DEFECT-GUIDED TIME-WARP FLOW DISTILLATION FOR SEMIGROUP-ROBUST FAST DIFFUSION GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fast diffusion generation has been improved through few-step distillation, and trajectory rectification. However, a central difficulty remains: a student model only trained on single-step transitions is deployed via repeated composition across different step counts. This resulting discrepancy between a direct transition and its multi-step composition is defined as the *semigroup defect*. We therefore recast fast diffusion distillation as a *time-geometry adaptation* problem, where temporal allocation along the trajectory is optimized instead of explicitly modifying the state-space path. Specifically, we propose **Defect-Guided Time-Warp Flow Distillation** (DG-TWFD), which jointly learns a continuous-time student flow map and a monotone time warp that reallocates temporal resolution toward teacher regions with high compositional difficulty. Training integrates teacher flow matching with defect-aware optimization and a time-warped curriculum. We evaluate our method on unconditional CIFAR-10 and ImageNet-64, achieving competitive Fréchet Inception Distance (FID) and Inception Score (IS) on CIFAR-10, and FID and recall on ImageNet-64 across different inference step settings. Our best results are 1.81 FID / 9.87 IS on CIFAR-10 and 2.49 FID / 0.67 recall on ImageNet-64. All code and data used in this work are available at Anonymous Github link.

1 INTRODUCTION

Diffusion and score-based generative models generate samples by evolving an initial noisy state along a time-indexed trajectory toward the data distribution (Ho et al., 2020; Song et al., 2021b). Existing acceleration methods treat the time axis as fixed and focus on reshaping the trajectory in state space, either by improving numerical accuracy along a fixed trajectory (Song et al., 2021a; Karras et al., 2022), or by rectifying its geometric shape toward straighter dynamics (Liu et al., 2023; Lee et al., 2024). Such methods improve generation quality by refining intermediate trajectory behavior, yet leave the temporal structure of the process itself unexamined.

In diffusion distillation, a student model is trained on local transition targets but deployed via recursive composition (Yin et al., 2024). In this setting, state-space mitigation strategies become less effective, as intermediate trajectory information is not explicitly supervised due to discrepancies in time-indexed alignment between teacher and student sampling trajectories (Zhou et al., 2024). In principle, a student may match teacher endpoints well on isolated pairs while still exhibiting large disagreement between a direct transition and its composed counterpart. We refer to this discrepancy as the *semigroup defect*.

To solve the core difficulty, consistency training objectives (Song et al., 2023) address the semigroup defect by enforcing cross-time consistency along the diffusion trajectory, but step-count brittleness under recursive composition persists. Solver-based improvements reduce numerical errors along a fixed teacher trajectory but do not alter this compositional burden (Song et al., 2021a; Karras et al., 2022). Local distillation objectives improve endpoint matching without guaranteeing consistency between direct and multi-step compositions (Salimans & Ho, 2022; Song et al., 2023; Kim et al., 2024). Explicit straightness regularization (Wang et al., 2025) remains a purely state-space intervention that may conflict with the inherently curved teacher dynamics. Recent work further shows that the benefits of rectification do not require geometric straightness in state space (Wang et al., 2025), and that generative modeling can be formulated as learning direct mappings between arbitrary time-

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.

indexed states (Sabour et al., 2025), suggesting that the semigroup defect can be viewed as a temporal alignment problem rather than a state-space one.

This motivates viewing diffusion distillation as a **compositional time-geometry problem**, where the time scale is redistributed to mitigate compositional inconsistency. Time reparameterization via adaptive scheduling has been shown to redefine effective time discretization with meaningful impact on generation (Stancevic et al., 2025), and geometric interpretations in physics further underscore the effectiveness of such temporal coordinate freedom (Saniga, 2003). We therefore propose **Defect-Guided Time-Warp Flow Distillation** (DG-TWFD) to learn a time-warp parameterization $g_\phi(t)$ that adaptively reshapes the temporal axis based on variational semigroup defect, together with a continuous-time student flow map $M_\theta(t, s, x_t)$. Under this warped coordinate, local teacher evolution becomes more nearly first-order, making the student easier to compose from 1 to few-step inference.

Our main contributions are: first, we reformulate fast diffusion distillation as a *time-geometry adaptation* problem and propose DG-TWFD, which couples continuous-time flow-map learning with a learnable monotone time warp. Second, we make *semigroup defect* a structural training target, enabling one student to better generalize across multiple step counts. Third, we propose a practical and robust framework leveraging cached teacher trajectories, warped-time control, a defect-adaptive curriculum, and an optional boundary stabilizer to operationalize our time-warping idea for fast diffusion generation.

2 METHODOLOGY: DEFECT-GUIDED TIME-WARP FLOW DISTILLATION

2.1 DEFINITION OF SEMIGROUP DEFECTS

Diffusion and score-based models formulate sampling as reverse-time evolution of a noise-perturbed distribution (Ho et al., 2020; Song et al., 2021b), which in continuous time is described by the probability flow ODE (PF-ODE) (Song et al., 2021b). Let $t \in [0, 1]$ denote the noise level, with $t = 1$ and $t = 0$ corresponding to the noisy and clean endpoints, respectively. The teacher flow map is defined as $\Phi_T(t \to r, x_t) = x_r, \quad 0 \le r < t \le 1$. It satisfies the semigroup property $\Phi_T(t \to r, x_t) = \Phi_T\big(s \to r, \Phi_T(t \to s, x_t)\big), \quad 0 \le r < s < t \le 1$. where t, s, r are three time points. Deviations from this property give rise to the *semigroup defect*, the discrepancy between direct and composed transitions. Specifically, we define $M_\theta(t_i, t_j, x_{t_i})$ as a student transition map to x_s, $M_\theta(t_i, t_j, x_{t_i}) \mapsto \hat{x}_{t_j}, 0 \le t_j < t_i \le 1$, where t_i and t_j are time points and x_{t_i} is the states of diffusion at the corresponding time. $\hat{x}_{t_j}$ approximates the teacher target $x_{t_j} = \Phi_T(t_i \to t_j, x_{t_i})$. Its compositional error, defined as *semigroup defect*, at three time point t, s, r is measured with the quadratic norm by

$$\Delta_\theta(t, s, r; x_t) = \left\| M_\theta(t, r, x_t) - M_\theta\left(s, r, M_\theta(t, s, x_t)\right) \right\|_2 . \tag{1}$$

A k-step sampler applies the student along a grid $t = t_0 > t_1 > \cdots > t_k = r$, so different step budgets are different partitions of the same interval. If Δ_θ vanished for all triplets, every partition would return the one-step output $M_\theta(t, r, x_t)$; a large Δ_θ means the output changes with the step count. We aim to keep Δ_θ small over the diversity of time range, so the same student could be stably reused across different step counts.

2.2 THE TIME-WARPED FLOW DISTILLATION FORMULATION

We propose **Defect-Guided Time-Warp Flow Distillation** (DG-TWFD), a coupled framework that learns a continuous-time student flow map together with a monotone time reparameterization. The student map is defined aforementioned.

To address the time-geometry issue discussed above, we propose learning a monotone time warp, rather than relying on predefined time distortion schemes:

$$g_\phi : [0, 1] \to [0, 1], \quad g_\phi(0) = 0, \quad g_\phi(1) = 1, \quad g'_\phi(t) > 0. \tag{2}$$

We denote the warped time coordinate by

$$u = g_\phi(t), \qquad t = g_\phi^{-1}(u). \tag{3}$$

The purpose of g_ϕ is not to alter the teacher state trajectory in ambient space, but to redistribute temporal resolution such that difficult regions occupy more mass in warped time. Given g_ϕ, the

teacher induces a warped-time flow map $\widetilde{\Phi}_T(u \to v, x_u) = \Phi_T\left(g_\phi^{-1}(u) \to g_\phi^{-1}(v), x_u\right)$. Similarly, any term with "$\sim$" denotes the corresponding process under warped time, and the student can be interpreted in warped coordinates as

$$\widetilde{M}_\theta(u, v, x_u) = M_\theta\left(g_\phi^{-1}(u), g_\phi^{-1}(v), x_u\right). \tag{4}$$

The key structural quantity in our method is the *time-warped semigroup defect*

$$\widetilde{\Delta}_\theta(u, v, w; x_u) = \left\| \widetilde{M}_\theta(u, w, x_u) - \widetilde{M}_\theta\left(v, w, \widetilde{M}_\theta(u, v, x_u)\right) \right\|_2, \text{where } w < v < u. \tag{5}$$

This defect measures whether multi-step application of the student is consistent with a direct transition. Our objective is not merely to fit teacher transitions, but to jointly learn (M_θ, g_ϕ) to minimize the semigroup defect such that the resulting transitions remain consistent under multi-step composition.

Unlike pre-defined time parameterizations, the learnable warp g_ϕ acts as a temporal density allocator: it expands regions of rapid trajectory change in u-space and compresses simpler regions, naturally adapting to varying trajectory complexity while retaining a uniform sampling rule in warped time. Crucially, as proved in Appendix A, the warp preserves the geometric image of the teacher path and only reparameterizes *how fast* it is traversed. This is fundamentally different from explicit straightness penalties, which attempt to flatten the trajectory in ambient space and may conflict with teacher dynamics when curvature carries semantic information. By contrast, DG-TWFD keeps the student teacher-aligned in state space, reducing temporal stiffness and improving compositional regularity without modifying the teacher path itself.

2.3 Algorithm and optimization objective

As aforementioned, our goal is to distill a frozen teacher into a single student flow map that can be reused under different inference step counts. The central difficulty is not only endpoint approximation, but also recursive composition: once the student is applied repeatedly, direct and composed transitions should remain consistent. DG-TWFD is therefore designed to couple endpoint supervision with normalized semigroup defect and converts the resulting defect statistics into an adaptive time warp. The effect is to allocate more training and inference resolution to the intervals that are empirically hardest to compose. The framework is shown in Figure 1, and we detail each components as follows.

Teacher bank and student flow map. The training data for students need to come from teacher trajectories. For training efficiency, we treat the teacher flow map as frozen and all student supervision generated offline. For each sample n, we store a sparse teacher trajectory

$$\mathcal{T}^{(n)} = \left\{(t_i, x_{t_i}^{(n)})\right\}_{i=0}^{m}, \qquad 1 = t_0 > t_1 > \cdots > t_m = 0, \tag{6}$$

with x representing the status of the flow, and collect all trajectories into a bank $\mathcal{B}$. For an arbitrary time $a \in [0, 1]$, we obtain the corresponding state by interpolation: $x_a^{(n)} = \mathcal{I}\left(\mathcal{T}^{(n)}, a\right)$. This offline bank removes expensive teacher simulation from the student loop and ensures that the same trajectory can provide both pairwise endpoint targets and triplet-based composition targets. A lightweight online teacher query is kept only as a fallback for cache misses. Implementation details of cache construction and interpolation are given in Appendix B.1. For an ordered triplet $t > s > r$, the student is evaluated along both a direct transition $\hat{x}_r^{\mathrm{dir}} = \widetilde{M}_\theta(t, r, x_t)$ (one large jump from t to r with warped time) and a composed transition $\hat{x}_r^{\mathrm{cmp}} = \widetilde{M}_\theta(s, r, \hat{x}_s)$ (the same transition decomposed as $t \to s \to r$ with warped time).

