GEOMETRY-AWARE TIME REPARAMETERIZATION FOR FLOW-MAP DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-map distillation enables one- and few-step generation by learning finite-time transitions of a pretrained generative ODE. We investigate whether changing the teacher's time parameterization can make these transitions easier to learn. Motivated by the hypothesis that trajectory segments with large normal acceleration are harder to distill, we propose a geometry-aware time reparameterization that allocates more student time to these regions while preserving the teacher's geometric paths and terminal distribution. We derive a shared clock that equalizes a population normal-acceleration statistic under suitable assumptions, and construct a practical approximation from robust, regularized estimates across teacher trajectories. We incorporate this clock into Lagrangian flow-map distillation, using the transformed time coordinate to condition the student. The clock is estimated once before distillation and requires neither teacher retraining nor additional student parameters or inference-time network evaluations. Experiments on synthetic data, CIFAR-10, and CelebA-64 show improved sample quality over identity-time distillation at matched inference budgets, including improvements in one- and two-step image generation. The gains in one-step generation, where no intermediate sampling times can be adjusted, highlight the benefits of time reparameterization during distillation.

1 INTRODUCTION

Diffusion and score-based generative models have emerged as a leading paradigm for learning complex data distributions (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b), enabling high-fidelity image synthesis and flexible conditioning at scale (Dhariwal & Nichol, 2021; Rombach et al., 2022). These models learn to reverse a gradual noising process, with generation described by reverse-time stochastic dynamics or an associated probability-flow ordinary differential equation (ODE). Their success, however, comes with substantial inference cost: producing a sample typically requires many sequential neural-network evaluations. Reducing this cost while preserving sample quality remains a central challenge, motivating extensive work on deterministic sampling and specialized numerical solvers (Song et al., 2021a; Lu et al., 2022).

Flow matching (FM) (Lipman et al., 2023) provides a flexible framework for learning continuous-time transport through a simulation-free regression objective. Together with closely related rectified-flow and stochastic-interpolant formulations (Liu et al., 2023; Albergo & Vanden-Eijnden, 2023), it learns a time-dependent velocity field that transports a simple source distribution to the data distribution, using analytically accessible conditional paths to construct training targets. This viewpoint accommodates diffusion probability paths as well as simpler alternatives, such as linear interpolation between noise and data, offering greater flexibility in the design of the generative dynamics. Importantly, however, straight conditional interpolation paths do not generally imply straight trajectories of the resulting marginal velocity field (Frans et al., 2025). Sampling therefore still requires numerical ODE integration, and using too few steps can introduce substantial discretization error.

One- and few-step generative models address this limitation by learning finite-time transport directly, either through distillation from a pretrained model (Salimans & Ho, 2022) or through direct training from data. Consistency models (Song et al., 2023) learn to map points along a probability-flow trajectory to a common endpoint, while consistency trajectory models (Kim et al., 2024) and flow map matching (FMM) (Boffi et al., 2025b) learn two-time maps connecting intermediate states along

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.

the dynamics. Related formulations include Shortcut Models (Frans et al., 2025), which learn step-size-conditioned updates by enforcing consistency between one large step and two smaller steps, and MeanFlow (Geng et al., 2025), which learns average velocities through a differential identity relating them to instantaneous velocities. By learning finite-time updates, these approaches replace many evaluations of an instantaneous velocity field with one or a few evaluations of a learned map. Their effectiveness consequently depends on how accurately this map can be learned.

We investigate whether changing a pretrained teacher's time parameterization can make its flow map easier to learn. Motivated by the hypothesis that segments with large normal acceleration are harder to distill, we construct a shared clock that allocates more student time to these regions while preserving the teacher's geometric paths and terminal distribution. The clock is estimated once from robust, regularized normal-acceleration statistics. We then train the student against the reparameterized velocity $\widetilde{b}_\tau(x) = \phi'(\tau) b_{\phi(\tau)}(x)$ using Lagrangian flow-map distillation. This requires neither teacher retraining nor changes to the student architecture or additional inference-time network evaluations.

Our main contributions are:

- **A normal-acceleration-based clock.** We derive a shared time transformation that equalizes a population normal-acceleration statistic under suitable assumptions, and estimate it robustly from teacher trajectories.
- **Geometry-aware flow-map distillation.** We incorporate this time coordinate into Lagrangian flow-map distillation, training a single student directly against the reparameterized teacher dynamics. The time map is estimated once before distillation and requires neither teacher retraining nor changes to the student architecture, with no additional inference-time network evaluations.
- **Improved few-step generation.** Through synthetic and image-generation experiments, we demonstrate improved sample quality over identity-time distillation at matched inference budgets. In particular, we observe meaningful improvements to one-step generation, where no intermediate sampling times can be adjusted, highlighting the role of time reparameterization in the distillation problem itself.

2 Related work

Flow-map learning. Progressive distillation (Salimans & Ho, 2022) compresses successive teacher updates, while consistency models and consistency trajectory models learn endpoint predictions or transitions between intermediate noise levels (Song et al., 2023; Kim et al., 2024). Flow map matching (Boffi et al., 2025b) provides a mathematical framework for these constructions, with objectives for both teacher distillation and direct training. Subsequent work develops self-distillation formulations (Boffi et al., 2025a) and practical continuous-time distillation objectives (Sabour et al., 2025). Shortcut Models and MeanFlow offer complementary formulations based on compositional consistency and average velocities (Frans et al., 2025; Geng et al., 2025). Recent extensions include parallel prediction of interval velocities (Shaul et al., 2026) and distribution-based distillation using average-velocity information (Zhao et al., 2026). Our work builds on Lagrangian flow-map distillation and investigates the time coordinate in which the teacher's finite-time transport is learned.

Trajectory geometry and coupling. Several approaches improve generative dynamics by simplifying their trajectories. Rectified flow and reflow encourage straighter transport (Liu et al., 2023), while minibatch optimal transport modifies source–target couplings to obtain simpler dynamics (Tong et al., 2024). Lee et al. (2023) learn couplings that reduce trajectory curvature and also investigate the benefits of straighter teachers for distillation. Isokinetic Flow Matching introduces a lookahead velocity-consistency regularizer motivated by reducing material acceleration (Khan, 2026). These methods modify the learned velocity field or its training coupling. Our transformation instead retains the geometric trajectories of a pretrained teacher and adjusts their traversal speed before flow-map distillation.

Sampling schedules and trajectory diagnostics. Sampling efficiency also depends on the placement of numerical integration steps. Align Your Steps optimizes schedules through a stochastic-calculus-based discrepancy criterion (Sabour et al., 2024), while Geometry-Inspired Time Scheduling uses trajectory information and dynamic programming to select integration times (Chen et al., 2024). Entropic Time Schedulers allocate sampling points according to conditional-entropy progress

(Stančević et al., 2025). Particularly close to our construction, Sharpen Your Flow estimates the full material-acceleration magnitude of a pretrained flow and derives a square-root sampling density implemented through inverse-CDF scheduling (Gupta et al., 2026). Our method uses the normal component of acceleration and applies the resulting clock to the teacher dynamics during student training. This distinction is particularly relevant to one-step flow-map generation, where no intermediate inference times can be adjusted.

Adaptive allocation of training effort. Temporal difficulty has also motivated adaptive training procedures. Curvature-Adaptive Consistency Flow Matching uses consistency-error feedback and reinforcement learning to allocate training across temporal stages (Tian et al., 2026). Its boundary-focused allocation provides qualitative evidence of nonuniform distillation difficulty, although its diagnostic differs from normal acceleration. Complexity-Balanced Diffusion Splitting uses acceleration or spatial regularity estimates to partition time among specialized networks (Issachar et al., 2026). Our approach trains a single student using a fixed time transformation estimated from the pretrained teacher.

Time reparameterization and learnability. Time parameterization can affect learning as well as numerical integration. Tsimpos et al. (2025) optimize the schedule of a fixed transport interpolation to control the spatial Lipschitz constant of its velocity field. Chen et al. (2025) develop Lipschitz-guided interpolation schedules and transfer formulas for applying them to pretrained models. Other approaches prescribe conditional velocity profiles (Bondar, 2026) or construct an inverse-CDF interpolation schedule from regression losses measured during a pilot training run (Tania & Khan, 2026). Time transformations also appear in Flow Map Language Models, where a decoding-error-based clock guides temporal sampling and flow-map learning (Lee et al., 2026). These works establish the relevance of temporal coordinates to generative learning. We investigate a shared clock derived from the normal acceleration of a frozen teacher, and train the student directly against the reparameterized teacher velocity in the Lagrangian distillation objective.

