stochastic transition distribution with shared scale $\rho > 0$ takes the form

$$p_M(x_{t'} \mid x_t, c) = \mathcal{N}\big(x_{t'}; x_t + (t' - t) v_M(x_t, t, c), \rho^2 I\big), \tag{15}$$

which gives $\kappa_t(x_t) \propto \|v_{\mathcal{T}}(x_t, t, c) - v_{\mathcal{S}}(x_t, t, c)\|_2^2$. Deterministic flow ODEs have degenerate one-step conditional distributions and directly use velocity or state-update regression (Fang et al., 2026).

Reindexing the one-step conditional KL as κ_i, shared covariances make its Teacher-to-Student and Student-to-Teacher forms equal. Applying the chain rule for KL divergence to Eq. (11) gives the following decomposition:

$$D_{\text{KL}}(P_{\mathcal{T}} \| P_{\mathcal{S}}) = \sum_i \mathbb{E}_{q_i^{\mathcal{T}}}[\kappa_i(x_i)], \qquad D_{\text{KL}}(P_{\mathcal{S}} \| P_{\mathcal{T}}) = \sum_i \mathbb{E}_{q_i^{\mathcal{S}}}[\kappa_i(x_i)]. \tag{16}$$

Here and below, the sum ranges over $i = 0, \ldots, N-1$. Choosing $(A, B) = (\mathcal{T}, \mathcal{S})$ or $(\mathcal{S}, \mathcal{T})$ yields Eq. equation 16: the difference lies in the visitation distribution induced by the preceding trajectory. This decomposition explains the state sources of TPD and OPD; FD-KD does not optimize a trajectory KL. Deterministic flow rollouts have degenerate transition distributions, so their state-source comparison uses $q_i^{\mathcal{T}}$ and $q_i^{\mathcal{S}}$ with local velocity matching without invoking a finite conditional KL.

A.2 First-Order Interpretation of Future Discrepancy

Fix the common state, condition, current split, and coupled continuation randomness. Write $z_i^{\mathcal{S}} = z_i^{\mathcal{T}} + d_i$ as in Eq. equation 3. The exact FD vector is the finite difference in Eq. equation 6. At the Teacher split output, define the continuation input Jacobian

$$J_{i \to j} = \nabla_z \mathcal{R}_{i \to j}^{\mathcal{T}}(z)\big|_{z = z_i^{\mathcal{T}}}. \tag{17}$$

If this Jacobian is locally Lipschitz along the segment between the split outputs, Taylor expansion gives

$$G_{i \to j} = J_{i \to j} d_i + r_{i,j}, \qquad \|r_{i,j}\|_2 \le \tfrac{L}{2} \|d_i\|_2^2, \tag{18}$$

where L is a Lipschitz constant on that segment. For $j > i$, let $A^{\mathcal{T}}_{i,k}$ be the input Jacobian of Teacher update $\mathcal{F}^{\mathcal{T}}_k$ evaluated at $z^{\mathcal{T}}_{i\to k-1}$. The chain rule gives

$$J_{i\to j} = A^{\mathcal{T}}_{i,j} \cdots A^{\mathcal{T}}_{i,i+1} \quad (j > i), \qquad J_{i\to i} = I. \tag{19}$$

Thus the local direction, not only its magnitude, affects the future discrepancy. FD training evaluates the finite nonlinear difference, not this Jacobian approximation. At $j = i$, continuation is the identity and $G_{i\to i} = d_i$ exactly. The resulting state-space loss corresponds to the immediate state-update difference; equating it with prediction-space local KD also requires aligned update scaling, position weights, and reductions.

B Experimental and Implementation Details

B.1 Architectures and Initialization

The ImageNet-1K analyses use SiT/2 models. Table 8 lists the Teachers and Students in the two Student construction settings; SiT-B serves as Teacher in both.

Table 8: SiT/2 architectures used in the ImageNet-1K analyses. Panel (a) uses Students pruned from SiT-B; panel (b) uses an independently pretrained SiT-S Student.

Model	Role	Transformer blocks	Parameters
(a) Depth-pruned Students			
SiT-B	Teacher	12	130M
SiT-B-D8	Student	8	88M
SiT-B-D6	Student	6	67M
SiT-B-D4	Student	4	45M
(b) Independently pretrained Student			
SiT-XL	Teacher	28	675M
SiT-L	Teacher	24	458M
SiT-B	Teacher	12	130M
SiT-S	Student	12	33M

Depth-pruned Students. We use the BK-SDM-style oracle index selection described in TinyFusion (Kim et al., 2024; Fang et al., 2025) to initialize compact Students from a pretrained SiT-B Teacher. For SiT-B-D6, number the Teacher's 12 Transformer blocks from 0 to 11 and group adjacent blocks into six pairs. We retain the first block from each of the first three pairs and the second block from each of the last three pairs, giving indices $[0, 2, 4, 7, 9, 11]$. The fixed retained indices for SiT-B-D8 and SiT-B-D4 are $[0, 1, 3, 4, 7, 8, 10, 11]$ and $[0, 3, 8, 11]$, respectively. Each Student copies these blocks in their original order from the SiT-B EMA checkpoint at 400K updates; all unpruned embedding, conditioning, and output weights are copied from the same checkpoint.