Student flow map match objective. Then, we train the student with the teacher bank we retrieve and get $x_t = \mathcal{I}(\mathcal{T}, t)$, $x_r = \mathcal{I}(\mathcal{T}, r)$ and $x_s = \mathcal{I}(\mathcal{T}, s)$. Endpoint supervision anchors the student to the teacher trajectory, so we match the transition with the teacher:

$$\mathcal{L}_{\mathrm{match}} = \mathbb{E}\left[\left\|\widetilde{M}_\theta(t, s, x_t) - x_s\right\|_2^2 + \left\|\widetilde{M}_\theta(t, r, x_t) - x_r\right\|_2^2\right]. \tag{7}$$

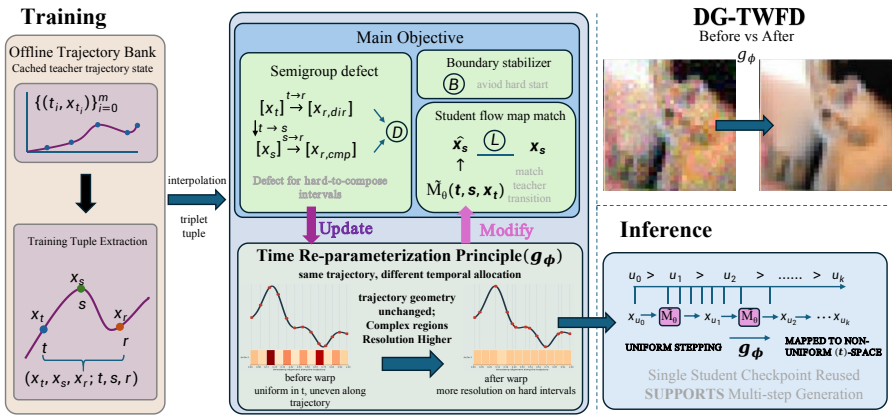


Figure 1: **Overview of DG-TWFD.** During *training*, DG-TWFD learns a time warp that re-parameterizes time according to composition difficulty. First, a cached teacher trajectory bank is constructed to supply offline data for supervising a continuous-time flow-map student ($\widetilde{M}_\theta$) via semigroup defect loss and student flow-map matching loss. The semigroup defect is computed by comparing the direct transition $(t \to r)$ against the composed transition $(t \to s \to r)$, while the matching loss aligns the student flow map with the corresponding teacher transition. Optionally, a boundary stabilizer can be incorporated to improve the first jump near the high-noise endpoint. The accumulated defect field then drives updates to the time re-parameterization g_ϕ, which expands high-complexity intervals in warped time and samples them more frequently, which concludes one training round. In each subsequent round, the updated re-parameterization is applied to recompute the training losses. During *inference*, uniformly spaced steps in warped time induce a non-uniform schedule in the original time domain, enabling robust multi-step generation. A representative sample is shown in the top-right corner.

Normalized semigroup defect. Teacher endpoint matching alone does not guarantee that the student can be reused recursively. For optimization and time allocation, we therefore use a normalized semigroup-defect score

$$\bar{\Delta}_\theta(t,s,r;x_t) = \frac{\|\widetilde{M}_\theta(t,r,x_t) - \widetilde{M}_\theta(s,r,\widetilde{M}_\theta(t,s,x_t))\|^2_{\text{mse}}}{\epsilon_\Delta + \|x_r - x_t\|^2_{\text{mse}}}, \qquad \|z\|^2_{\text{mse}} := \frac{1}{d}\|z\|_2^2. \tag{8}$$

The corresponding structural regularizer is $\mathcal{L}_{\text{def}} = \mathbb{E}\left[\bar{\Delta}_\theta(t,s,r;x_t)\right]$. However, $\mathcal{L}_{\text{def}}$ cannot be used alone, since direct and composed predictions may agree while both deviate from the teacher. We therefore optimize

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{match}} + \lambda_{\text{def}}\mathcal{L}_{\text{def}} + \mathbf{1}_{\text{bdry}}\lambda_{\text{bdry}}\mathcal{L}_{\text{bdry}}, \tag{9}$$

where the boundary term bdry is used for improving the first jump near the high-noise endpoint where the teacher dynamics are often the most fragile for low-step generation, and is optional. Its exact form and activation rule are deferred to Appendix B.2, without compromising the core design of the time reparameterization. λ is the hyper-parameter to adjust the importance of each term.

Defect-guided time re-parameterization principle. After student loss calculation, we need to adapt the training and inference time geometry as aforementioned. **Normalized semigroup defect has been used for supervising time re-parameterization, and details are discussed below.** We partition the diffusion clock into bins

$$\mathcal{P} = \{I_b\}_{b=1}^B, \qquad I_b = [\xi_{b-1}, \xi_b), \qquad 0 = \xi_0 < \xi_1 < \cdots < \xi_B = 1, \tag{10}$$

where $\ell_b = \xi_b - \xi_{b-1}$ be the bin length and the number of bin is a hyperparameter we decide. ℓ_b shows the reparameterization of time, where the originally uniform segments are adjusted based on the complexity of transitions.

Specifically, given a triplet of time indices t, r, s, where r is randomly sampled, we identify the corresponding transition sample by determining within which ℓ_b interval the value of r falls. With

such strategy, we allocate all samples with their corresponding bin. For each bin, we maintain two running profiles on this same partition: $\widehat{d}_b$, the Exponential Moving Average (EMA) (Hunter, 1986) of the normalized semigroup defect, and $\widehat{v}_b$, the EMA of a teacher-side local variation statistic. **We use semigroup defect to supervise** $\hat{d}_b$ and $\hat{v}_b$ is computed from cached teacher triplets by comparing adjacent warped-time teacher increments,

$$\delta_{32} = x_{t_2} - x_{t_3}, \qquad \delta_{21} = x_{t_1} - x_{t_2}, \qquad V(t_3, t_2, t_1) = \|\delta_{32} - \delta_{21}\|_{\text{mse}}. \tag{11}$$

Unlike the student defect, V serves as a teacher-side prior that prevents neglecting intervals with rapid teacher dynamics, encouraging the student to stay close to the teacher flow, as in Rectified Diffusion (Wang et al., 2025). The EMA updates and normalization used to estimate $\widehat{d}_b$ and $\widehat{v}_b$ are described in Appendix B.3. After normalizing both profiles across bins, we form $\widetilde{d}_b$ and $\widetilde{v}_b$ to define

$$\pi_b = k \left(\epsilon_w + \widetilde{d}_b + \lambda_{\text{var}} \widetilde{v}_b\right)^{\gamma}, \qquad \gamma \geq 0, \tag{12}$$

where k represents other trivial factors such as normalization strategy. The aforementioned procedure constitutes a single update cycle for both the time reparameterization and the backbone architecture, guided by the student loss. The training process then iteratively proceeds by recomputing the student matching objective.

Monotone warp and inference. The same bin probabilities define the warp. At the bin boundaries we set $g(\xi_0) = 0, g(\xi_b) = \sum_{j=1}^{b} \pi_j, b \in [1, B]$. Inside each bin I_b, we interpolate linearly:

$$g(t) = g(\xi_{b-1}) + \frac{t - \xi_{b-1}}{\ell_b} \pi_b, \qquad t \in I_b. \tag{13}$$

g is strictly increasing and satisfies $g(0) = 0$ and $g(1) = 1$. The identity $g(\xi_b) - g(\xi_{b-1}) = \pi_b$ shows that the warped-time length assigned to a bin is exactly the same quantity that determines how often it is sampled during training. The key design is that the same adaptive mass π_b is used twice: it is the probability of sampling training triplets from I_b, and it is also the warped-time length assigned to I_b. This ties the training curriculum to the inference schedule. The concrete triplet sampler induced by this warp is provided in Appendix B.4.

At test time, the final adaptive warp is converted into a non-uniform original-time schedule. Given a desired step count K, we use a uniform grid in warped time: $\tau_0 = 1 > \tau_1 > \cdots > \tau_K = 0, \tau_k = 1 - k/K$. Generation then recursively applies the same student:

$$x_{\tau_{k+1}} = \widetilde{M}_\theta(\tau_k, \tau_{k+1}, x_{\tau_k}), \qquad k = 0, \ldots, K-1. \tag{14}$$

The same checkpoint is evaluated under multiple step counts without retraining. The complete training loop is summarized in Appendix B.5.

2.4 Relation to prior works

Our method differs from prior fast diffusion paradigms along two axes: the *network prediction target (i.e., loss design)* and the *problem formulation via time reparameterization*.

Versus noise-prediction diffusion training. Standard diffusion models predict noise, score, or velocity at a single time t, optimizing locally within the denoising field, and rely on external numerical solvers to propagate states (Ho et al., 2020; Song et al., 2021b). This was extended by fixing a teacher model and employing dedicated high-order samplers such as DPM-Solver and DPM-Solver++ (Lu et al., 2022a;b). In contrast, along the axis of network prediction target, our student directly predicts a transition map $M_\theta(t, s, x_t)$ between arbitrary time pairs, shifting the loss from score space into transport and composition space.

Versus consistency-style and continuous-time local projection models. Consistency Models (CMs) learn a projection-like map from noisy to cleaner states for one-step generation (Song et al., 2023), while Consistency Trajectory Models (CTMs) extend this to trajectory-level mappings along the probability-flow ODE (Kim et al., 2024), both related to our network prediction target axis. However, standard CMs suffer from step-count degradation; Sabour et al. (2025) proposed continuous-time flow-map distillation objectives robust under composition, aligning with our semigroup stability

emphasis. DE-CM (Dong et al., 2026) further improves training stability via critical sub-trajectory selection, boundary regularization, and explicit noise-to-noise modeling, highlighting that effective few-step generation depends on how modeling capacity is distributed along the trajectory, not merely on matching teacher states. Continuous-time consistency models (Lu & Song, 2025) and truncated consistency models (Lee et al., 2025) improve one- and two-step quality, while inductive moment matching (Zhou et al., 2025), shortcut models (Frans et al., 2025) and MeanFlow (Geng et al., 2025) train few-step generators without per-budget retraining. Shortcut models in particular require one large step to agree with two half steps, which is a semigroup-type self-consistency condition on the step size. Along the time reparameterization axis, DG-TWFD goes further by introducing a *monotone time reparameterization* guided by empirical semigroup defect, allocating denser effective time resolution to high-defect regions.

Versus explicit trajectory straightening. Rectified-flow methods such as Flow Matching (Lipman et al., 2023) and Rectified Flow (Liu et al., 2023) learn straighter transport trajectories to simplify sampling (Lee et al., 2024), inspiring reflow-based acceleration such as InstaFlow (Liu et al., 2024). However, Wang et al. (2025) suggest their effectiveness stems less from geometric straightness than from matched teacher-generated pairs. Our method preserves the teacher trajectory while reshaping its *time geometry* via semigroup defect, not an external geometric constraint.

Versus pre-defined adaptive time scheme. Some prior works also modify the time schedule: Karras et al. (2022) reformulate diffusion time via noise-level parameterization, Sabour et al. (2024) align solver- and model-specific sampling schedules, Pei et al. (2025) optimize discrete step sizes during sampling, and Stancevic et al. (2025) select entropy-based sampling points. The timestep shift of Esser et al. (2024) moves the timestep distribution toward high noise with the closed-form map $p' = fp/(1 + (f - 1)p)$, chosen per resolution. These methods prescribe or optimize schedules for a fixed generative process. In contrast, DG-TWFD learns a continuous monotone time coordinate from model-induced semigroup defect and couples it with the student transition map. Relative to a timestep shift, three differences follow: the warp is estimated from the student's own composition failures instead of being fixed a priori; the same bin masses drive both the training curriculum and the inference schedule, whereas a shift acts at inference only; and a shift is a one-parameter member of the monotone-warp family over which g_ϕ ranges. Appendix C.2 compares the two directly.

Summary of the technical positioning of DG-TWFD. Beyond the methods discussed above, recent approaches combine elements from multiple paradigms: SCoT unifies consistency models and rectified flows via straight-consistent trajectories, while TraFlow distills rectified flows using joint output, velocity, and self-consistency constraints (Wu et al., 2025a;b). Collectively, they reveal two key bottlenecks for few-step generation: *composition error* in trajectory maps and *trajectory heterogeneity* over time, corresponding precisely to our prediction target and time reparameterization axes. DG-TWFD unifies both: unlike prior work treating them independently, we show that the semigroup defect provides a principled signal for time reparameterization.

3 Experiments

We evaluate DG-TWFD on unconditional CIFAR-10 (Krizhevsky & Hinton, 2009) and class-conditional ImageNet64 derived from ImageNet (Deng et al., 2009), following the low-step evaluation protocols used by CMs and CTMs (Song et al., 2023; Kim et al., 2024). For CIFAR-10, we report Fréchet Inception Distance (FID) and Inception Score (IS); for ImageNet64, we report FID and recall following prior work. FID is computed from $50\,000$ generated samples per step, and all reported $1/2/4/8$-step DG-TWFD evaluations reuse the same selected EMA checkpoint. In addition, we track the *mean semigroup defect* Δ_{mean} on $10\,240$ held-out teacher triplets by averaging the ℓ_2 discrepancy between direct and composed transitions (640 teacher trajectories, 16 triplets each, fixed seed; Appendix D).

We consider an EDM/CTM-style U-Net (Ronneberger et al., 2015) as the student backbone, but replace the standard denoising interface with the explicit transition map used in Section 2, and train it from scratch. Concretely, the network is exposed as $M_\theta(t, s, x_t)$: the source state x_t is processed with EDM-style preconditioning, while the source and target times (t, s) are encoded as transition conditioning. The network is therefore trained to predict the target state x_s, rather than only a

denoised estimate associated with the source time t. This keeps the backbone comparable to prior few-step diffusion students while aligning its input-output interface with the compositional flow-map objective. The related hyperparameters are found in Appendix B.6.