3 Background

In this section, we first formulate the flow matching problem and then introduce flow-map distillation methods.

3.1 Flow Matching

Let $(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1) \sim \pi \in \Pi(\rho_0, \rho_1)$ denote a pair of samples from a coupling of the source and target distributions. Following the stochastic-interpolant formulation (Albergo & Vanden-Eijnden, 2023), we define intermediate states through a time-differentiable interpolant:

$$I_t(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1) := \alpha_t \check{\mathbf{x}}_0 + \beta_t \check{\mathbf{x}}_1,$$

with boundary conditions

$$I_0(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1) = \check{\mathbf{x}}_0, \qquad I_1(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1) = \check{\mathbf{x}}_1.$$

For brevity, we write $I_t := I_t(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1)$, where $(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1) \sim \pi$.

Flow matching (Lipman et al., 2023) learns a time-dependent velocity field by regressing the interpolant velocity:

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) = \mathbb{E}_{t,(\check{\mathbf{x}}_0,\check{\mathbf{x}}_1)\sim\pi}\left[\|b_t^{\boldsymbol{\theta}}(I_t) - \dot{I}_t\|^2\right],$$

where $t \sim \mathcal{U}([0,1])$ independently of the endpoint pair and $\dot{I}_t = \frac{\mathrm{d}}{\mathrm{d}t} I_t$. This loss is minimized by

$$b_t^\star(\boldsymbol{x}) = \mathbb{E}\left[\dot{I}_t \mid I_t = \boldsymbol{x}\right].$$

Under suitable regularity assumptions, the ODE driven by $b_t^\star$ and initialized from ρ_0 reproduces the interpolant marginals. In all our experiments, we use linear interpolation,

$$I_t(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1) = (1-t)\check{\mathbf{x}}_0 + t\check{\mathbf{x}}_1, \qquad \dot{I}_t(\check{\mathbf{x}}_0, \check{\mathbf{x}}_1) = \check{\mathbf{x}}_1 - \check{\mathbf{x}}_0.$$

We denote the trained teacher field by b_t and reserve $\mathbf{x}_t$ for its ODE trajectories:

$$\dot{\mathbf{x}}_t = b_t(\mathbf{x}_t), \qquad \mathbf{x}_0 \sim \rho_0.$$

These trajectories generally differ from individual interpolation paths. Generation requires numerical ODE integration, typically involving many network evaluations and motivating few-step samplers and distillation methods.

3.2 Flow-Map Distillation

Consistency models (Song et al., 2023) learn to map states along a probability-flow trajectory to a common data endpoint. Flow-map distillation (Boffi et al., 2025b) learns two-time maps connecting intermediate states of a pretrained teacher ODE. Let $X_{s,t}$ denote its solution map from time s to time t:

$$X_{s,t}(\mathbf{x}_s) = \mathbf{x}_t. \tag{1}$$

Approximating this map allows a student to replace many evaluations of the teacher velocity b_t with a single finite-time update.

Following Boffi et al. (2025b), we parameterize the student as

$$\widehat{X}^{\boldsymbol{\theta}}_{s,t}(\boldsymbol{x}) = \boldsymbol{x} + (t-s)v_{\boldsymbol{\theta}}(\boldsymbol{x}, s, t), \tag{2}$$

where $v_{\boldsymbol{\theta}}$ approximates the average velocity over the interval $[s,t]$. This parameterization enforces the identity condition $\widehat{X}^{\boldsymbol{\theta}}_{s,s}(\boldsymbol{x}) = \boldsymbol{x}$.

Assuming the teacher ODE has unique solutions, the flow-map relation equation 1 is equivalently characterized by the Lagrangian equation

$$\partial_t X_{s,t}(\boldsymbol{x}) = b_t\big(X_{s,t}(\boldsymbol{x})\big), \qquad X_{s,s}(\boldsymbol{x}) = \boldsymbol{x}.$$

Since our parameterization enforces the identity condition, Lagrangian flow-map distillation (LMD) (Boffi et al., 2025b) trains the student by minimizing the residual of this equation. We use the following stop-gradient formulation:

$$\mathcal{L}_{\mathrm{LMD}}(\boldsymbol{\theta}) = \mathbb{E}_{s,t,(\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1)\sim\pi}\left[\left\|\partial_t \widehat{X}^{\boldsymbol{\theta}}_{s,t}(I_s) - \mathrm{sg}\Big(b_t\Big(\widehat{X}^{\boldsymbol{\theta}}_{s,t}(I_s)\Big)\Big)\right\|^2\right]. \tag{3}$$

The time-pair sampling distribution is specified in the experimental setup.

Following the stop-gradient treatment of Lagrangian distillation discussed by Boffi et al. (2025a), we apply $\mathrm{sg}(\cdot)$ to the teacher velocity evaluated at the student output. This avoids backpropagation through the teacher's spatial Jacobian, while the teacher parameters remain fixed throughout training. We also found this treatment important for training stability in our experiments; further details are provided in the appendix.

Sampling from the flow map Given a partition $0 = t_0 < \cdots < t_K = 1$, we generate samples by composing the learned maps:

$$\widehat{\mathbf{x}}_{t_{k+1}} = \widehat{X}^{\boldsymbol{\theta}}_{t_k,t_{k+1}}(\widehat{\mathbf{x}}_{t_k}), \qquad k = 0,\dots,K-1, \qquad \widehat{\mathbf{x}}_0 \sim \rho_0.$$

The terminal state $\widehat{\mathbf{x}}_1$ is the generated sample. Each update requires one student-network evaluation, allowing the sampling budget K to be chosen after training. In particular, one-step generation applies $\widehat{X}^{\boldsymbol{\theta}}_{0,1}$ directly to a source sample.

4 Proposed Method

We first motivate normal acceleration as a proxy for flow-map learning difficulty. We then construct a time transformation that redistributes this quantity across time and incorporate the resulting clock into Lagrangian distillation.

4.1 Normal Acceleration As A Proxy For Distillation Difficulty

Our working hypothesis is that trajectory segments with large normal acceleration are harder to represent through finite-time flow maps. To motivate this, consider the exact average velocity along a teacher trajectory:

$$\frac{\mathbf{x}_t - \mathbf{x}_s}{t-s} = \frac{1}{t-s}\int_s^t b_u(\mathbf{x}_u)\,\mathrm{d}u, \qquad s < t.$$

If the velocity is constant on $[s,t]$, this average equals $b_s(\mathbf{x}_s)$. More generally, if its direction is fixed, the average remains aligned with that direction, and only its magnitude varies with the interval. When the direction changes, both the magnitude and direction of the average can depend on the two endpoints. This motivates using normal acceleration, the component associated with directional changes, as a diagnostic of temporal learning difficulty. The resulting connection to distillation error is an empirical hypothesis, rather than a consequence of the flow-map identity.

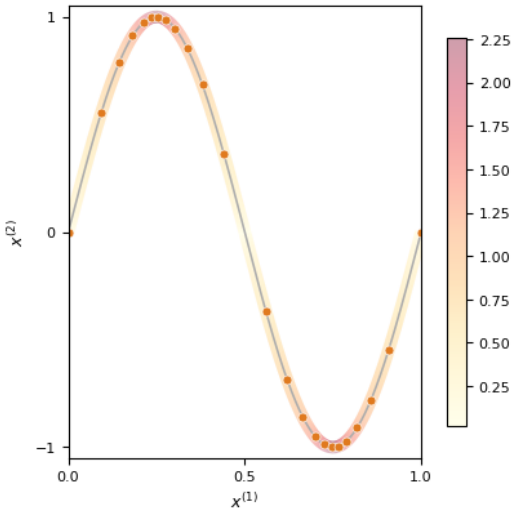

Figure 1: Time warping represented on the 2d trajectory $\boldsymbol{x}_t = (t, \sin(2\pi t))$. The orange dots represent the reparametrized trajectory $\boldsymbol{x}_{\phi(\tau_K)}$ with uniform time steps $\tau_k = k/(N+1)$. The colorscale represents the density $q = \psi\prime$ of ϕ(t) with t $\sim \mathcal{U}[0,1]$.