Independently pretrained Student. Following the separately pretrained Teacher–Student setting in GKD (Agarwal et al., 2024), SiT-S is trained on ImageNet-1K independently of the Teachers before distillation. SiT-S, SiT-B, and SiT-L use the flow-matching training objective of SiT (Ma et al., 2024) and are pretrained for 80 epochs (400K updates) each. Pretraining uses AdamW with learning rate 10^{-4}, zero weight decay, EMA decay 0.9999, and bfloat16 precision. SiT-XL instead uses the official SiT release trained for 7M updates (Ma et al., 2024).

B.2 Training Details

Distillation uses a constant learning rate of 10^{-4} and gradient clipping. The SiT runs use AdamW, EMA with decay 0.9999, and bfloat16 precision. We run the experiments in PyTorch on two RTX 4090 GPUs. The complete SiT-B $\to$ SiT-B-D6 FD-KD recipe uses $\lambda_{\mathrm{KD}} = 1$.

At a fixed horizon, Dense covers every valid position, Global Random uniformly selects a finite subset from the common support, and Stratified partitions the candidate positions into contiguous temporal regions and samples within each region. Dense in Table 4 repeats the $h = 1$ setting of

Table 3. Subsequent FD experiments use $h = 1$ and three stratified positions per trajectory unless otherwise specified.

Each update draws temporal pairs (i, j) from a fresh rollout, recomputes the Teacher–Student split at detached states, and aggregates the resulting FD losses after frozen-Teacher continuation. Classic KD is computed in parallel on independently noised data states. Rollout states are detached; frozen Teacher continuation retains input gradients on the Student branch as specified in Section 3.5. Inference uses only the compact Student.

C Mechanism Analysis Details

This section gives the measurements behind Figures 3 and 4. All diagnostics use fixed checkpoints; trained Student checkpoints use 25K updates, and model evaluations use CFG 1.5.

C.1 Sampling-History State Gap

Figure 3(a) uses 32 paired noise–class inputs, 50-step Euler sampling for SiT-B $\rightarrow$ SiT-B-D6. The Teacher and each recursive Student rollout start from the same noise. At position q, the plotted gap for checkpoint m is the root-mean-square latent-state gap, $\left[(32d_z)^{-1}\sum_{n=1}^{32}\|z^m_{q,n} - z^{\mathcal{T}}_{q,n}\|_2^2\right]^{1/2}$, where d_z is the number of latent coordinates. For the Initial Student one-step curve, we instead reset to each preceding Teacher state, take one Student state update, and compare its next state with the paired Teacher next state. This isolates the current-step mismatch from accumulated sampling history. Shading gives paired-input bootstrap 95% intervals from 5,000 resamples. The final near-zero scheduler update is retained in the records but omitted from the plotted support. The classic-KD, TPD, and OPD checkpoints each use 25K training updates. The curves characterize sampling-history deviations.

C.2 Paired Local-to-Future Comparison

Figure 3(b,c) evaluates SiT-B $\rightarrow$ SiT-B-D6 TPD and OPD checkpoints trained for 25K updates on the same pre-update Teacher states $x^{\mathcal{T}}_q$. For checkpoint m, with time and condition implicit, the local prediction error is $P^m_q = \|f_m(x^{\mathcal{T}}_q) - f_{\mathcal{T}}(x^{\mathcal{T}}_q)\|_2^2$. The future error $H^m_{q,h} = \|G^m_{q\rightarrow q+h}\|_2^2$ uses the same input $x^{\mathcal{T}}_q$ and checkpoint m for the split, followed by shared frozen-Teacher continuation. Both squared norms sum over latent coordinates. The plotted ratios are $x = P^{\mathrm{TPD}}_q / P^{\mathrm{OPD}}_q$ and $y_h = H^{\mathrm{TPD}}_{q,h} / H^{\mathrm{OPD}}_{q,h}$; the color grouping in panel (b) uses $g_h = y_h/x$.

We use 128 paired noise–class trajectories, query positions $q \in \{1, 4\}$, and 10-step Euler sampling. Panel (b) shows all 256 trajectory–position pairs at $h = 3$. Panel (c) selects the 63 pairs with $\max(x, 1/x) \leq 1.05$ and follows these same pairs through $h = 1, \ldots, 5$. Its markers show the median, 25th–75th percentiles, and 10th–90th percentiles at each horizon.