4 RESULTS

Main findings. We report evaluation results for our models against several diffusion models in Tabs. 1 across two datasets using FID score. Baseline model scores are obtained following the settings in their original papers. While we do not exhaustively reproduce all prior results, our model achieves competitive FID scores across standard baselines used in diffusion distillation methods.

Table 1: **Comparison on CIFAR-10 and ImageNet64.** Reused denotes whether a single model can be directly applied to multi-step evaluation (yes) or needs to be retrained for different steps (weak). '-' represents scores not reported or that cannot be reproduced. NFE: number of function evaluations. **Bold**: best value in FID columns within a dataset. DG-TWFD rows report the final evaluation of Tab. 2.

Dataset	Method	Reused	FID_{1-step}	FID_{2-step}
CIFAR-10	Knowledge Distillation (Luhman & Luhman, 2021)	weak	9.36	-
	Fast Sampling (Zheng et al., 2023)	weak	3.78	-
	Diff-Ins (Luo et al., 2023)	weak	4.53	-
	PD (Salimans & Ho, 2022; Song et al., 2023)	weak	8.34	5.58
	Zheng et al. (2023)	weak	4.12	-
	CD (Song et al., 2023)	weak	3.55	2.93
	TRACT (Berthelot et al., 2023)	weak	3.78	3.32
	sCT (Lu & Song, 2025)	weak	2.97	2.06
	sCD (Lu & Song, 2025)	weak	3.66	2.52
	TraFlow-16M (Wu et al., 2025b)	yes	5.8	5.0
	TraFlow-28M (Wu et al., 2025b)	yes	4.5	4.0
	TCM (Lee et al., 2025)	weak	2.70	2.08
	IMM (Zhou et al., 2025)	yes	3.20	2.24
	CTM (w/o post-training) (Kim et al., 2024)	yes	7.10	4.73
	DG-TWFD (ours)	yes	**2.62**	**2.05**
ImageNet64	PD (Salimans & Ho, 2022; Heek et al., 2024; Salimans et al., 2024)	weak	15.39	8.95
	Fast Sampling (Zheng et al., 2023)	weak	7.83	-
	Diff-Ins (Luo et al., 2023)	weak	5.57	-
	Zheng et al. (2023)	weak	8.35	-
	CD (Song et al., 2023)	weak	6.20	4.70
	TRACT (Berthelot et al., 2023)	weak	7.43	4.97
	CTM (w/o post-training) (Kim et al., 2024)	yes	10.5	7.2
	DG-TWFD (ours)	yes	**4.86**	**3.34**

Notably, **our method supports multi-step evaluation**, whereas most of the above baselines do not. In addition, our checkpoint can be reused across different numbers of inference steps without retraining, demonstrating its flexibility. We report results under 1/2/4/8-step settings in Table 2, using FID, Inception Score, or recall depending on the dataset.

Table 2: **Multi-step evaluation of DG-TWFD on CIFAR-10 and ImageNet64.** CIFAR-10 is evaluated using FID and IS, while ImageNet64 is evaluated using FID and recall.

Dataset	CIFAR-10				ImageNet64			
Steps	1	2	4	8	1	2	4	8
FID ↓	2.62	2.05	1.84	**1.81**	4.86	3.34	2.58	**2.49**
IS / Recall ↑	9.63	9.79	9.86	**9.87**	0.64	0.66	**0.67**	**0.67**

Besides, our method exhibits strong performance even with few inference steps; increasing the number of inference steps consistently improves generation quality, demonstrating that our framework

supports effective iterative refinement. The improvements gradually saturate at higher numbers of steps. We attribute this to the fact that multi-step inference introduces intermediate states that act as implicit structural guidance, reducing per-step uncertainty and compositional error accumulation.

Importance of semigroup defect constraint A defect-based method is only justified if the defect is set by the training objective and decides how a checkpoint behaves across step counts. We test both points with a single-variable intervention (Tab. 3). Two CIFAR-10 runs of a CTM-style student share every setting and differ only in the target time of the consistency objective: *anytime* targets train transitions between arbitrary time pairs and thus impose the semigroup property, while *endpoint* targets train only jumps to the clean end, as in consistency models.

The two runs coincide at 1K updates (Δ_{mean} 0.0648 vs. 0.0647) and separate from 2K on, after which the endpoint run carries a 23–59% higher defect. The consequence is specific to composition. At 7K updates one-step FID is the same (7.05 vs. 7.10), whereas four-step FID nearly doubles (5.24 vs. 10.29). The four-step schedule visits grid indices $0, 4, 8, 13, 17$; its three intermediate targets are never trained by the endpoint run, whose four-step FID stays between 10.1 and 10.6 from 4K updates onward while the anytime run keeps improving to 5.24. The semigroup defect is therefore the quantity that governs reuse across budgets, and it is largely independent of single-step quality.

Table 3: **Removing the semigroup constraint.** Two CIFAR-10 runs of the same CTM-style student without time warping, identical in every hyperparameter, initialization and evaluation setting except the target time s of the consistency objective: *anytime* samples $s \in (\sigma_{\min}, t)$, which trains transitions between arbitrary time pairs; *endpoint* fixes $s = \sigma_{\min}$, as in consistency models. Δ_{m}: mean semigroup defect on 10,240 held-out teacher triplets. FID: 50k samples per step count. All ten checkpoints are listed in Appendix F.1. Up.: update steps.

Up.	Target s	$\Delta_{\mathrm{m}} \downarrow$	FID$_1$ $\downarrow$	FID$_2$ $\downarrow$	FID$_4$ $\downarrow$	FID$_8$ $\downarrow$
2K	anytime	**0.0488**	**40.30**	**11.12**	**11.39**	7.73
	endpoint	0.0600	55.59	17.26	16.05	**7.64**
4K	anytime	**0.0310**	**8.85**	**5.29**	**6.06**	**4.93**
	endpoint	0.0494	9.16	8.37	10.53	5.47
7K	anytime	**0.0275**	**7.05**	**5.18**	**5.24**	**4.37**
	endpoint	0.0389	7.10	6.01	10.29	5.23

The time warp also lowers the defect. In a latent ImageNet-256 pair that differs only in whether the warp is updated (the two runs log identical losses for the first 250 updates, i.e. until the first warp update), the warped run has 6–12% lower Δ_{mean} at each of five matched checkpoints between 2K and 10K updates (Appendix F.2). We have also separately evaluated defect quality and single-step generation quality in the Appendix C.4, using FID scores to provide a fair and consistent assessment. We also report an ablation of the loss terms in Appendix C.1.

Table 4: **Time-coordinate design ablation.** We compare a fixed identity clock against the full defect-guided warped-time design under the same transition-map framework.

	CIFAR-10 FID$\downarrow$				ImageNet64 FID$\downarrow$			
time design	**1-step**	**2-step**	**4-step**	**8-step**	**1-step**	**2-step**	**4-step**	**8-step**
EDM reference (Karras et al., 2022)	315.46	467.60	89.82	8.85	623.86	438.52	93.12	11.65
Optimal Stepsize baseline (Pei et al., 2025)	5.92	6.33	6.41	6.62	9.27	8.93	9.24	9.82
Entropic schedule (Stancevic et al., 2025)	6.16	6.31	6.66	6.98	7.22	9.42	9.80	11.01
Piecewise linear	16.39	13.52	13.37	15.46	18.44	16.51	13.93	9.88
Spline warp	13.51	9.35	11.23	11.21	15.68	14.46	11.92	8.84
Identity clock ($\tau(t) = t$)	20.84	11.92	6.87	4.96	26.40	15.72	8.94	5.41
Full DG-TWFD	**3.20**	**2.84**	**2.62**	**2.50**	**5.60**	**4.35**	**3.30**	**2.95**

Time reparameterization design Tab. 4 examines the necessity of the proposed time reparameterization. Under the same transition-map framework, replacing the learned warped coordinate with existing time-warping strategies on CIFAR-10 and ImageNet64 leads to consistently inferior performance. This indicates that learning a transport-adapted clock is essential rather than cosmetic, aligning with prior findings that sample quality is highly sensitive to the parameterization and discretization of the time/noise axis (Karras et al., 2022). Notably, the defect-guided time reparameterization learned by DG-TWFD yields the best performance. We also visualise the effects of time reparameterization on held-out CIFAR-10 teacher trajectories in Fig. 2. The learned clock assigns high local warped-time density to regions with high empirical semigroup defect.

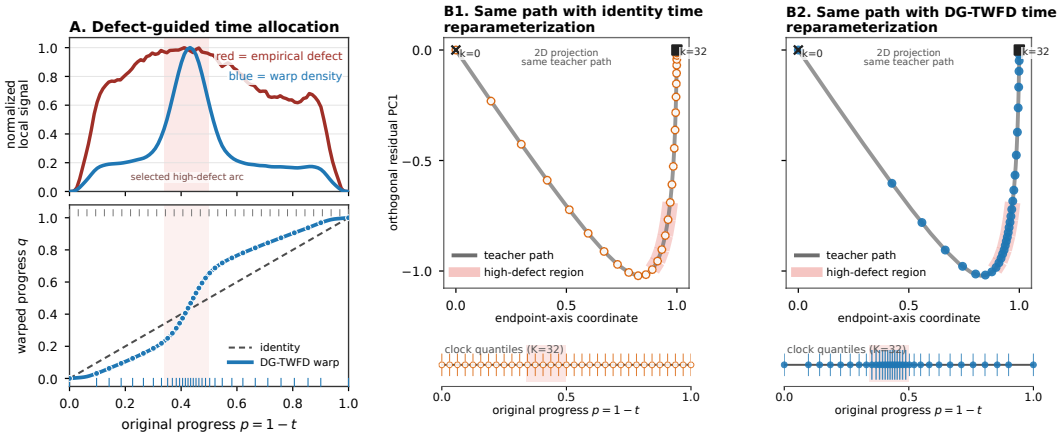


Figure 2: **Mechanism of defect-guided time warping.** (A) Empirical semigroup defect gives the local difficulty signal, while the blue warp density gives the learned clock allocation used to form quantiles. (B1–B2) Both panels project the same teacher path and show exact original-progress quantiles: identity time is nearly uniform, whereas DG-TWFD concentrates more quantiles on the selected high-defect arc. The $K = 32$ waypoints visualize time allocation only, not a change to the teacher path or evaluation steps.

Scaling and qualitative behavior Scaling results in Appendix C.3 show that the base model captures most gains, the large model gives smaller but consistent average FID reductions on both datasets (0.36 and 0.59), and longer training reduces semigroup defect, improving multi-step reuse more than single-step quality alone. For visualisation, Fig. 3 compares low-step samples, using class-controlled CIFAR-10 rows and class-locked ImageNet64 rows; additional panels are in Appendix C.5.

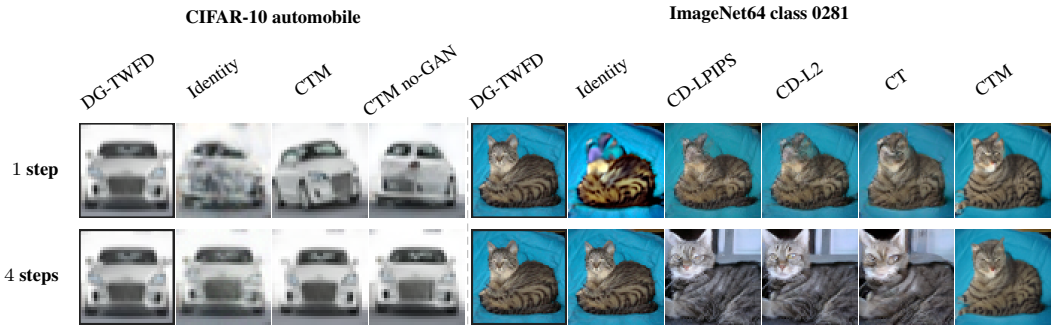


Figure 3: **Low-step qualitative comparison.** Rows fix the inference budget and columns fix the method for a CIFAR-10 automobile seed and ImageNet64 class 0281. DG-TWFD preserves the target category and object layout more consistently at one and four steps.