Normal-acceleration diagnostic. For the teacher field b_t, define the material acceleration

$$a_t(\boldsymbol{x}) := \partial_t b_t(\boldsymbol{x}) + D_{\boldsymbol{x}} b_t(\boldsymbol{x})\, b_t(\boldsymbol{x}).$$

Along a teacher trajectory, $\ddot{\mathbf{x}}_t = a_t(\mathbf{x}_t)$. Where $b_t(\boldsymbol{x}) \neq 0$, let $\widehat{b}_t(\boldsymbol{x}) = b_t(\boldsymbol{x})/\|b_t(\boldsymbol{x})\|$ and decompose

$$a_t^{\parallel}(\boldsymbol{x}) = \big(\widehat{b}_t(\boldsymbol{x})^\top a_t(\boldsymbol{x})\big)\widehat{b}_t(\boldsymbol{x}), \qquad a_t^{\perp}(\boldsymbol{x}) = a_t(\boldsymbol{x}) - a_t^{\parallel}(\boldsymbol{x}).$$

The tangential component changes speed, while the normal component changes velocity direction. We define

$$n_t(\boldsymbol{x}) := \|a_t^{\perp}(\boldsymbol{x})\|.$$

Geometrically, n_t measures how strongly a trajectory bends at its current traversal speed: it equals the geometric curvature multiplied by the squared speed. Thus, slowing down reduces normal acceleration without altering the path taken by a trajectory. To summarize this diagnostic across teacher trajectories, we define the *hardness profile*

$$h_p(t) := \left(\mathbb{E}\left[n_t(\mathbf{x}_t)^p\right]\right)^{1/p}, \qquad p > 0,$$

where the expectation is over teacher trajectories initialized from $\mathbf{x}_0 \sim \rho_0$. We use this profile as a proxy for time-dependent distillation difficulty.

4.2 Geometry-Aware Time Reparameterization

The hardness profile identifies where additional temporal resolution may be useful. We therefore seek a shared change of time that slows traversal through regions of high hardness and accelerates it where hardness is low. Such a transformation preserves the teacher's geometric paths and terminal distribution, while changing its normal acceleration. Our goal is to choose the new time coordinate so that the resulting hardness profile is constant. The following theorem gives an explicit construction.

Theorem 4.1 (Equalizing the hardness profile)**.** *Assume the teacher trajectories are C^2 with nonzero velocity, and h_p is positive and continuously differentiable. Define*

$$\psi(t) := \frac{\int_0^t \sqrt{h_p(u)}\,\mathrm{d}u}{\int_0^1 \sqrt{h_p(u)}\,\mathrm{d}u}, \qquad \phi := \psi^{-1}.$$

Then the reparameterized trajectories $\mathbf{y}_\tau := \mathbf{x}_{\phi(\tau)}$ *have constant hardness:*

$$\widetilde{h}_p(\tau) := (\mathbb{E}[\widetilde{n}_\tau(\mathbf{y}_\tau)^p])^{1/p} = \left(\int_0^1 \sqrt{h_p(u)}\,\mathrm{d}u\right)^2, \qquad \tau \in [0,1],$$

where $\widetilde{n}_\tau$ *denotes the normal-acceleration magnitude of the reparameterized dynamics.*

Proof. See Appendix A.1 □

Remark 4.2 (Tempering the time transformation). The strength of the transformation can be controlled by choosing $\psi'(t) \propto (h_p(t) + \xi)^{\delta/2}$, where $\delta \geq 0$ and $\xi \geq 0$ is an optional numerical floor. The choice $\delta = 0$ recovers the identity clock, $0 < \delta < 1$ softens the transformation, and $\delta > 1$ increases its concentration on high-hardness regions. The theorem corresponds to $\delta = 1$ and $\xi = 0$.

Practical construction. We estimate the time map once before distillation using M teacher trajectories evaluated on a fine grid $t_k = k/N$, $k = 0, \ldots, N$, with $N \geq 100$. We estimate material acceleration by finite differences of the teacher velocities along each trajectory, then project onto the normal direction to obtain $n_{k,m}$. At each time, we discard the largest 5% of these values before computing the hardness profile. This limits the influence of rare sharp turns, which can produce particularly large normal accelerations (phenomenon which we mainly observed in 2d experiments). We integrate the resulting clock density numerically and fit a monotonicity-preserving PCHIP interpolant to the inverse pairs (τ_k, t_k) (Fritsch & Butland, 1984). The resulting C^1 map has an analytically evaluable derivative. A positive floor ξ ensures a strictly increasing cumulative clock; in our experiments, the estimated hardness already remained bounded away from zero. Trimming, tempering, and numerical approximation mean that the practical clock need not exactly equalize the population profile. Algorithm 1 summarizes the procedure.

Algorithm 1 Offline construction of the shared time map

Require: Teacher b_t, source ρ_0, trajectory count M, grid size N, aggregation $p > 0$, tempering $\delta \geq 0$, floor $\xi \geq 0$

1: Generate M teacher trajectories-velocity pairs $(\mathbf{x}_{k,m}, \mathbf{b}_{k,m})$ from $\mathbf{x}_{0,m} \sim \rho_0$ on $t_k = k/N$, $k = 0, \ldots, N$, $m = 1, \ldots, M$
2: $\mathbf{a}_{k,m} \leftarrow \mathrm{FD}_t[\mathbf{b}_{\cdot,m}]_k$
3: $\hat{\mathbf{b}}_{k,m} \leftarrow \mathbf{b}_{k,m}/\|\mathbf{b}_{k,m}\|$; $n_{k,m} \leftarrow \|\mathbf{a}_{k,m} - (\hat{\mathbf{b}}_{k,m}^\top \mathbf{a}_{k,m})\hat{\mathbf{b}}_{k,m}\|$
4: **for** $k = 0, \ldots, N$ **do**
5: $\quad \mathcal{I}_k \leftarrow$ indices of the smallest $\lceil 0.95M \rceil$ values of $\{n_{k,m}\}_{m=1}^M$
6: $\quad \widehat{h}_{p,k} \leftarrow \left(\frac{1}{|\mathcal{I}_k|}\sum_{m\in\mathcal{I}_k} n_{k,m}^p\right)^{1/p}$; $w_k \leftarrow (\widehat{h}_{p,k} + \xi)^{\delta/2}$
7: **end for**
8: $S_0 \leftarrow 0$; $S_k \leftarrow S_{k-1} + \frac{t_k - t_{k-1}}{2}(w_{k-1} + w_k)$, $k = 1, \ldots, N$
9: $\tau_k \leftarrow S_k/S_N$, $k = 0, \ldots, N$
10: $\phi \leftarrow \mathrm{PCHIP}\big(\{(\tau_k, t_k)\}_{k=0}^N\big)$
11: **return** ϕ and its derivative ϕ'

Flow-map distillation in warped time Given the clock $\phi = \psi^{-1}$ (Theorem 4.1), we train a student to approximate the reparameterized flow map

$$Y_{\alpha,\beta} := X_{\phi(\alpha),\phi(\beta)}, \qquad 0 \leq \alpha \leq \beta \leq 1.$$

The trajectories $\mathbf{y}_\tau = \mathbf{x}_{\phi(\tau)}$ satisfy

$$\frac{\mathrm{d}\mathbf{y}_\tau}{\mathrm{d}\tau} = \widetilde{b}_\tau(\mathbf{y}_\tau), \qquad \widetilde{b}_\tau(\boldsymbol{x}) := \phi'(\tau) b_{\phi(\tau)}(\boldsymbol{x}).$$

For image experiments, we condition the student on the reparameterized times while retaining the physical-time interval as its displacement scale:

$$\widehat{Y}^{\boldsymbol{\theta}}_{\alpha,\beta}(\boldsymbol{x}) = \boldsymbol{x} + \big(\phi(\beta) - \phi(\alpha)\big) v_{\boldsymbol{\theta}}(\boldsymbol{x}, \alpha, \beta). \tag{4}$$

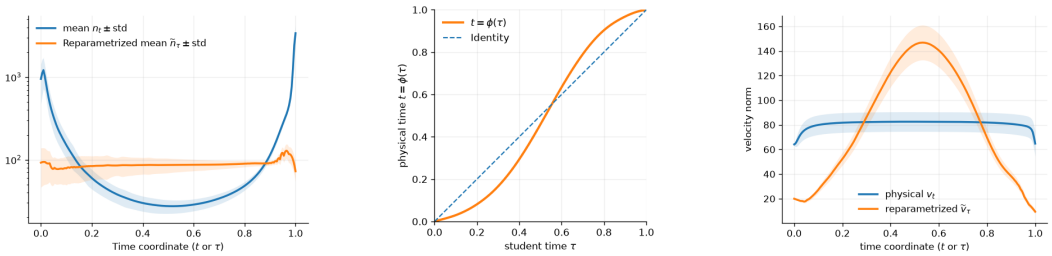

Figure 2: CelebA64 flow matching model (teacher) measurements plotted against their respective coordinates. **Left**: estimated hardness before and after reparametrization. **Middle**: estimated map ϕ. **Right**: original and reparameterized velocity norms.