C.3 Temporal Signal Coverage

Figure 4 uses 256 Student trajectories from one fixed FD SiT-B $\rightarrow$ SiT-B-D6 checkpoint, $h = 1$, and all nine candidate positions $q = 0, \ldots, 8$. For trajectory n, let K_n be the uncentered Gram matrix of its nine flattened FD vectors, with entries $K_n[a, b] = (G^{(n)}_{a\rightarrow a+1})^\top G^{(n)}_{b\rightarrow b+1}$. Panel (a) computes cosine similarity from the pooled Gram $\sum_n K_n$. For a selected position subset S, panel (b) measures its linear-span coverage of the dense FD vectors by

$$\mathrm{Ret}_n(S) = \frac{\mathrm{tr}\big(K_n[:, S]K_n[S, S]^\dagger K_n[S, :]\big)}{\mathrm{tr}(K_n)}. \tag{20}$$

Here $\dagger$ denotes the Moore–Penrose pseudoinverse. At budget R, Global Random averages all R-position subsets; Stratified averages subsets with one position from each of R contiguous groups. Each curve point averages these exact within-trajectory expectations equally across trajectories; shading shows their 25th–75th percentiles, not a confidence interval. This is an offline span-coverage measure. Generation quality and cost in Table 4 come from separately trained policies.

D Additional Ablations

We examine run-to-run variation, the classic KD weight, the number of Student rollouts, and the training rollout length on ImageNet-1K with SiT-B $\rightarrow$ SiT-B-D6. All FID values use 50K generated samples. Each sweep changes its named factor while holding its other training and evaluation settings fixed.

D.1 Variation Across Independent Runs

Table 9 collects the SiT-B $\rightarrow$ SiT-B-D6 means reported in the main text. The dense trajectory-supervision rows correspond to Table 1; the complete FD-KD rows use sparse temporal sampling as in Table 5. Available standard deviations are retained; pending entries are marked TBD.

Table 9: Independent-run reporting for SiT-B $\rightarrow$ SiT-B-D6 on ImageNet-1K. Entries are mean $\pm$ sample standard deviation over three training runs, using the same 50K evaluation samples. TBD marks pending standard deviations.

Method	FID↓	IS↑	Precision↑	Recall↑
Classic KD	24.29 $\pm$ 0.094	66.14 $\pm$ 0.138	0.61 $\pm$ 0.001	0.52 $\pm$ 0.003
Dense trajectory supervision (Table 1)				
TPD	16.09 $\pm$0.041	109.49 $\pm$0.124	0.72 $\pm$0.001	0.36 $\pm$0.002
FD($\mathcal{T}$)	15.07 $\pm$0.036	112.94 $\pm$0.113	0.74 $\pm$0.001	0.35 $\pm$0.002
OPD	15.64 $\pm$0.037	119.07 $\pm$0.113	0.74 $\pm$0.003	0.31 $\pm$0.002
FD($\mathcal{S}$)	14.29 $\pm$0.040	125.57 $\pm$0.130	0.76 $\pm$0.002	0.30 $\pm$0.002
Complete FD-KD with sparse sampling (Table 5)				
FD-KD($\mathcal{T}$)	14.98 $\pm$0.036	111.67 $\pm$0.109	0.72 $\pm$0.002	0.39 $\pm$0.002
FD-KD($\mathcal{S}$)	14.35 $\pm$ 0.036	116.66 $\pm$ 0.113	0.73 $\pm$ 0.003	0.37 $\pm$ 0.001

Table 10: Classic KD weight for SiT-B $\rightarrow$ SiT-B-D6.

λ_{KD}	Recipe	FID↓	IS↑
0	FD($\mathcal{S}$)	14.89	124.92
0.5	FD-KD($\mathcal{S}$)	14.70	115.00
1	FD-KD($\mathcal{S}$)	14.35	116.65
2	FD-KD($\mathcal{S}$)	14.21	116.57

Table 11: Student rollout trajectories per update.

K	FID↓	IS↑	Cost↓
32	16.65	113.97	0.45
64	14.89	124.92	0.72
128	13.91	131.55	1.40

Table 12: Euler steps per training rollout.

N	FID↓	IS↑	Cost↓
5	25.80	80.75	0.62
10	14.89	124.92	0.72
20	14.09	131.56	0.92

D.2 Weight of the Classic KD Branch

Table 10 varies λ_{KD} in Eq. (10), holding the FD term, loss reductions, rollout sampling, and update budget fixed. At zero weight, the objective is FD($\mathcal{S}$). Across the tested weights, FID decreases from 14.89 to 14.21, while IS at each positive weight is below the zero-weight value of 124.92. The weight also changes the total gradient scale, so this sweep characterizes sensitivity of the training recipe rather than isolating a single mechanism.