5 Conclusion

We presented DG-TWFD, a diffusion distillation framework that treats few-step robustness as a problem of compositional time geometry. By combining a continuous-time flow-map student with a learnable monotone time warp and semigroup-defect minimization, DG-TWFD improves multi-step consistency under aggressive step reduction while enabling checkpoint reuse across step counts. Experiments and controlled ablations show that the time warp reduces the semigroup defect and provides a principled way to allocate effective time resolution along the trajectory. Limitations include training cost, reliance on cached teacher trajectories, and gaps in one- and two-step performance and high-resolution evaluation. Future work will explore more expressive time reparameterizations, adaptive loss weighting, and scaling the framework to larger conditional generation settings.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, specifically to assist in writing and debugging the experiment and evaluation code. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, or interpret results, and cleaning and reformatting datasets and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, edit the paper to improve readability, identify relevant literature, and format references. We have reviewed all AI-assisted work. All experiments were conducted by the authors; the AI-assisted code was checked by the authors and validated by reproducing the published results of two baselines, and every reported number is taken from the output of our own evaluation runs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

For broader impacts, this work is foundational research and is not intended for high-risk or real-world decision-making systems. We do not identify direct safety, security, or human rights risks, as it does not involve physical systems, sensitive personal data, or critical infrastructure. This work uses only public benchmark datasets (CIFAR-10, ImageNet) and publicly released teacher models, and involves no human subjects. As with other generative models, faster sampling lowers the cost of producing synthetic images, including misleading ones; we release no new large-scale generator beyond the distilled students of existing public teachers.

REPRODUCIBILITY STATEMENT

Appendix D specifies the evaluation protocol: sample counts, evaluator, reference statistics, sampling grids, the construction of the 10,240 held-out teacher triplets used for Δ_{mean}, and the definitions of $D(k)$ and of the distance to the teacher. It also states how each re-measured baseline was run and how closely it reproduces its published value. Tables 12 and 9 state which checkpoint each of their measurements uses. Training hyperparameters and compute are in Appendix B.6, the training loop in Algorithm 1, and a five-seed re-evaluation in Appendix E. Code is available at the anonymous link in the abstract.

REFERENCES

David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation. *arXiv preprint arXiv:2303.04248*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. URL https://ieeexplore.ieee.org/document/5206848.

Linwei Dong, Ruoyu Guo, Ge Bai, Zehuan Yuan, Yawei Luo, and Changqing Zou. Dual-end consistency model. In *European Conference on Computer Vision*, 2026.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

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

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025.

Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

J Stuart Hunter. The exponentially weighted moving average. *Journal of quality technology*, 18(4): 203–210, 1986.

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

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

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. In *Advances in Neural Information Processing Systems*, 2024.

Sangyun Lee, Yilun Xu, Tomas Geffner, Giulia Fanti, Karsten Kreis, Arash Vahdat, and Weili Nie. Truncated consistency models. In *International Conference on Learning Representations*, 2025.

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

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

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and Qiang Liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2024.

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5775–5787, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

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

Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 76525–76546, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Jianning Pei, Han Hu, and Shuyang Gu. Optimal stepsize for diffusion sampling. *arXiv preprint arXiv:2503.21774*, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42947–42975. PMLR, 2024.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*, 2025.

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

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

Metod Saniga. Geometry of time and dimensionality of space. In Rosolino Buccheri, Metod Saniga, and William Mark Stuckey (eds.), *The Nature of Time: Geometry, Physics and Perception*, pp. 131–143. Springer Netherlands, 2003. doi: 10.1007/978-94-010-0155-7_14.

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

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.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pp. 32211–32252. PMLR, 2023.

Dejan Stancevic, Florian Handke, and Luca Ambrogioni. Entropic time schedulers for generative diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

Fu-Yun Wang, Ling Yang, Zhaoyang Huang, Mengdi Wang, and Hongsheng Li. Rectified diffusion: Straightness is not your need in rectified flow. In *International Conference on Learning Representations*, 2025.

Zhangkai Wu, Xuhui Fan, Hongyu Wu, and Longbing Cao. Scot: Unifying consistency models and rectified flows via straight-consistent trajectories. In *Advances in Neural Information Processing Systems*, 2025a.

Zhangkai Wu, Xuhui Fan, Hongyu Wu, and Longbing Cao. Traflow: Trajectory distillation on pre-trained rectified flow. *arXiv preprint arXiv:2502.16972v1*, 2025b.

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

Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *International Conference on Machine Learning*, pp. 42390–42402. PMLR, 2023.

Linqi Zhou, Stefano Ermon, and Jiaming Song. Inductive moment matching. In *International Conference on Machine Learning*, 2025.

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, 2024.

A THEORETICAL JUSTIFICATION

Our central claim is that *step-count robustness is a time-geometry problem*. Existing methods often attack the issue either by improving numerical integration on a fixed teacher trajectory or by imposing explicit geometric regularity in state space. We instead argue that the object controlling multi-step stability is the semigroup defect of the student, and that this defect can be reduced more naturally by adapting the time coordinate than by prescribing a straight state-space trajectory. We therefore prove the importance of re-parameterizinging the time mathematically as follows.

Suppose that under the learned warp, the teacher flow map is sufficiently smooth so that its local evolution admits a first-order expansion in u with bounded higher-order terms, in the sense that for small $\delta > 0$,

$$\widetilde{\Phi}_T(u \to u - \delta, x_u) = x_u - \delta\, \widetilde{v}_T(x_u, u) + \mathcal{O}(\delta^2). \tag{15}$$

Then, a student trained on local transitions incurs a composition error whose leading discrepancy can be interpreted as a higher-order effect. For a partition $\{u_k\}_{k=0}^{K}$ and sufficiently regular maps,

$$\left\| \widetilde{M}_\theta\left(u_0, u_K, x_{u_0}\right) - \widetilde{\Phi}_T\left(u_0, u_K, x_{u_0}\right) \right\| \lesssim \sum_{k=0}^{K-1} \varepsilon_k + C \sum_{k=0}^{K-2} \widetilde{\Delta}_\theta\left(u_k, u_{k+1}, u_{k+2};\, x_{u_k}\right), \tag{16}$$

where ε_k denotes the local teacher-matching error of the student on segment (u_k, u_{k+1}). This decomposition highlights that few-step degradation arises from both local approximation error and compositional inconsistency, motivating our joint treatment of both effects.

Since exact curvature of the teacher flow in state space is both expensive and overly restrictive, we instead regularize a practical surrogate built from cached teacher triplets. This surrogate measures whether adjacent warped-time teacher displacements become more uniform, which directly lowers the difficulty of composing learned transitions. Therefore, our framework replaces explicit state-space straightness with teacher-compatible temporal rectification.

We then show that our formulation generalizes existing state-space methods and reduces the semigroup defect via two boundary cases.

Boundary case 1: identity warp. If $g_\phi(t) = t$, then DG-TWFD reduces to standard continuous-time flow-map distillation. In this case, the framework degenerates to learning M_θ directly on the original teacher time axis:

$$\widetilde{M}_\theta(u, v, x) = M_\theta(u, v, x). \tag{17}$$

Therefore, our method strictly generalizes continuous-time map learning (Sabour et al., 2025). Any benefit from the proposed framework must come from learning a nontrivial time geometry, rather than from restricting the hypothesis class of the student.

Boundary case 2: perfectly compositional student. If M_θ exactly equals the teacher flow map under the warped coordinate, then for any partition $u_0 > u_1 > \cdots > u_K$,

$$\widetilde{\Delta}_\theta(u_i, u_{i+1}, u_{i+2}; x) = 0, \tag{18}$$

and repeated composition is exact for all step counts. In practice, the student is approximate, but this boundary case clarifies that the desired property is semigroup compatibility, not merely small one-step error.

B IMPLEMENTATION DETAILS OF DG-TWFD

B.1 TRAJECTORY BANK AND INTERPOLATION

For each teacher sample n, we cache a sparse trajectory

$$\mathcal{T}^{(n)} = \{(t_i, x_{t_i}^{(n)})\}_{i=0}^{m}, \qquad 1 = t_0 > t_1 > \cdots > t_m = 0. \tag{19}$$

The interpolation operator $\mathcal{I}(\mathcal{T}^{(n)}, a)$ returns the teacher state at an arbitrary time $a \in [0, 1]$. In the implementation, if a coincides with a cached timestamp, we reuse the stored state directly; otherwise, we interpolate linearly between the two nearest cached teacher states on the same trajectory. This is sufficient because the bank is constructed on a dense teacher grid, and the reported training path uses cached states rather than online teacher simulation.

B.2 Optional boundary stabilizer

The main formulation in Section 2 already yields a complete method. We additionally consider a lightweight boundary stabilizer to improve the first jump near the high-noise endpoint, where teacher dynamics are often most fragile for low-step generation.

Let $t_{\max}$ denote the noisiest endpoint and let $\delta > 0$ be a small step toward the clean side. We introduce an optional boundary corrector

$$B_\psi(x_{t_{\max}}) \approx x_{t_{\max}-\delta}, \tag{20}$$

optimized by

$$\mathcal{L}_{\mathrm{bdry}} = \mathbb{E}\left[\|B_\psi(x_{t_{\max}}) - x_{t_{\max}-\delta}\|_2^2\right]. \tag{21}$$

To avoid wasting capacity on trajectories that are irrelevant to later composition, this module is gated: it is activated only in early training or when the estimated local defect near $t_{\max}$ exceeds a threshold. Formally, with a binary gate $w_{\mathrm{bdry}} \in \{0, 1\}$,

$$\mathcal{L}_{\mathrm{bdry}} = \mathbb{E}\left[w_{\mathrm{bdry}}\,\|B_\psi(x_{t_{\max}}) - x_{t_{\max}-\delta}\|_2^2\right]. \tag{22}$$

This term is not part of the core method and is disabled in the corresponding ablations. Its role is to locally stabilize the earliest high-noise transition when that transition empirically dominates the later error budget. This design is consistent with the main framework because it targets the same compositional objective: if the first jump is unstable, then all later compositions inherit amplified error. The boundary stabilizer is therefore a local intervention for the most fragile endpoint without changing the global principle of time-warped semigroup reduction.

B.3 EMA defect and variation profiles

The adaptive schedule is driven by two bin-level EMA profiles on the shared partition $\mathcal{P} = \{I_b\}_{b=1}^B$. For a minibatch, we first aggregate the observed normalized defects and teacher-variation scores inside each visited bin:

$$\bar{d}_b = \mathrm{Avg}\left\{\bar{\Delta}_\theta(t, s, r; x_t) : s \in I_b\right\}, \qquad \bar{v}_b = \mathrm{Avg}\left\{V(t_3, t_2, t_1) : t_2 \in I_b\right\}. \tag{23}$$

If a bin is not visited in the current minibatch, its value is left unchanged. Otherwise, we update

$$\widehat{d}_b \leftarrow (1 - \alpha_{\mathrm{ema}})\widehat{d}_b + \alpha_{\mathrm{ema}}\bar{d}_b, \qquad \widehat{v}_b \leftarrow (1 - \alpha_{\mathrm{ema}})\widehat{v}_b + \alpha_{\mathrm{ema}}\bar{v}_b. \tag{24}$$

Before forming the adaptive mass, both EMA profiles are normalized by their across-bin running mean:

$$\widetilde{d}_b := \frac{\widehat{d}_b}{\epsilon_{\mathrm{norm}} + \frac{1}{B}\sum_{j=1}^B \widehat{d}_j}, \qquad \widetilde{v}_b := \frac{\widehat{v}_b}{\epsilon_{\mathrm{norm}} + \frac{1}{B}\sum_{j=1}^B \widehat{v}_j}. \tag{25}$$

In the main experiments, α_{ema} is fixed within each run. During warm-up, we set $\gamma = 0$, so the profiles are collected but do not yet alter the schedule; after warm-up, γ is switched to its target value. The teacher-variation profile $\widehat{v}_b$ can be precomputed from the trajectory bank or updated online with the same EMA rule; we use the online form for notational uniformity.

B.4 Executable warped-time triplet sampler

Given the current adaptive masses π_b and the induced monotone warp g, the training triplet sampler proceeds as follows. First, draw a bin $b \sim \pi$. Then, sample the warped midpoint inside the corresponding warped interval:

$$\tau_s \sim \mathrm{Uniform}\big(g(\xi_{b-1}), g(\xi_b)\big). \tag{26}$$

Next, sample left and right spans $\Delta_-, \Delta_+ \sim p_\Delta$, and set

$$\tau_t = \min(1, \tau_s + \Delta_+), \qquad \tau_r = \max(0, \tau_s - \Delta_-). \tag{27}$$

If clipping breaks the strict ordering $\tau_t > \tau_s > \tau_r$, we resample the spans. Finally, map back to the original diffusion times:

$$t = g^{-1}(\tau_t), \qquad s = g^{-1}(\tau_s), \qquad r = g^{-1}(\tau_r). \tag{28}$$

This sampler makes the conditioned warped-time triplet construction fully explicit and directly implementable.