This choice separates temporal allocation from velocity scaling. Empirically, the original teacher's velocity norm is approximately constant over most of the time interval, whereas reparameterization introduces substantial variation through the factor ϕ' (see Figure 2). The parameterization also enforces $\widehat{Y}^{\boldsymbol{\theta}}_{\alpha,\alpha}(\boldsymbol{x}) = \boldsymbol{x}$. For the toy experiments, we instead use displacement scale $\beta - \alpha$; the warped-time distillation objective below applies to either parameterization. Both scalings coincide under the identity clock.

We apply the Lagrangian distillation objective to the reparameterized teacher:

$$\mathcal{L}^{\phi}_{\mathrm{LMD}}(\boldsymbol{\theta}) = \mathbb{E}_{\alpha,\beta,(\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1)\sim\pi}\left[\left\|\partial_\beta \widehat{Y}^{\boldsymbol{\theta}}_{\alpha,\beta}(I_{\phi(\alpha)}) - \mathrm{sg}\Big(\widetilde{b}_\beta\Big(\widehat{Y}^{\boldsymbol{\theta}}_{\alpha,\beta}(I_{\phi(\alpha)})\Big)\Big)\right\|^2\right]. \tag{5}$$

Training inputs are sampled from the interpolant at physical time $\phi(\alpha)$. Both the teacher and the clock remain fixed during training. For image experiments, we additionally apply learned loss weighting, as detailed in Appendix C.

We sample analogously by composing $\widehat{Y}^{\boldsymbol{\theta}}_{\tau_k,\tau_{k+1}}$ over a student-time partition $0 = \tau_0 < \cdots < \tau_K = 1$, starting from $\widehat{\mathbf{y}}_0 \sim \rho_0$. The overhead that comes from computing the scaling factor $(\phi(\tau_{k+1}) - \phi(\tau_k))$ is negligible compared with neural-network evaluation, and these differences can be precomputed once for a fixed partition.

5 Experiments

We first examine the temporal distribution of normal acceleration and the effect of our clock on teacher sampling. We then evaluate whether training in the resulting time coordinate improves flow-map distillation on synthetic and image datasets.

5.1 Temporal Geometry and Teacher Sampling

Normal-acceleration profiles. We evaluate normal-acceleration profiles along trajectories of the CIFAR-10 and CelebA64 teachers. Both exhibit a similar U-shaped temporal profile, with larger normal acceleration near the endpoints and smaller values in the middle. Figure 2 shows CelebA64 as a representative example; the CIFAR-10 profiles exhibit the same qualitative behavior, with different magnitudes. This concentration motivates allocating more student time to the boundary regions. It is also qualitatively consistent with the U-shaped distillation difficulty reported by Tian et al. (2026), who learn an adaptive time-sampling curriculum using reinforcement learning. This agreement motivates our diagnostic, although it does not establish a direct correspondence between normal acceleration and distillation error.

Figure 2 illustrates the resulting transformation for CelebA64. The estimated clock makes the normal-acceleration profile more uniform across time. Meanwhile, the original teacher's velocity norm is approximately constant over most of the interval, whereas the warped velocity exhibits greater variation. This observation motivates the physical-time displacement scaling used for image distillation.

Teacher sampling. Before evaluating distillation, we test whether the estimated clock improves numerical sampling from the frozen CIFAR-10 teacher. For fixed-step solvers with K integration steps, we integrate the original field b_t using either uniform nodes $t_k = k/K$ or clock-adapted nodes $t_k = \phi(k/K)$, for $k = 0, \dots, K$. For the adaptive Dopri5 solver, we instead compare integration of b_t with integration of the reparameterized field $\widetilde{b}_\tau = \phi'(\tau) b_{\phi(\tau)}$. We report FID against the number of velocity-network evaluations (NFE), which dominate sampling cost.

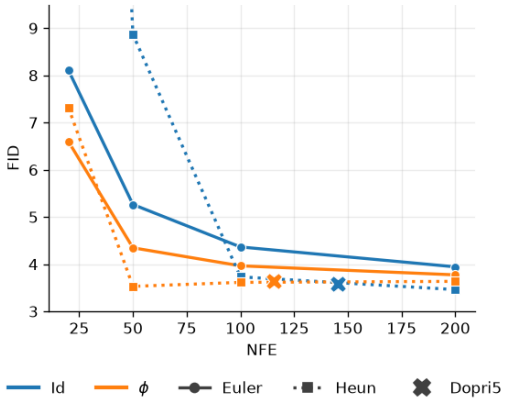

Figure 3: CIFAR-10 Flow Matching model (teacher) evaluated FID using different solvers and number of steps.

Figure 3 shows that clock-adapted grids improve sample quality at low evaluation budgets. With 25 Heun steps, FID decreases from 8.8 on the uniform grid to 3.5 on the clock-adapted grid, approaching the teacher's performance at larger budgets. Naturally, the advantage diminishes as the evaluation budget increases and as we get closer to the real ODE's solutions.

These results support the usefulness of the estimated clock for allocating a limited fixed-step integration budget. The following experiments test whether training in the corresponding time coordinate also improves flow-map distillation.

5.2 Flow-Map Distillation

We compare distillation under the identity clock, $\mathrm{Id}(t) = t$, with distillation under our estimated clock ϕ. For each dataset, we use a fixed flow-matching teacher and train two students with matched architectures, training budgets, and optimization settings. The Id student learns from b, while the ϕ student learns from its reparameterized counterpart $\widetilde{b}$. Before distillation, we estimate ϕ from 256 teacher trajectories generated with an adaptive ODE solver. The clock remains fixed throughout training. We use student-time displacement scaling for toy data and physical-time displacement scaling for images, as described in the method section. All comparisons use matched numbers of student-network evaluations. Implementation details and additional results are provided in the appendix.

5.2.1 Toy Experiments

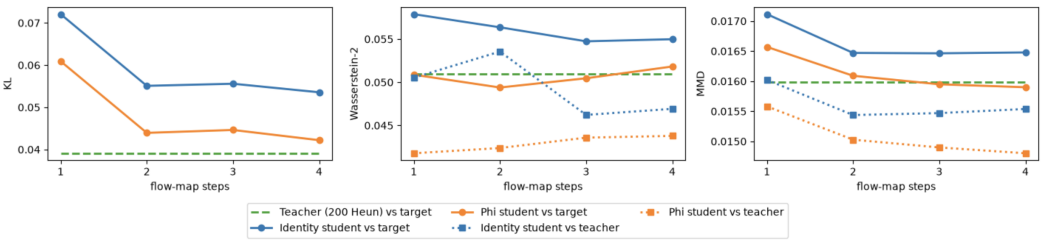

Figure 4: Quantitative results of the flow matching teacher and its distilled flow maps trained on the 2D checkerboard distribution.