D.3 Number of Rollout Trajectories and Steps

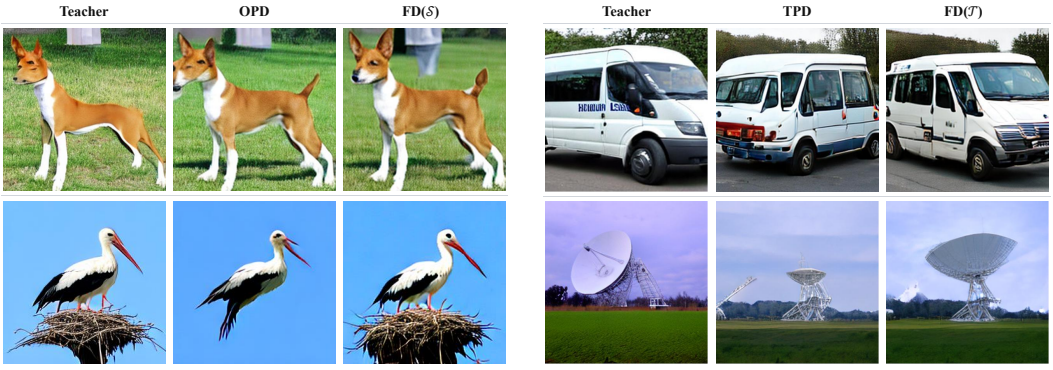
Table 11 varies the number K of fresh Student rollout trajectories per update, with the training budget, rollout steps, and temporal queries per trajectory fixed. Raising K from 32 to 128 reduces FID from 16.65 to 13.91 and raises IS from 113.97 to 131.55, while iteration time rises from 0.45 to 1.40 seconds.

Table 12 varies the training rollout length N while fixing K, the number of temporal queries, training updates, and the evaluation sampler. Candidate positions cover comparable relative times on each grid, with $h = 1$ on every grid. From 5 to 20 Euler steps, FID falls from 25.80 to 14.09 and iteration time rises from 0.62 to 0.92 seconds. Because the physical interval of one continuation step changes with N, this is a sensitivity test of the training discretization.

E Additional Visualizations

This section presents class-conditional ImageNet-1K and text-conditional MS-COCO samples for the configurations discussed in the main text.

Class-conditional ImageNet-1K. Figure 7 shows two configurations with opposite local-source preferences: OPD versus FD($\mathcal{S}$) for SiT-B $\rightarrow$ SiT-B-D6, and TPD versus FD($\mathcal{T}$) for SiT-XL $\rightarrow$ SiT-S. Within each panel, both methods use the same trajectory-state source.



(a) SiT-B $\rightarrow$ SiT-B-D6 (b) SiT-XL $\rightarrow$ SiT-S

Figure 7: Class-conditional ImageNet-1K samples for two Teacher–Student configurations. (a) shows the Teacher, OPD, and FD($\mathcal{S}$); (b) shows the Teacher, TPD, and FD($\mathcal{T}$).

Text-conditional MS-COCO. Figure 8 compares BK-SDM with FD-KD($\mathcal{T}$) under the same prompts and sampling settings for Base and Tiny Students.

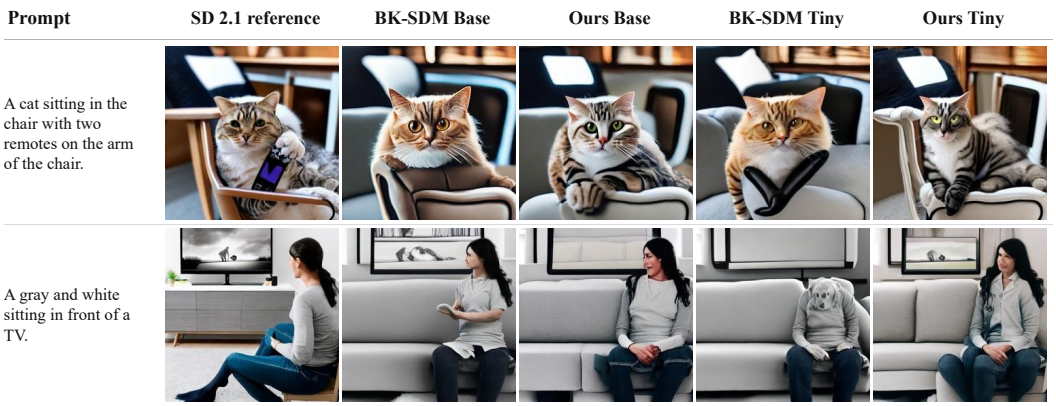


Figure 8: Generated samples for two MS-COCO prompts, one per row. Columns show SD 2.1, BK-SDM Base, FD-KD($\mathcal{T}$) Base, BK-SDM Tiny, and FD-KD($\mathcal{T}$) Tiny. All columns use CFG 1.5 and 25 denoising steps.