B.5 FULL TRAINING ALGORITHM

The full algorithm is described in Algorithm 1. q_b^{base} represents the teacher sampling probability. While $\gamma = 0$, it falls back to normal linear time schedule, leading to $\pi_b = q_b^{base}$.

Algorithm 1 DG-TWFD training loop

Require: Teacher trajectory bank $\mathcal{B}$, student M_θ, bin partition $\mathcal{P}$, base bin masses q_b^{base}, warm-up length T_{warm}

Initialize EMA profiles $\widehat{d}_b = 0$ and $\widehat{v}_b = 0$ for all b
for each training iteration **do**
  if iteration $\le T_{\text{warm}}$ **then**
    Set $\gamma = 0$
  else
    Set γ to its target value
  end if
  Normalize the EMA profiles to obtain $\widetilde{d}_b$ and $\widetilde{v}_b$
  Compute adaptive masses

$$w_b = q_b^{\text{base}} \left(\epsilon_w + \widetilde{d}_b + \lambda_{\text{var}} \widetilde{v}_b\right)^\gamma, \qquad \pi_b = \frac{w_b}{\sum_j w_j}$$

  Construct the monotone warp g from the cumulative bin masses

$$g(\xi_b) = \sum_{j=1}^{b} \pi_j, \qquad g(t) = g(\xi_{b-1}) + \frac{t - \xi_{b-1}}{\ell_b} \pi_b, \;\; t \in I_b$$

  Sample a trajectory $\mathcal{T} \sim \mathcal{B}$ and a bin $I_b \sim \pi$
  Sample $\tau_s \sim \text{Uniform}(g(\xi_{b-1}), g(\xi_b))$
  Sample $\Delta_+, \Delta_- \sim p_\Delta$, set $\tau_t = \min(1, \tau_s + \Delta_+)$, $\tau_r = \max(0, \tau_s - \Delta_-)$, and resample if ordering fails
  Map to original time:

$$t = g^{-1}(\tau_t), \qquad s = g^{-1}(\tau_s), \qquad r = g^{-1}(\tau_r)$$

  Retrieve teacher states $x_t = \mathcal{I}(\mathcal{T}, t)$, $x_s = \mathcal{I}(\mathcal{T}, s)$, $x_r = \mathcal{I}(\mathcal{T}, r)$
  Evaluate student predictions:

$$\hat{x}_s = \widetilde{M}_\theta(t, s, x_t), \qquad \hat{x}_r^{\text{dir}} = \widetilde{M}_\theta(t, r, x_t), \qquad \hat{x}_r^{\text{cmp}} = \widetilde{M}_\theta(s, r, \hat{x}_s)$$

  Compute $\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{match}} + \lambda_{\text{def}} \mathcal{L}_{\text{def}} + \mathbf{1}_{\text{bdry}} \lambda_{\text{bdry}} \mathcal{L}_{\text{bdry}}$
  Update θ by gradient descent
  Compute the observed defect $\bar{\Delta}_\theta(t, s, r; x_t)$ and update the EMA defect profile of the bin containing s
  Sample a local warped-time triplet $\tau_3 > \tau_2 > \tau_1$, compute $V(t_3, t_2, t_1)$, and update the EMA variation profile of the bin containing t_2
end for
return student M_θ and final warp g

B.6 HYPERPARAMETERS AND ABLATIONS

Tab. 5 summarizes the student configurations used in the ablations and scaling studies, Tab. 6 reports the compute budget used for the final runs, and Tab. 7 gives the DDPM/DDIM teacher-bank record. Unless otherwise stated, the main tables use the base model. All models share the same high-level recipe: an offline teacher trajectory bank, direct endpoint prediction, defect-guided time allocation, and explicit semigroup-defect minimization. For DDPM-based CIFAR-10 teacher experiments, we use the public `google/ddpm-cifar10-32` checkpoint as the teacher and generate deterministic

Table 5: **DG-TWFD implementation summary.** Model hyperparameters are included. The base model is used for all main tables unless otherwise stated.

configs	small	base	large
params (M)	58	118	243
depth	12	16	24
hidden dim	384	512	768
attn heads	6	8	12
MLP ratio		4.0	
time-pair embed dim	128	192	256
warp bins	16	32	32
boundary head dim	96	128	192
optimizer		AdamW, $\beta = (0.9, 0.95)$	
learning rate	2e-4	2e-4	1.5e-4
weight decay		0.03	
EMA decay		0.9995	
training schedule		uniform in warped time + defect-adaptive resampling	
inference steps		$1/2/4/8$ with uniform warped-time partition	
conditioning	CIFAR-10: none	ImageNet64: class conditional	
λ_{def}		0.25	
λ_{warp}		0.10	
λ_{bdry}		0.05	
curriculum temperature η		2.0	
boundary threshold		0.015	

Table 6: **Compute summary.** Each training run uses one NVIDIA A100-SXM4-80GB GPU with fp16 mixed precision and dynamic loss scaling on a single-node server with 2 Intel Xeon Platinum 8378A CPUs and 1.0 TiB RAM.

experiment	updates	batch	GPUs	wall-clock	GPU-hours
CIFAR-10 main	50–70K	256	1	130–150h	130–150
CIFAR-10 time-warp ablation	30–50K	256	1	70–90h	70–90
CIFAR-10 no-warp baseline	30–50K	256	1	70–90h	70–90
ImageNet64 main	100–200K	64	1	180–260h	180–260
CIFAR-10 DDPM/DDIM teacher bank	–	–	1	$>$ 20h	$>$ 20
qualitative sampling	–	–	1	$<$ 1h	$<$ 1

trajectories with a 128-step DDIM rollout, retaining 33 states per trajectory. DDIM is only the rollout solver for the same DDPM checkpoint, not an independent teacher model. For ImageNet64, class-conditional banks are generated from the EDM ImageNet64 teacher checkpoint with Heun PF-ODE rollouts. The implementation keeps an online rollout fallback for cache misses, but the reported training path uses cached teacher trajectories. For the final main runs, CIFAR-10 uses batch size 256 for 50–70K updates, while ImageNet64 uses batch size 64 for 100–200K updates. The same selected EMA checkpoint is evaluated at $1/2/4/8$ steps by stepping uniformly in warped time and mapping the schedule back to the original teacher clock through g^{-1}.

C Additional results

C.1 Choice of matching loss

We investigate the design of the matching loss, which we compare with direct endpoint prediction (ours), residual prediction, and velocity-based prediction. Results show that direct endpoint prediction consistently outperforms the alternatives; further details are provided as below. More importantly, we evaluate the contribution of each loss term through an ablation study, with results shown in Fig. 4 reporting 4-step FID across two datasets. Using only $\mathcal{L}_{\mathrm{match}}$ leads to limited improvement and early saturation, indicating insufficient multi-step consistency. Augmenting with $\mathcal{L}_{\mathrm{def}}$ yields noticeable

Table 7: **DDPM/DDIM CIFAR-10 teacher bank.** DDIM is used only as a deterministic rollout solver for the same DDPM teacher checkpoint, not as a separate teacher model.

item	setting
teacher checkpoint	`google/ddpm-cifar10-32` Diffusers DDPM checkpoint
rollout solver	DDIM scheduler on the same DDPM pipeline, 128 inference steps
retained states	33 uniformly spaced states in the DG-TWFD time coordinate
time convention	$u = 0$ noise, $u = 1$ clean; internally mapped to DDPM/DDIM noise time $\tau = 1 - u$
trajectory splits	50K training trajectories and 10K validation trajectories
storage	fp16 shards of 1024 trajectories, about 13–16GB including shard overhead
seed	42
reuse	reused across DDPM/DDIM teacher, schedule, and time-warp ablations

gains, suggesting its effectiveness in enforcing agreement between direct and composed transitions. Further incorporating time warp strategy provides additional improvements and achieves the best performance overall, highlighting its role in shaping a more suitable time parameterization that facilitates learning such consistency. Together, these terms act synergistically to improve convergence and final sample quality.

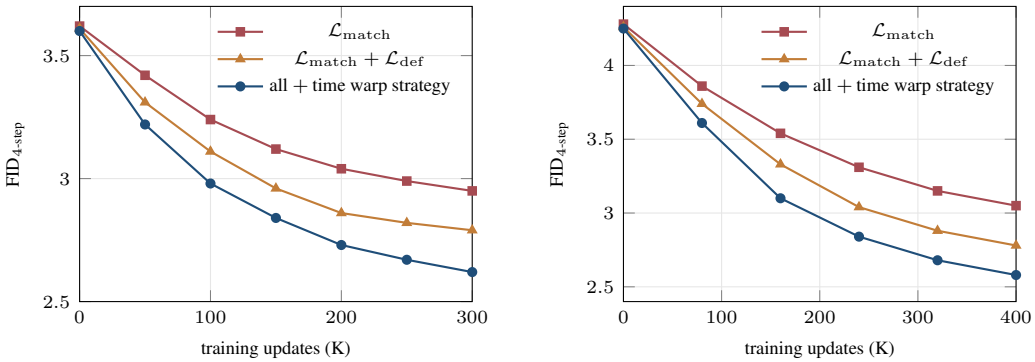


Figure 4: **Ablation of loss terms.** $\text{FID}_{4\text{-step}}$ versus training updates on CIFAR-10 (left) and ImageNet64 (right). Using only $\mathcal{L}_{\text{match}}$ leads to early saturation, while adding $\mathcal{L}_{\text{def}}$ provides consistent improvements by enforcing agreement between direct and composed transitions.

Tab. 8 studies which quantity the student should predict when the final objective is a time-pair transport map. We compare direct endpoint prediction, residual prediction, and velocity-style prediction under the same backbone, teacher bank, and training budget. The comparison is motivated by two representative target parameterizations from the literature: velocity-field regression in Flow Matching (Lipman et al., 2023) and residual-style prediction in TRACT (Berthelot et al., 2023). In our transport-map setting, however, both alternatives introduce an additional conversion layer before composition: residual prediction must be added back to the source state, and velocity prediction must be scaled by the time interval before it becomes a transition operator.

The result is consistent across both datasets. Direct endpoint prediction yields the lowest 4-step FID and the smallest semigroup defect, while residual prediction is competitive but consistently weaker, and velocity-style prediction performs worst. This trend supports the main design choice of DG-TWFD: the student is evaluated in transport space, so supervising the transported state x_s directly is better aligned with the recursive inference operator than predicting an intermediate quantity that must be post-converted before composition. Put differently, endpoint prediction removes an avoidable interface mismatch between training and test-time reuse.

A useful secondary observation is that the ranking of defect calculation mirrors the ranking in FID. Sec. 4 shows the same link between defect and multi-step FID in a controlled setting. This indicates that the benefit of endpoint prediction is not merely a better local regression target, but a more stable compositional map. Therefore, all subsequent DG-TWFD experiments adopt direct endpoint prediction as the default parameterization.

Table 8: **Prediction target ablation.** Direct endpoint prediction is consistently better than residual or velocity-style targets when the student is evaluated as a compositional transport map. The velocity-style row is motivated by velocity-field generative modeling (Lipman et al., 2023), while the residual row serves as a residual-style counterpart inspired by time-distillation baselines (Berthelot et al., 2023).

Prediction target	CIFAR-10		ImageNet64	
	$\text{FID}_{4\text{-step}}$ ↓	**Defect↓**	**$\text{FID}_{4\text{-step}}$ ↓**	**Defect↓**
velocity-style (v) (Lipman et al., 2023)	2.58	0.031	3.34	0.066
residual (Δx) (Berthelot et al., 2023)	2.29	0.025	2.96	0.056
direct endpoint (x_s)	**2.04**	**0.018**	**2.67**	**0.044**

C.2 Comparison with a timestep shift

To isolate the inference schedule we keep one checkpoint and change only the grid on which its steps are placed, using the SD3 shift $p' = fp/(1 + (f - 1)p)$ on the 18-point grid with $f \in \{0.5, 2, 3\}$, where $f < 1$ densifies the clean end (Tab. 9). No single shift is best at every budget: the default grid wins at two steps and f=0.5 at four and eight steps, where it improves on the default by 0.50 and 0.12 FID. Setting a shift therefore requires a per-budget FID sweep. The semigroup defect gives a label-free alternative: for each budget we average the normalized defect (Eq. (8)) over the grid's consecutive triplets on 256 held-out teacher trajectories and take the minimizer. This picks the FID-optimal grid at four and eight steps and misses at two steps, where it selects f=3. The time grid thus matters, and the defect can steer it without sample-quality evaluation. Whether the training-time warp alone outperforms a tuned inference shift remains open.