We evaluate on a two-dimensional checkerboard distribution, whose sharp boundaries and disconnected support provide a test of sample fidelity. The teacher and students use four-layer MLPs of width 512, trained for 50,000 updates with a batch size of 8192. The source distribution is $\rho_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Figure 7 shows that ϕ produces sharper boundaries and fewer samples outside the target support than Id. We quantify sample quality using a histogram-based forward KL statistic, Gaussian-kernel

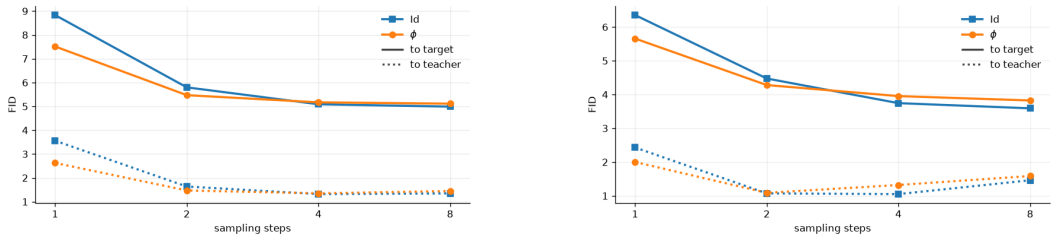

Figure 5: **Left**: CIFAR-10 Flow Map FID after 300k training steps. **Right**: CelebA64 Flow Map FID after 350k training steps.

MMD with bandwidth $\sigma = 0.05$, and empirical Wasserstein-2 distance. The KL computation exploits the known target density; its discretization and normalization are specified in Appendix B. MMD and Wasserstein-2 are evaluated against both the target distribution and teacher-generated samples, measuring target fidelity and distillation fidelity, respectively. Across the evaluated budgets of 1, 2, 3, and 4 network evaluations, ϕ improves all three reported metrics relative to Id (Figure 4).

5.2.2 Image Experiments

We evaluate unconditional generation on CIFAR-10 (Krizhevsky et al., 2009) and CelebA64 (Liu et al., 2015). Both experiments use RAdam (Liu et al., 2020) and learned loss weighting following EDM2 (Karras et al., 2024) and its adaptation to flow maps (Boffi et al., 2025a). We report FID here and provide full FID/KID tables, including additional checkpoints, in Appendix C.

CIFAR-10. We use approximately 35M-parameter U-Nets and a pretrained TorchCFM teacher (Tong et al., 2023; 2024). Students are trained for 400,000 updates with a global batch size of 196.

CelebA64. We use approximately 124M-parameter EDM2 U-Nets. The teacher is trained on the official CelebA training partition for 350,000 updates, achieving FID 2.36 with 100 Heun integration steps. Students are trained for 400,000 updates with a global batch size of 256.

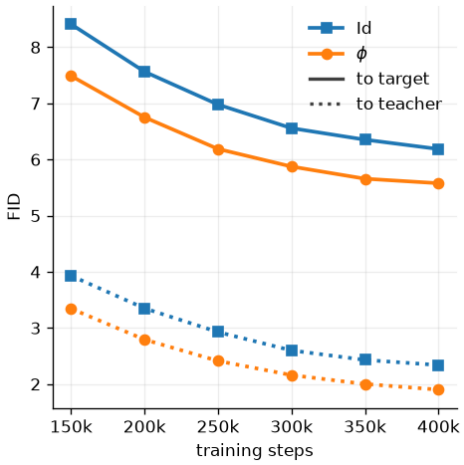

Figure 6: CelebA64 1 step FID vs training budget.

Results. Figure 5 compares Id and ϕ at matched inference budgets. On both datasets, ϕ improves one- and two-step FID, whereas Id performs better at the evaluated budgets at four and eight steps. The one-step comparison is particularly informative: both clocks have identical endpoints and displacement scales, so improvements reflect the learned map rather than the placement of intermediate sampling times. Moreover Figure 6 shows that the 1 step improvement is consistent across training steps for the CelebA64 model (CIFAR-10 shows a very similar trend).

6 Conclusion

We proposed a geometry-aware time reparameterization that redistributes normal acceleration across time to improve flow-map distillation. The clock is estimated once, preserves teacher trajectories, and requires neither teacher retraining nor additional inference-time network evaluations. Experiments show improved few-step generation, in particular one-step gains that highlight the role of time coordinates in learning the map itself. Establishing a theoretical link between normal acceleration and distillation difficulty remains a direction for future work.

AI USE STATEMENT

We used ChatGPT and AI coding assistants to support implementation and debugging, literature searches, manuscript drafting and revision, and figure preparation. We also used these tools to discuss and refine mathematical arguments and experimental methodology. The authors take responsibility for the final manuscript, code, references, and reported findings.

The authors reviewed and revised AI-assisted text, checked mathematical arguments and cited sources, and tested AI-assisted code against the stated methods.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=li7qeBbCR1t.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021. URL https://arxiv.org/abs/1801.01401.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation, 2025a. URL https://arxiv.org/abs/2505.18825.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025b. URL https://openreview.net/forum?id=cqDH0e6ak2.

Vitalii Bondar. Velocity scheduled flow matching, 2026. URL https://arxiv.org/abs/2607.11442.

Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the trajectory regularity of ode-based diffusion sampling, 2024. URL https://arxiv.org/abs/2405.11326.

Yifan Chen, Eric Vanden-Eijnden, and Jiawei Xu. Lipschitz-guided design of interpolation schedules in generative models, 2025. URL https://arxiv.org/abs/2509.01629.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

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

F. N. Fritsch and J. Butland. A method for constructing local monotone piecewise cubic interpolants. *SIAM Journal on Scientific and Statistical Computing*, 5(2):300–304, 1984. doi: 10.1137/0905021. URL https://doi.org/10.1137/0905021.

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*, volume 38, 2025. URL https://openreview.net/forum?id=uWj4s7rMnR.

Aditi Gupta, Soon Hoe Lim, Annan Yu, and N. Benjamin Erichson. Sharpen your flow: Sharpness-aware sampling for flow matching, 2026. URL https://arxiv.org/abs/2605.11547.

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, 2018. URL https://arxiv.org/abs/1706.08500.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Noam Issachar, Dani Lischinski, and Raanan Fattal. Complexity-balanced diffusion splitting, 2026. URL https://arxiv.org/abs/2606.06477.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models, 2024. URL https://arxiv.org/abs/2312.02696.

Tauhid Khan. Isokinetic flow matching for pathwise straightening of generative flows, 2026. URL https://arxiv.org/abs/2604.04491.

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. URL https://openreview.net/forum?id=ymjI8feDTD.

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

Chanhyuk Lee, Jaehoon Yoo, Manan Agarwal, Sheel Shah, Jerry Huang, Aditi Raghunathan, Seunghoon Hong, Nicholas M. Boffi, and Jinwoo Kim. Flow map language models: One-step language modeling via continuous denoising, 2026. URL https://arxiv.org/abs/2602.16813.

Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ODE-based generative models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18957–18973. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lee23j.html.

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. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, 2020.

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

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

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, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/260a14acce2a89dad36adc8eefe7c59e-Abstract-Conference.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models, 2024. URL https://arxiv.org/abs/2404.14507.

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.14603.

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

Neta Shaul, Chao Liu, Arash Vahdat, and Julius Berner. Parallel decoding distillation for fast image and video generation, 2026. URL https://arxiv.org/abs/2607.26004.

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

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

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Dejan Stančević, Florian Handke, and Luca Ambrogioni. Entropic time schedulers for generative diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2504.13612.

Airin Akter Tania and Md Raihan Khan. Difficulty-calibrated interpolation paths for conditional flow matching, 2026. URL https://arxiv.org/abs/2608.21286.

Songtao Tian, Guhan Chen, Bohan Li, Jingyi Ma, and Zixiong Yu. Curvature-adaptive consistency flow matching: Autonomous trajectory optimization via reinforcement learning, 2026. URL https://arxiv.org.

Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. *arXiv preprint 2307.03672*, 2023.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Panos Tsimpos, Zhi Ren, Jakob Zech, and Youssef Marzouk. Optimal scheduling of dynamic transport. In *Proceedings of the Thirty-Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 5441–5505. PMLR, 2025. URL https://proceedings.mlr.press/v291/tsimpos25a.html.

An Zhao, Shengyuan Zhang, Zhongjian Sun, Yixiang Zhou, Zejian Li, Ling Yang, Tianrun Chen, and Lingyun Sun. Mean flow distillation: Robust and stable distillation for flow matching models, 2026. URL https://arxiv.org/abs/2606.11155.

A Mathematical Details and Proofs

A.1 Proof of the main Theorem and construction of ϕ

Proof of Theorem 4.1. Let $t \in [0,1]$ denote teacher time and $\tau \in [0,1]$ denote student time. Write

$$Z := \int_0^1 \sqrt{h_p(u)}\,\mathrm{d}u.$$

Since h_p is positive and continuously differentiable, $0 < Z < \infty$ and

$$\psi'(t) = \frac{\sqrt{h_p(t)}}{Z} > 0.$$

Thus ψ is a strictly increasing C^2 map with $\psi(0) = 0$ and $\psi(1) = 1$. Its inverse $\phi = \psi^{-1}$ is also C^2, with $\phi' > 0$, and satisfies

$$t = \phi(\tau), \qquad \tau = \psi(t).$$

Consider a teacher trajectory $\mathbf{x}$ and its reparameterized version

$$\mathbf{y}_\tau = \mathbf{x}_{\phi(\tau)}.$$

By the chain rule,

$$\dot{\mathbf{y}}_\tau = \phi'(\tau)\dot{\mathbf{x}}_{\phi(\tau)}, \qquad \ddot{\mathbf{y}}_\tau = \phi''(\tau)\dot{\mathbf{x}}_{\phi(\tau)} + \phi'(\tau)^2\ddot{\mathbf{x}}_{\phi(\tau)}.$$

Since $\phi'(\tau) > 0$, the two trajectories have the same unit tangent at corresponding times. The term containing ϕ'' is tangential, so

$$\widetilde{a}^{\perp}_\tau(\mathbf{y}_\tau) = \phi'(\tau)^2 a^{\perp}_{\phi(\tau)}(\mathbf{x}_{\phi(\tau)}), \qquad \widetilde{n}_\tau(\mathbf{y}_\tau) = \phi'(\tau)^2 n_{\phi(\tau)}(\mathbf{x}_{\phi(\tau)}).$$

The same deterministic clock applies to every trajectory, hence

$$\widetilde{h}_p(\tau) = \phi'(\tau)^2 h_p(\phi(\tau)).$$

Here both expectations are taken over the same initial distribution of teacher trajectories.

Finally, differentiating $\psi(\phi(\tau)) = \tau$ gives

$$\psi'(\phi(\tau))\phi'(\tau) = 1, \qquad \phi'(\tau) = \frac{Z}{\sqrt{h_p(\phi(\tau))}}.$$

Substituting this expression into the preceding identity, we obtain

$$\widetilde{h}_p(\tau) = \frac{Z^2}{h_p(\phi(\tau))} h_p(\phi(\tau)) = \left(\int_0^1 \sqrt{h_p(u)}\,\mathrm{d}u\right)^2.$$

which is independent of τ. □

B Toy Experiments: Implementation Details

Dataset and training. The target is a uniform distribution on the occupied squares of a two-dimensional checkerboard. The source distribution is $\rho_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

The teacher and students use four-layer MLPs of width 512 and SiLU activation function. Each model is trained for 50,000 optimizer updates with a batch size of 8192. The teacher is optimized using AdamW (Loshchilov & Hutter, 2019) with learning rate 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}, and $\epsilon = 10^{-8}$. Its learning rate follows a cosine schedule down to 10^{-6}, and its exponential moving average uses decay 0.998. The students use AdamW with learning rate 10^{-2}, the same Adam coefficients, weight decay, and numerical stabilization, followed by an exponential learning-rate schedule with multiplicative decay 0.9999 after each optimizer update. Their exponential moving average uses decay 0.999. The maximum gradient norms are 10 for the teacher and 1 for the students. EMA weights are used for validation and evaluation. We use a constant loss weight $\frac{1}{2}$. The ϕ student uses displacement scale $\beta - \alpha$.

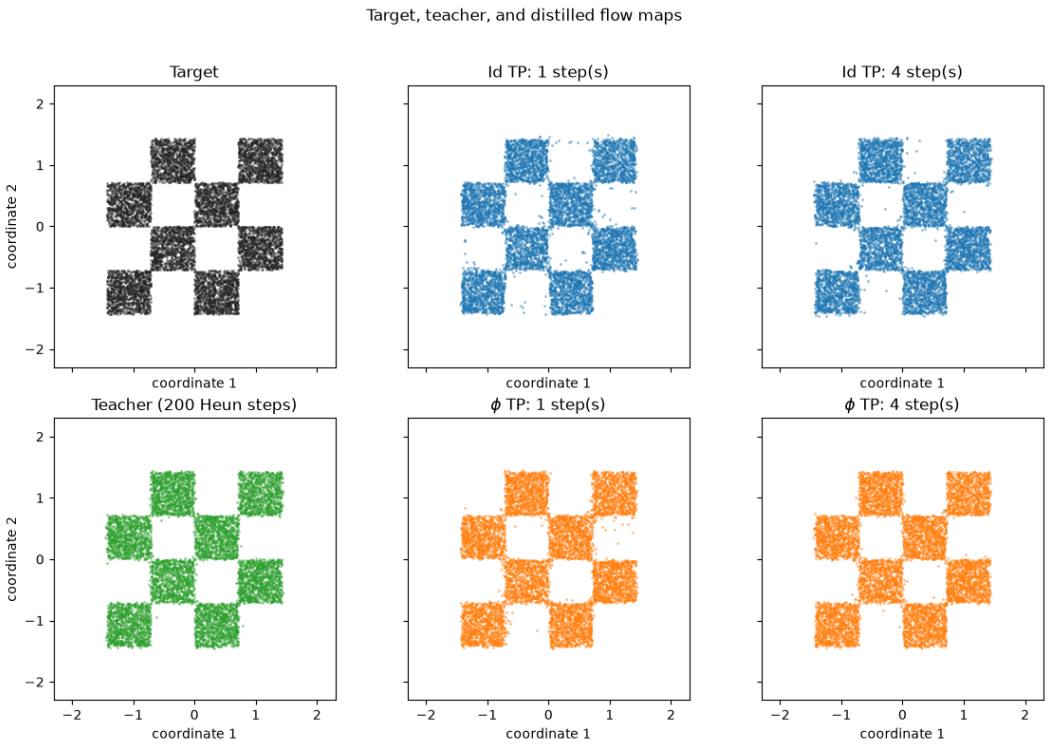

Figure 7: 8192 samples from the target distribution (checkerboard, top left, black), the flow matching teacher (bottom left, green), the Id-TP student (baseline, top middle and right) and the ϕ-TP student (ours, bottom middle and right).

Evaluation protocol. We evaluate generated samples using a histogram-based forward KL, Gaussian-kernel MMD, and empirical Wasserstein-2 distance. KL uses $64{,}000$ generated samples and the known target density. For MMD and Wasserstein-2, we use 8192 samples per distribution and compare against both target samples and teacher-generated samples, measuring target fidelity and distillation fidelity, respectively.

Histogram-based forward KL. We estimate the forward divergence $D_{\mathrm{KL}}(p\|q)$ using the known checkerboard density p and a histogram estimate of the generated density q. A fixed domain containing the target support is partitioned into 64×64 equal-area bins. For bin center c_{ij}, bin area ΔA, and generated count n_{ij}, we set

$$\widehat{q}(c_{ij}) = \frac{n_{ij}}{N \Delta A},$$

where N is the total number of generated samples, including those outside the histogram domain. We compute

$$\widehat{D}_{\mathrm{KL}}^{\varepsilon} = \sum_{i,j:\, p(c_{ij})>0} p(c_{ij}) \Delta A \log\left(\frac{p(c_{ij})}{\widehat{q}(c_{ij}) + \varepsilon}\right), \qquad \varepsilon = 10^{-10}.$$

Normalization by N retains the penalty associated with generated mass outside the domain. The density floor prevents infinite values from empty target-positive bins. This statistic approximates the forward KL at the chosen spatial resolution; its value depends on binning, sample count, and regularization. We use identical evaluation settings for all models.