Table 9: **Inference grid vs. SD3 timestep shift** on one checkpoint (the 61K checkpoint of Tab. 12), CIFAR-10 FID with 50k samples; one-step sampling is unaffected by the grid. ✓ Grid selected by the minimum semigroup defect, computed without any FID evaluation. The default row uses the integer grid indices of our standard evaluation; shifted grids use fractional indices.

Inference grid	2 steps	4 steps	8 steps
Default	**2.93**	3.24	3.10
Shift f=0.5	3.29	**2.74**✓	**2.98**✓
Shift f=2	2.97	3.56	3.00
Shift f=3	3.42✓	3.20	3.02

C.3 Scaling on model size and training steps

Tabs. 10 and 11 show that the method scales predictably with model size and training steps. The base model already captures most of the gain, while the large model brings a smaller but consistent improvement on both datasets. Longer training mainly reduces semigroup defect and helps low-step reuse, which is why the gain is more visible in average multi-step FID than in one-step quality alone. The qualitative comparison in Fig. 5 complements the quantitative tables by showing how low-step outputs evolve under the same inference budgets.

C.4 Defect and single-step quality are separate axes

The ablation above already shows that the defect leaves one-step FID untouched. The same separation appears across models (Tab. 12). CTM and DG-TWFD have nearly the same end-to-end deviation $D(k)$ but different one-step FID; IMM has a roughly six-fold larger $D(k)$ yet competitive FID at every budget. The defect therefore does not rank single-budget quality, and we do not claim that it does. What it measures is whether the budgets agree. For IMM, a large $D(k)$ means that different budgets produce different images of similar quality, so the checkpoint does not act as a single map.

Table 10: **Backbone scaling on the completed benchmarks.** The base model already captures most of the benefit; the large model improves further but with diminishing returns.

model	params	updates	**CIFAR-10**		**ImageNet64**	
size	(M)	(K)	**FID**$_{1-step}$	**FID**$_{4-step}$	**FID**$_{2-step}$	**FID**$_{4-step}$
small	58	200	2.96	2.08	4.02	2.84
base	118	400	2.62	1.84	3.34	2.58
large	243	600	2.55	1.77	3.21	2.47

Table 11: **Training-budget scaling for the base model.** Longer training mostly helps through lower semigroup defect, which then translates into a better average low-step FID.

updates	teacher pairs	CIFAR-10 FID$_{mean}$	ImageNet64 FID$_{4-step}$	Defect$_{mean}$
200K	40M	2.21	3.06	0.024
400K	80M	2.08	2.74	0.020
600K	120M	2.01	2.58	0.017

For TCM, $D(k)$ grows from 21.7 to 88.4 gray levels between two and eight steps, and its FID degrades over the same range (2.08, 2.54, 2.87).

A small $D(k)$ could in principle come from outputs that agree while all being poor. We rule this out by measuring each output's distance to the teacher's reference solution, its 18-step Heun integration of the PF-ODE from the same noise on the grid the teacher is defined on. The one-step DG-TWFD output lies 11.7 gray levels from it, and the 2-, 4- and 8-step outputs 10.5, 10.2 and 9.7. The teacher's own 2-, 4- and 8-step Heun solutions lie 134.8, 45.1 and 10.9 gray levels away, so one student step lands nearly as close to the reference solution as eight teacher steps do (11.7 vs. 10.9; Appendix D). The student's budgets agree because each of them tracks the same ODE solution.

Table 12: **One checkpoint across budgets, under one protocol** (CIFAR-10). All rows are measured by us on a shared EDM grid (ρ=7) with the official checkpoints and samplers. FID: 50k samples, OpenAI evaluator. $D(k)$: RMSE between the one-step and the k-step output from the same noise, in gray levels (2048 samples). *The 61K checkpoint is an earlier checkpoint of the same recipe and the one available for these measurements; the final checkpoint behind Tab. 2 was not re-measured here.

	FID↓				$D(k)$ (gray levels)		
Model	1	2	4	8	k=2	k=4	k=8
CTM (official, with GAN)	1.91	1.86	1.88	1.97	7.6	8.5	10.1
DG-TWFD (61K)*	3.21	2.93	3.24	3.10	8.0	8.7	10.1
IMM	3.20	2.24	2.11	2.20	45.1	57.5	66.1
TCM	2.70	2.08	2.54	2.87	21.7	67.2	88.4

C.5 Visualization of generated samples

Across the reported benchmarks, the empirical picture is consistent with the qualitative comparison in Fig. 5. On CIFAR-10, DG-TWFD remains stable as the same checkpoint is reused across low-step budgets. On ImageNet64, the learned warp yields a better FID-recall tradeoff than conventional consistency-style baselines when the step budget is moderately relaxed. Fig. 3 contrasts methods at one and four steps. Figs. 6–9 (CIFAR-10) and Figs. 10–12 (ImageNet64) extend it with further class and seed pairs, each at $1/2/4/8$ steps.

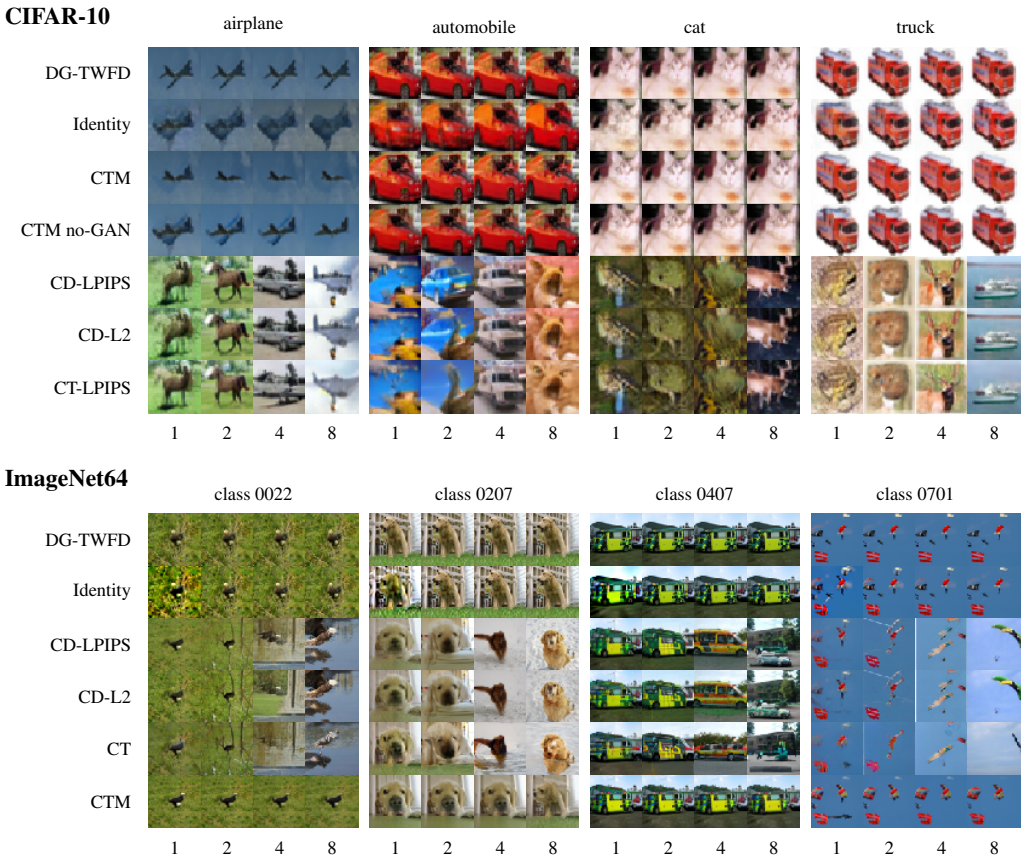


Figure 5: **Low-step qualitative comparison across datasets.** Each mini-grid fixes one seed/class and shows 1, 2, 4, and 8 sampling steps from left to right; model rows are labeled on the left. Rows use matched class labels and initial noise seeds for class-conditional methods. CIFAR-10 consistency-model JAX checkpoints are unconditional and therefore shown only as seed-only references.

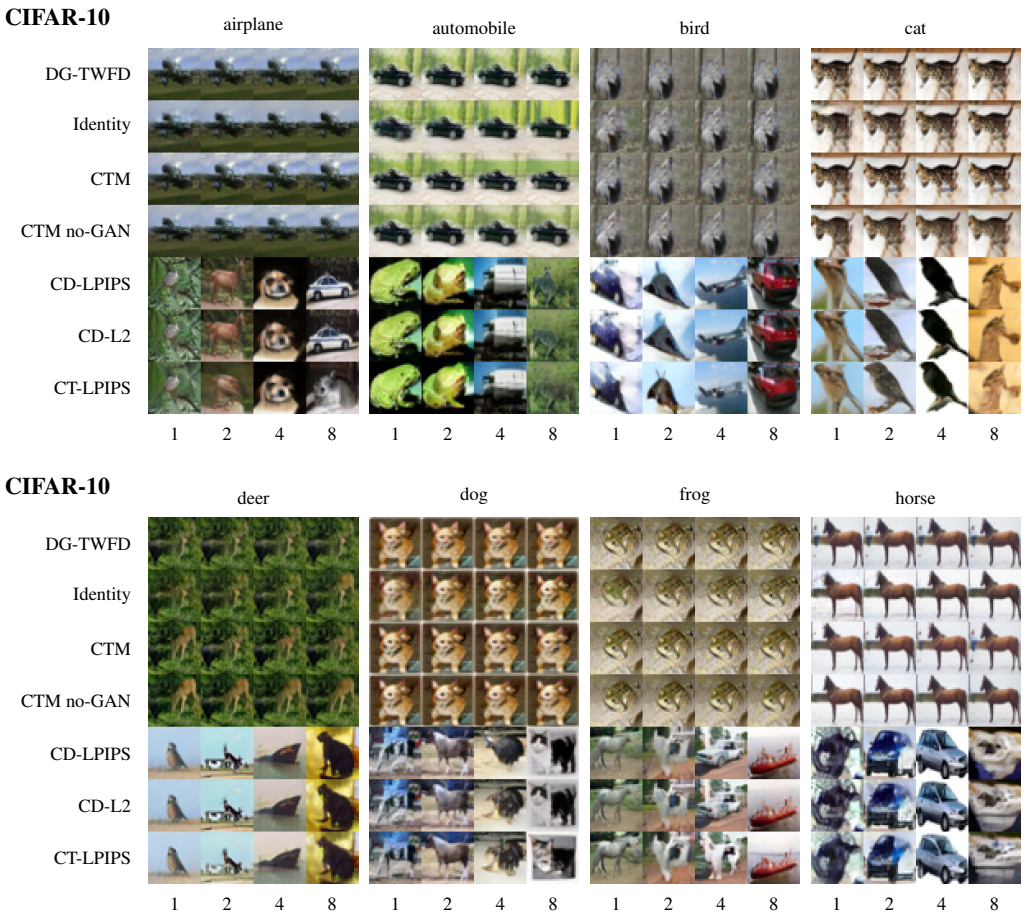


Figure 6: **CIFAR-10 qualitative sample bank, part 1.** Additional class/seed pairs from the sample bank follow the same row and $1/2/4/8$-step layout as the main qualitative panel; CIFAR-10 consistency-model JAX rows remain seed-only references.