Maximum mean discrepancy. We use the Gaussian kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}\right), \qquad \sigma = 0.05.$$

For distributions μ and ν, the squared population discrepancy is

$$\mathrm{MMD}_k^2(\mu,\nu) = \mathbb{E}[k(X,X')] + \mathbb{E}[k(Y,Y')] - 2\mathbb{E}[k(X,Y)],$$

where $X, X' \sim \mu$ and $Y, Y' \sim \nu$ are independent. We report the biased estimate

$$\widehat{\mathrm{MMD}}_{\mathrm{b}} = \left(\frac{1}{n^2}\sum_{i,j} k(x_i,x_j) + \frac{1}{m^2}\sum_{i,j} k(y_i,y_j) - \frac{2}{nm}\sum_{i,j} k(x_i,y_j). \right)^{1/2}.$$

Empirical Wasserstein distance. We compare empirical sample measures using the quadratic transport cost:

$$W_2^2(\widehat{\mu},\widehat{\nu}) = \min_{\gamma\in\Pi(\widehat{\mu},\widehat{\nu})} \int \|\boldsymbol{x}-\boldsymbol{y}\|^2 \,\mathrm{d}\gamma(\boldsymbol{x},\boldsymbol{y}).$$

For two equally sized point clouds with n samples, we solve the assignment problem with cost matrix $C_{ij} = \|x_i - y_j\|^2$ using SciPy's `linear_sum_assignment`. If σ is the resulting optimal permutation, we report

$$W_2(\widehat{\mu},\widehat{\nu}) = \left(\frac{1}{n}\sum_{i=1}^{n} \|x_i - y_{\sigma(i)}\|^2 \right)^{1/2}.$$

C Image Experiments: Implementation and Additional Results

C.1 Shared distillation setup

Training inputs and time sampling. Both image experiments use unconditional students trained with the stop-gradient Lagrangian distillation objective. The teacher and, where applicable, the estimated clock remain fixed throughout distillation. Input states are sampled directly from the linear interpolant: I_α for Id and $I_{\phi(\alpha)}$ for ϕ.

For off-diagonal examples, we draw $u, v \sim \mathcal{U}([0,1])$ independently and set

$$\alpha = \min(u,v), \qquad \beta = \max(u,v).$$

This gives a uniform distribution over the upper triangle. We found that sampling no time steps on the diagonal improved training results.

Learned loss weighting. We use the learned weighting introduced for diffusion training in EDM2 (Karras et al., 2024) and adapted to two-time flow-map objectives by Boffi et al. (2025a). Let $\ell_{\alpha,\beta}$ denote the unweighted per-example squared distillation residual. A scalar head with parameters η predicts

$$w_{\alpha,\beta} := w_\eta(\alpha, \beta-\alpha).$$

We jointly optimize the student and weighting head using

$$\mathcal{L}_{\mathrm{weighted}}(\boldsymbol{\theta},\eta) = \mathbb{E}\left[e^{-w_{\alpha,\beta}}\ell_{\alpha,\beta} + w_{\alpha,\beta}\right]. \tag{6}$$

The exponential factor rescales the contribution of each time pair, while the additive term penalizes arbitrarily suppressing the residual through large positive weights. We found this weighting important for training stability. Each student learns its own weighting function, using the same weighting scheme for Id and ϕ. The weighting head is used only during training and adds no inference-time network evaluations.

Hyperparameter selection. For each image dataset, we compared candidate learning rates and decay schedules using FID and KID computed from 50,000 generated samples at multiple training checkpoints and sampling budgets. Both metrics consistently selected the same learning-rate and decay configuration across these checkpoints and budgets. The selected configuration was also the same for Id and ϕ.

Training Steps	TP	FID to target					FID to teacher				
		1	2	4	8	20	1	2	4	8	20
150k	Id	10.09	6.31	**5.54**	**5.49**	**5.65**	4.44	1.83	**1.44**	**1.52**	**1.75**
150k	ϕ	**8.59**	**5.90**	5.63	5.71	5.73	**3.24**	**1.59**	1.47	1.68	1.89
300k	Id	8.83	5.80	**5.09**	**4.99**	**5.08**	3.56	1.64	**1.32**	**1.35**	**1.48**
300k	ϕ	**7.51**	**5.47**	5.17	5.11	5.14	**2.63**	**1.47**	1.35	1.45	1.59
400k	Id	8.57	5.71	**4.98**	**4.89**	5.02	3.40	1.62	**1.30**	**1.31**	**1.45**
400k	ϕ	**7.44**	**5.31**	5.07	5.06	**4.99**	**2.59**	**1.41**	1.33	1.41	1.53

Training Steps	TP	KID to target ($\times 10^3$)					KID to teacher ($\times 10^3$)				
		1	2	4	8	20	1	2	4	8	20
150k	Id	6.07	3.37	2.91	**3.04**	**3.24**	2.43	0.61	0.33	**0.45**	**0.68**
150k	ϕ	**4.97**	**2.85**	**2.90**	3.40	3.36	**1.53**	**0.36**	**0.32**	0.66	0.79
300k	Id	5.43	3.16	**2.71**	**2.75**	**2.90**	1.91	0.52	**0.28**	**0.31**	**0.48**
300k	ϕ	**4.38**	**2.85**	2.78	2.90	3.09	**1.15**	**0.34**	0.32	0.44	0.57
400k	Id	5.21	3.13	2.71	**2.62**	**2.92**	1.78	0.52	**0.27**	**0.28**	**0.45**
400k	ϕ	**4.43**	**2.69**	**2.68**	2.96	2.95	**1.17**	**0.31**	0.27	0.43	0.54

Table 1: CIFAR-10 distillation results across training checkpoints and sampling budgets. KID values are multiplied by 10^3. For each checkpoint, reference distribution, and sampling budget, the lower value across the two clocks is highlighted.

Student evaluation. We evaluate EMA student weights, using decay 0.9999 in both image experiments. Comparisons between Id and ϕ use matched training checkpoints and numbers of student network evaluations. The main text reports FID (Heusel et al., 2018); the additional tables report both FID and KID (Bińkowski et al., 2021) at multiple training stages. Both statistics are computed using 50,000 generated samples and legacy TensorFlow-compatible Inception-v3 features. For the KID we use a standard cubic polynomial kernel

$$k(x, y) = \left(\frac{x^\top y}{d} + 1 \right)^3,$$

where d is the Inception feature dimension. We report the mean unbiased estimate of MMD_k^2 over 100 subsets of 1000 samples per distribution.

Each reported configuration uses a single training run. Preliminary repetitions of selected configurations yielded very similar training curves and negligible differences between evaluation results.

Construction of ϕ We used the parameters $p = 2$, $\delta = 1$, $\xi = 10^{-5}$, $N = 100$ as described in 1. In practice, we found that the parameter p does not have a major effect on the resulting ϕ (for tested values in $[1, 4]$). Moreover the numerical floor ξ has negligible effect in all our experiments since the estimate h_p has very high values (in range 10^2 and above).

C.2 CIFAR-10

Data. We use the $50{,}000$ training images of CIFAR-10 (Krizhevsky et al., 2009) at their native 32×32 resolution, without class conditioning. The source distribution is $\rho_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. We add horizontal flips. The images are then mapped to [-1,1].

Architecture. The teacher and students use the guided-diffusion U-Net architecture of Dhariwal & Nichol (2021), as implemented in TorchCFM (Tong et al., 2023; 2024), with base width 128 and channel multipliers $(1, 2, 2, 2)$. As in Boffi et al. (2025b), the student adapts the time-conditioning

pathway to accept two times. The corresponding resolution–channel pairs are

$$(32^2, 128), \quad (16^2, 256), \quad (8^2, 256), \quad (4^2, 256).$$

The backbone uses two encoder residual blocks and three skip-connected decoder blocks per resolution, with self-attention at 16×16 and in the bottleneck. Attention uses four heads with 64 channels per head. The models contain approximately 35M parameters.

The teacher embeds physical time t, whereas the student embeds α and β separately, using 128-dimensional Fourier features with time-input scale 1000. These features are projected into the residual-block conditioning stream.

Pretrained teacher. We use a publicly released TorchCFM checkpoint (Tong et al., 2023; 2024). The supplied training configuration uses Adam with learning rate $2 \cdot 10^{-4}$, default optimizer parameters, a batch size of 128, $5{,}000$ learning-rate warmup updates, gradient-norm clipping at 1.0, EMA decay 0.9999, and $400{,}000$ training updates. The exact checkpoint is TorchCFM release v1.0.4, `cfm_cifar10_weights_step_400000.pt`, trained using Independent Conditional Flow Matching (I-CFM) with independent source–target coupling and noise $\sigma = 0$. The teacher achieves approximately 50k-FID 3.5 (see Fig 3)

Student optimization. Both students use RAdam (Liu et al., 2020), a global batch size of 196, and $400{,}000$ optimizer updates. The learning rate at optimizer update k is

$$\eta(k) = 2 \cdot 10^{-4}\, 0.992^{k/1000}.$$

Both students use the learned weighting from Appendix C.1.

Evaluation. The main-text figure uses the checkpoint at 300k training steps. Full results across inference budgets and training steps are reported in Table 1.

C.3 CelebA64

Data and preprocessing. We use the official CelebA training partition (Liu et al., 2015), containing $162{,}770$ images. Validation uses $3{,}072$ fixed images sampled with seed 0 from the official validation partition. The official test partition is not used for training. Images are center-cropped or padded to 140×140, resized to 64×64 using area interpolation, and converted from uint8 RGB to $[-1, 1]$. We use no horizontal flipping or other online augmentation.

The source distribution is $\rho_0 = \mathcal{N}(\mathbf{0}, 0.25\boldsymbol{I})$. Source and data samples are coupled independently. Teacher training uses uniformly sampled physical times and linear interpolation without additional noise.

Architecture. We use the NVLabs EDM2 U-Net backbone (Karras et al., 2024) with base width 128 and channel multipliers $(1, 2, 3, 4)$, giving

$$(64^2, 128), \quad (32^2, 256), \quad (16^2, 384), \quad (8^2, 512).$$

There are three encoder residual blocks and four skip-connected decoder blocks per resolution, with attention at 16×16 and 8×8. Teacher dropout is zero; student dropout is 0.1. The teacher and student U-Nets contain $124{,}198{,}821$ and $124{,}264{,}357$ trainable parameters, respectively.

The backbone uses magnitude-preserving convolutions, pixel normalization, magnitude-preserving residual and skip mixing, normalized attention, and learnable residual and embedding gains. An internal constant-one plane is appended to RGB, so the first convolution receives four channels. No spatial time-conditioning planes are used.

The teacher uses a deterministic 128-dimensional Fourier embedding of physical time,

$$z(t) = \sqrt{2}\,[\cos(t\omega), \sin(t\omega)],$$

with logarithmically spaced frequencies and maximum period 10^4. A magnitude-preserving projection maps this embedding to a 512-dimensional conditioning stream. Students instead concatenate separate 128-dimensional embeddings of α and β and project the result to 512 dimensions. After magnitude-preserving SiLU, the student embedding modulates every residual block through the original conditioning pathway.

Training Steps	TP	FID to target					FID to teacher				
		1	2	4	8	20	1	2	4	8	20
150k	Id	8.41	5.73	**4.70**	**4.35**	**4.15**	3.93	1.73	**1.48**	**1.76**	1.98
150k	ϕ	**7.49**	**5.34**	4.70	4.44	4.37	**3.34**	**1.66**	1.64	1.77	**1.89**
200k	Id	7.56	5.14	**4.18**	**3.92**	**3.82**	3.35	1.44	**1.25**	**1.57**	1.81
200k	ϕ	**6.75**	**4.92**	4.40	4.13	4.04	**2.79**	**1.39**	1.47	1.61	**1.76**
250k	Id	6.98	4.78	**3.93**	**3.74**	**3.66**	2.93	1.26	**1.14**	**1.51**	**1.75**
250k	ϕ	**6.19**	**4.57**	4.20	4.03	3.95	**2.41**	**1.23**	1.42	1.63	1.77
300k	Id	6.56	4.54	**3.75**	**3.60**	**3.58**	2.60	**1.11**	**1.06**	**1.46**	1.74
300k	ϕ	**5.87**	**4.39**	4.05	3.93	3.83	**2.16**	1.13	1.34	1.59	**1.73**
350k	Id	6.35	4.47	**3.74**	**3.59**	**3.56**	2.43	**1.07**	**1.05**	**1.46**	1.75
350k	ϕ	**5.66**	**4.28**	3.95	3.82	3.70	**2.00**	1.09	1.32	1.59	**1.74**
400k	Id	6.19	4.35	**3.63**	**3.51**	**3.51**	2.34	**1.02**	**1.01**	**1.44**	1.74
400k	ϕ	**5.58**	**4.25**	3.94	3.80	3.68	**1.90**	1.07	1.31	1.56	**1.71**

Training Steps	TP	KID to target ($\times 10^3$)					KID to teacher ($\times 10^3$)				
		1	2	4	8	20	1	2	4	8	20
150k	Id	5.43	3.55	3.06	3.01	2.91	2.72	1.05	**1.17**	1.69	1.92
150k	ϕ	**4.67**	**3.22**	**3.01**	**2.86**	**2.83**	**2.22**	**1.00**	1.39	**1.54**	**1.61**
200k	Id	4.84	3.13	**2.68**	2.71	2.72	2.32	0.86	**1.01**	1.54	1.79
200k	ϕ	**4.25**	**3.00**	2.86	**2.70**	**2.68**	**1.91**	**0.84**	1.25	**1.42**	**1.51**
250k	Id	4.43	2.93	**2.53**	**2.60**	**2.60**	1.99	**0.75**	**0.93**	**1.48**	1.73
250k	ϕ	**3.87**	**2.80**	2.76	2.70	2.68	**1.65**	0.77	1.25	1.49	**1.58**
300k	Id	4.15	2.78	**2.43**	**2.53**	**2.59**	1.74	**0.64**	**0.86**	**1.44**	1.74
300k	ϕ	**3.67**	**2.69**	2.65	2.64	2.62	**1.49**	0.73	1.19	1.47	**1.57**
350k	Id	4.01	2.76	**2.43**	**2.54**	2.59	1.61	**0.62**	**0.86**	**1.44**	1.75
350k	ϕ	**3.53**	**2.63**	2.61	2.60	**2.54**	**1.36**	0.70	1.19	1.49	**1.61**
400k	Id	3.89	2.66	**2.33**	**2.45**	**2.54**	1.56	**0.58**	**0.82**	**1.42**	1.74
400k	ϕ	**3.49**	**2.63**	2.62	2.59	2.54	**1.27**	0.70	1.19	1.47	**1.57**

Table 2: CelebA64 distillation results across training checkpoints and sampling budgets. KID values are multiplied by 10^3. For each checkpoint, reference distribution, and sampling budget, the lower value across the two clocks is highlighted.

Teacher optimization. The teacher minimizes the independent conditional flow-matching loss, averaged over batch, channels, and spatial dimensions, without learned loss weighting. We use RAdam with $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$, a global batch size of 256, and trained for $350{,}000$ updates. The learning-rate schedule is

$$\eta(k) = \frac{10^{-2}}{\sqrt{\max(k/35000, 1)}}.$$

Training uses four A100 GPUs with 64 images per GPU, BF16 computation, gradient-norm clipping at 1.0, and EMA decay 0.9999. EDM2 weights are renormalized after every optimizer update.

We freeze the step-$350{,}000$ EMA state as an immutable, hashed snapshot for distillation. Its evaluation with 100 Heun steps and $50{,}000$ generated samples gives approximately FID 2.36 and KID $1.46 \cdot 10^{-3}$.

Student optimization and weighting implementation. Both students use one RAdam optimizer with separate parameter groups for the U-Net and weighting head. The groups share the same learning rate, $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay. Students are trained for $400{,}000$ updates with learning-rate schedule

$$\eta(k) = 3 \cdot 10^{-3}\, 0.992^{k/1000}.$$

The global batch size is 256, with 64 examples per GPU on four GPUs. We use BF16 computation, gradient-norm clipping at 1.0, and EMA decay 0.9999. EDM2 weights are renormalized after every optimizer update and every EMA update. There is no early stopping.

For Equation 6, $\ell_{\alpha,\beta}$ is a sum over all $3 \cdot 64 \cdot 64$ image coordinates. Weighted losses are then averaged over the batch. The weighting head embeds α and $\beta - \alpha$ separately with 128-dimensional sinusoidal features, concatenates them, and applies a zero-initialized linear projection to one scalar. It contains 257 parameters in addition to the U-Net.