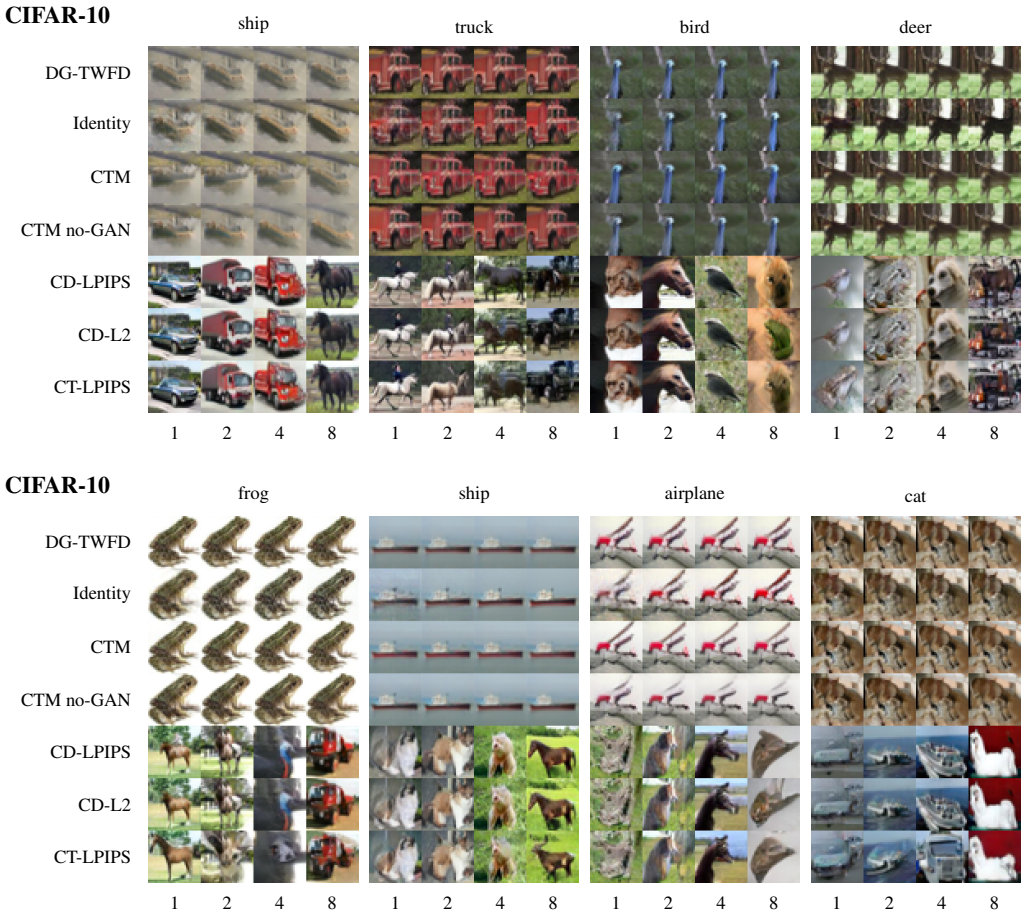


Figure 7: **CIFAR-10 qualitative sample bank, part 2.** The panel broadens the CIFAR-10 qualitative coverage while preserving the same class/seed alignment conventions as the main qualitative panel.

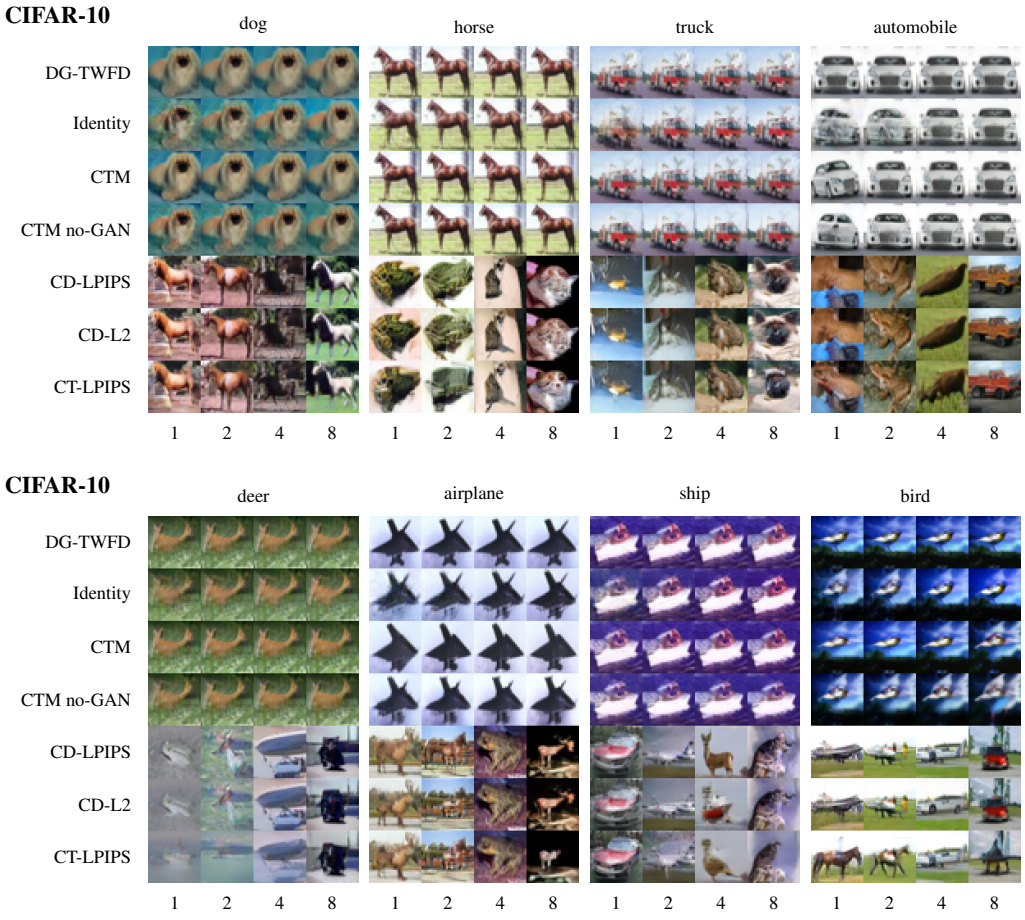


Figure 8: **CIFAR-10 qualitative sample bank, part 3.** These additional class/seed pairs provide broader qualitative coverage without changing the row ordering or step layout.

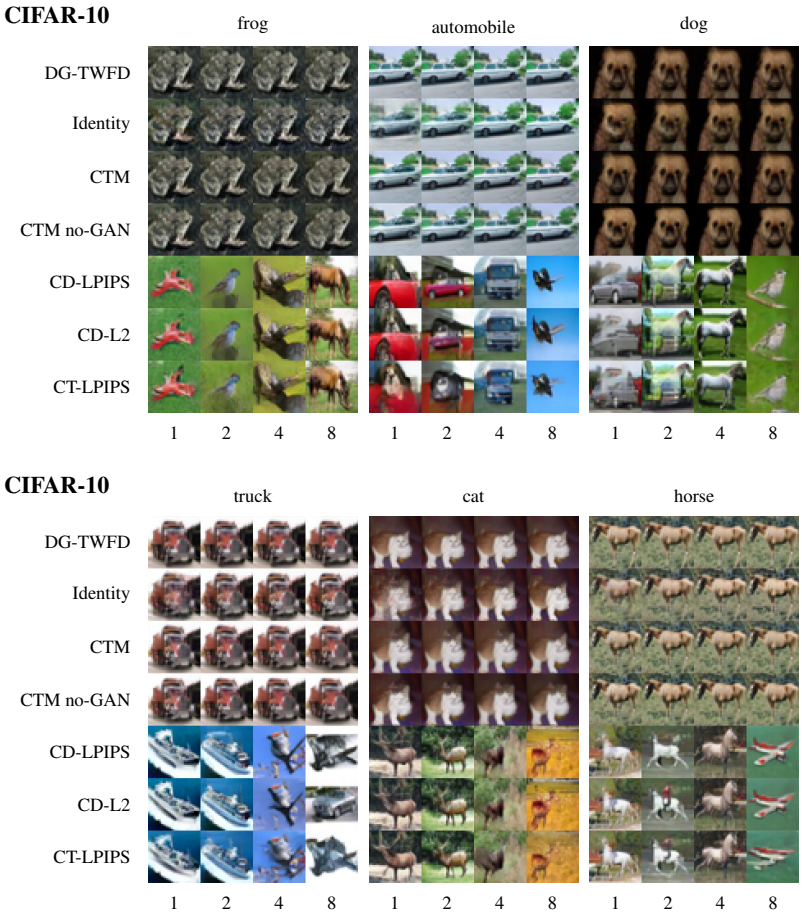


Figure 9: **CIFAR-10 qualitative sample bank, part 4.** The final CIFAR-10 sample-bank panel completes the 30 new class/seed pairs supplied in the qualitative bank.

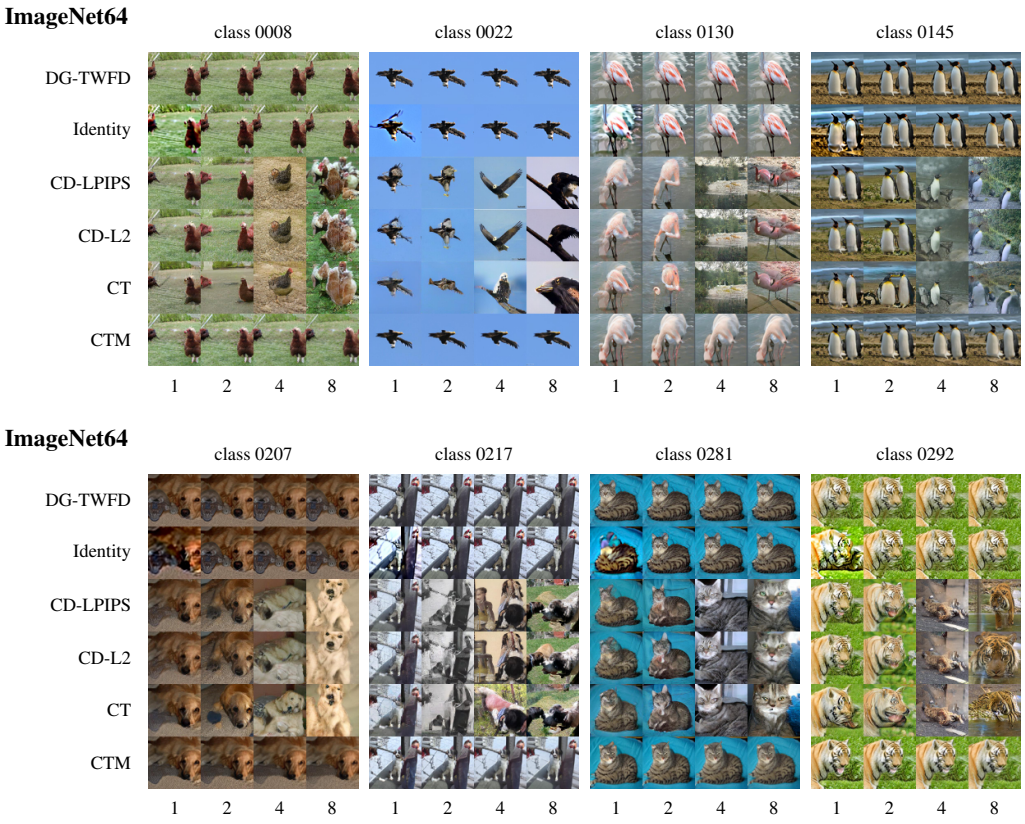


Figure 10: **ImageNet64 qualitative sample bank, part 1.** Additional class-locked ImageNet64 class/seed pairs from the sample bank follow the same row and $1/2/4/8$-step layout as the main qualitative panel.

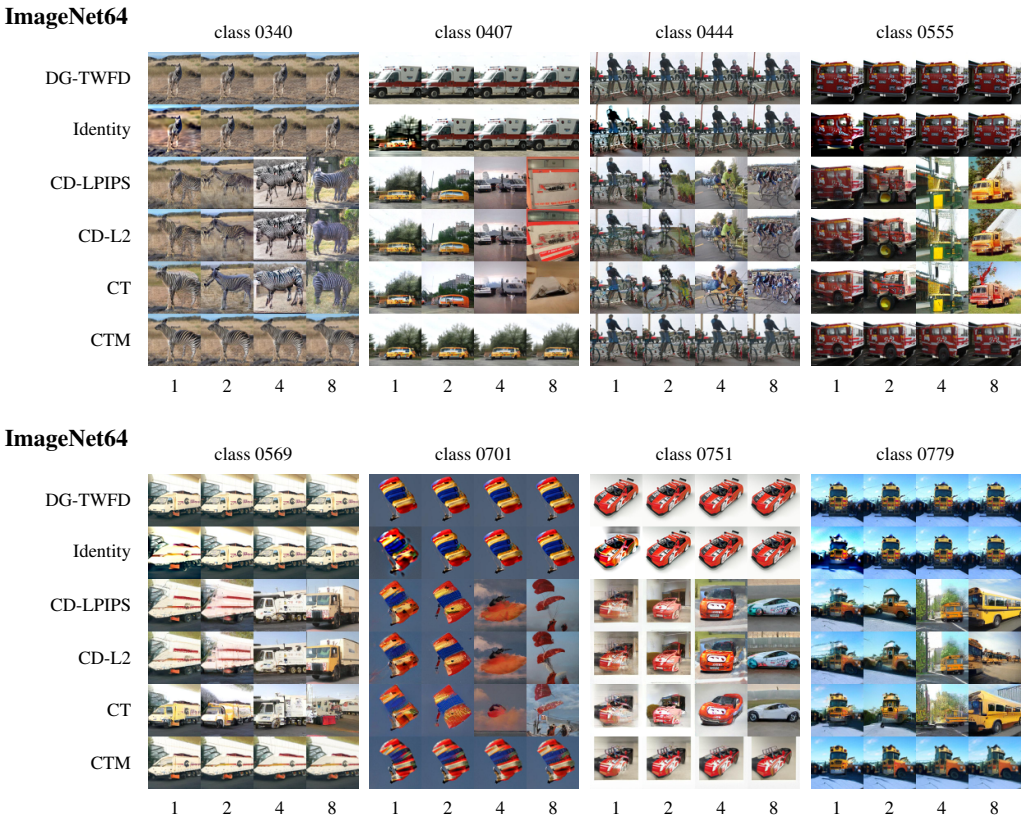


Figure 11: **ImageNet64 qualitative sample bank, part 2.** These additional ImageNet64 class/seed pairs provide broader coverage of object categories and visual structure under the same comparison format.

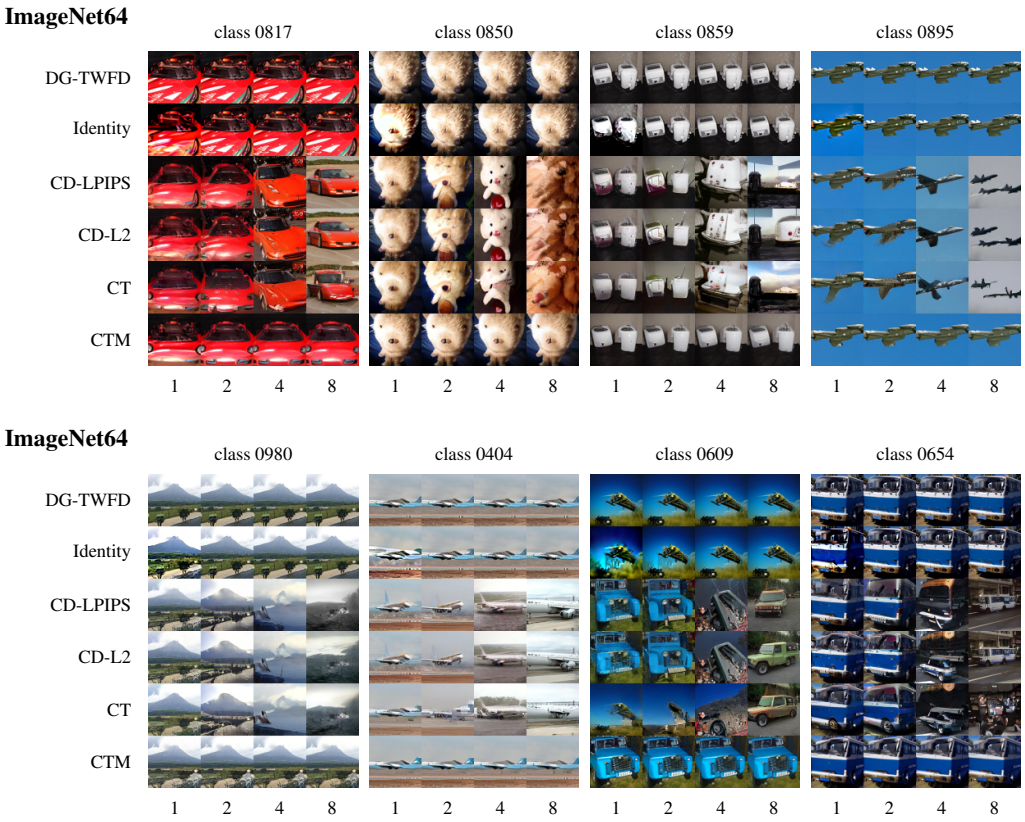


Figure 12: **ImageNet64 qualitative sample bank, part 3.** The final ImageNet64 sample-bank panel completes the 24 new class/seed pairs supplied in the qualitative bank.

D EVALUATION PROTOCOL AND DEFECT MEASUREMENTS

Sample quality. All CIFAR-10 FIDs in the main text use 50k generated samples, the OpenAI Inception evaluator and the reference statistics released with CTM, sampled with the exact (deterministic) transition-map sampler. Unless a grid is stated explicitly, a k-step evaluation places its steps at the integer indices $\mathrm{round}(i \cdot 17/k)$, $i = 0, \ldots, k$, of the 18-point Karras grid ($\rho = 7$, $\sigma \in [0.002, 80]$). The five-seed results of Tab. 2 repeat this evaluation with five sampling seeds on one EMA checkpoint.

Re-measured baselines. TCM, IMM and the official CTM checkpoint were evaluated with their released weights and samplers under the protocol above. TCM uses its official two-step intermediate time 0.821 and Karras midpoints for four and eight steps. IMM uses its official network and pushforward sampler; Tab. 12 places its steps on the shared EDM grid, and its official tuned two-step grid gives 1.982, reproducing the published 1.98. The official CTM gives 1.91 / 1.86 at one / two steps against 1.98 / 1.87 reported.

Mean semigroup defect Δ_{mean}. We roll the teacher out with Heun's method from 640 fixed-seed noise samples at $\sigma_{\max}$ and keep its state at every grid index. From each trajectory we draw 16 triplets $i < j < k$ uniformly without replacement from the 18 indices, giving 10,240 triplets shared by all models. For each triplet we evaluate Eq. (1) with x_t the teacher state at index i, i.e. the ℓ_2 discrepancy between $M_\theta(i \to k)$ and $M_\theta(j \to k) \circ M_\theta(i \to j)$, and report the unnormalized mean in the data's $[-1, 1]$ scale. Models are loaded with their own conditioning; loading a class-conditional checkpoint as unconditional silently drops its label embedding and inflates the defect about three-fold, so we verify that every checkpoint loads with no missing or unexpected parameters.

End-to-end deviation $D(k)$ and distance to the teacher. $D(k)$ is the per-sample RMSE between the one-step and the k-step output generated from the same noise, averaged over 2048 samples and reported in gray levels ($\times 127.5$). It needs no triplet sampling and applies to endpoint models, which is why Tab. 12 uses it across method families. The distance to the teacher is the same RMSE between a model's k-step output and the teacher's 18-step Heun solution from the same noise.

Table 13: **Mean semigroup defect Δ_{mean} across models** (CIFAR-10, same 10,240 triplets). All rows are unconditional.

Model	Updates	$\Delta_{\mathrm{mean}} \downarrow$
CTM-style student, anytime targets, no warp	1K	0.0648
CTM-style student, anytime targets, no warp	7K	0.0275
DG-TWFD	21K	0.0243
DG-TWFD	44K	0.0240
DG-TWFD	61K	0.0258
CTM (official, with GAN)	95K	0.0240

Tab. 13 shows where the defect varies. It falls by 58% within the first 7K updates and then settles in a narrow band (0.0240 to 0.0258) for the well-trained models of this family. Along the DG-TWFD trajectory it is lowest at 44K updates and rises to 0.0258 at 61K, after an adversarial perceptual stage was added between the two; that stage improves one-step FID while giving back part of the composition consistency.

E MULTI-SEED RESULTS

Tab. 14 lists a five-seed re-evaluation of DG-TWFD. Its means lie within 0.05 FID of the final single evaluation in Tab. 2 at every step count on both datasets.

Table 14: **Five-seed re-evaluation** reported during the review period (sampling seeds 0–4, 50k samples per seed and step count). The main-text results of Tab. 2 are the final single evaluation.

	CIFAR-10 FID↓				CIFAR-10 IS↑				ImageNet64 FID↓				ImageNet64 Recall↑			
Seed	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8
0	2.44	1.93	1.74	1.74	9.68	9.83	9.90	9.93	4.88	3.37	2.64	2.50	.656	.662	.682	.679
1	2.69	2.10	1.90	1.88	9.60	9.86	9.85	9.85	4.69	3.29	2.48	2.42	.638	.652	.668	.678
2	2.85	2.21	1.96	1.93	9.55	9.76	9.85	9.80	5.00	3.40	2.63	2.55	.627	.647	.658	.670
3	2.54	2.00	1.81	1.80	9.59	9.70	9.80	9.83	5.04	3.37	2.64	2.57	.636	.652	.668	.668
4	2.67	2.10	1.91	1.83	9.68	9.86	9.97	9.94	4.93	3.31	2.64	2.50	.640	.648	.663	.662
mean	2.64	2.07	1.86	1.84	9.62	9.80	9.87	9.87	4.91	3.35	2.61	2.51	.639	.652	.668	.671

F Additional experiments

F.1 Full semigroup ablation

Tab. 15 reports every 1K checkpoint of both runs in Tab. 3. The two runs start from nearly the same Δ_{mean} at 1K and separate from 2K onward.

Table 15: **All checkpoints of the semigroup ablation** (Tab. 3). The endpoint run was trained for one additional 1K segment.

	anytime targets $s \in (\sigma_{\min}, t)$					endpoint targets $s = \sigma_{\min}$				
Updates	Δ_{mean}	FID_1	FID_2	FID_4	FID_8	Δ_{mean}	FID_1	FID_2	FID_4	FID_8
1K	0.0648	204.88	59.17	35.91	15.72	0.0647	239.26	69.86	37.36	14.12
2K	0.0488	40.30	11.12	11.39	7.73	0.0600	55.59	17.26	16.05	7.64
3K	0.0348	13.15	5.82	6.84	5.29	0.0546	14.30	10.16	11.52	5.79
4K	0.0310	8.85	5.29	6.06	4.93	0.0494	9.16	8.37	10.53	5.47
5K	0.0293	7.75	5.20	5.65	4.70	0.0454	7.82	7.35	10.52	5.39
6K	0.0282	7.32	5.21	5.39	4.51	0.0418	7.34	6.52	10.47	5.26
7K	0.0275	7.05	5.18	5.24	4.37	0.0389	7.10	6.01	10.29	5.23
8K	–	–	–	–	–	0.0366	6.89	5.67	10.09	5.25

F.2 Latent ImageNet-256 with a DiT student

Setup. To test whether the mechanism transfers to a different space, teacher and backbone, we train DG-TWFD in the latent space of the Stable Diffusion VAE (Rombach et al., 2022) at 256×256 resolution. The teacher is the public class-conditional DiT-XL/2 (Peebles & Xie, 2023), wrapped as an EDM denoiser; the student is a 131M-parameter DiT with (t, s) conditioning. Training uses a bank of 20,000 latents generated by the teacher (guidance 1.5, 18 steps), an ℓ_2 objective in latent space, batch size 64, learning rate 10^{-4} and a single A100 GPU. A second run is identical except that the warp is never updated; the two runs log identical losses for the first 250 updates, until the first warp update.

Results. Tab. 16 lists the defect of the matched pair and the FID of the warped run. The warp lowers Δ_{mean} at each of the five matched checkpoints (ratio 0.88 to 0.94). From 4K updates on, the warped run's FID decreases strictly with the number of steps at all 33 evaluated checkpoints. Absolute FID is far from the teacher: FID is computed on 2048 samples, for which the teacher itself scores 21.32, and a student trained only on teacher-generated latents is bounded by the teacher. We therefore use this experiment as evidence for the mechanism and make no performance claim at this resolution.

F.3 Pixel-space DiT backbone on CIFAR-10

We replace the U-Net with a DiT-S/2 (33M parameters) with (t, s) conditioning and keep every other setting of the U-Net run. At 10K, 20K and 22K updates the one- to eight-step FIDs are 40.5/44.2/65.1/60.0, 28.9/30.5/38.0/47.9 and 27.6/28.8/36.3/45.0 (10k samples). The gap between

Table 16: **Latent ImageNet-256.** Left: Δ_{mean} (latent space, 10,240 teacher triplets) of the matched warp / no-warp pair. Right: FID of the warped run at 1/2/4/8 steps, 2048 samples (teacher: 21.32).

Updates	warp	no warp	ratio
2K	0.0645	0.0688	0.938
4K	0.0374	0.0424	0.884
6K	0.0372	0.0403	0.924
8K	0.0331	0.0367	0.902
10K	0.0317	0.0350	0.905

Updates	1	2	4	8
4K	423.5	403.2	378.7	366.9
10K	268.1	263.2	251.9	247.6
20K	197.7	195.2	190.9	186.1
52K	184.8	179.6	176.4	173.7

four and one steps shrinks from 24.5 to 8.7 FID over this range, and $D(4)$ falls from 22.1 to 18.0 gray levels, but within this budget the run does not reach the monotone multi-step behavior of the U-Net. Together with the latent experiment above, where a DiT student is monotone, we conclude that the mechanism does not depend on U-Net-specific structure, while the absolute quality reached within a given budget does depend on how well the backbone suits the data space.