PATHGUIDE: DYNAMIC CLASSIFIER-FREE GUIDANCE VIA ON-POLICY TRANSPORT ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

While modern generative models excel at modeling complex data, precise inference-time control in conditional generation remains a critical challenge. Classifier-free guidance (CFG) is a primary mechanism for such control, yet it is typically treated as a static tuning parameter. In flow-based models, however, the guidance scale fundamentally dictates the velocity field and the resulting probability path, making guidance selection a dynamic path-optimization problem. We introduce *PathGuide*, a framework that reformulates scalar CFG selection as an on-policy transport problem. Leveraging the *weak form of the continuity equation*, we derive a selection criterion with a direct path-correctness interpretation: we prove that if the guided field is weakly equivalent to the exact conditional field along the generated rollout, the sampler's path coincides with the target conditional law. For scalar CFG, this criterion yields a strictly quadratic local objective with an efficient, closed-form selector for each solver interval. PathGuide enables optimal guidance scales to be computed and used *online* during generation or fitted offline as a *reusable* piecewise-constant schedule. We validate our method on controlled settings across various continuous-time flow constructions, demonstrating that this transport-based selector improves path alignment and sample fidelity over both fixed and state-of-the-art adaptive guidance baselines.

1 INTRODUCTION

Modern generative modeling has achieved remarkable success due to its ability to model complex data distributions with minimal discrepancy between real and generated data laws. This is largely achieved by learning continuous-time dynamics (Dhariwal & Nichol, 2021b; Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022; Betker et al., 2023; Esser et al., 2024). Within this paradigm, score-based models learn reverse stochastic or probability-flow dynamics (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b;a; Karras et al., 2022), while flow-based models learn probability paths via velocity-field regression (Lipman et al., 2023; Liu et al., 2022; Tong et al., 2024; Albergo et al., 2025; Huang et al., 2026; Chen & Lipman, 2024; Kapuśniak et al., 2024).

Conditional generative modeling has enabled a vast array of machine learning applications, including text-to-image synthesis (Ramesh et al., 2022; Rombach et al., 2022; Saharia et al., 2022; Podell et al., 2023; Dai et al., 2023), video generation (Blattmann et al., 2023b;a; Singer et al., 2022), audio synthesis (Evans et al., 2024; Wang et al., 2023; Le et al., 2023), robotics and decision-making (Chi et al., 2023; Chen et al., 2021; Janner et al., 2021; 2022; Ajay et al., 2022), biomolecular modeling (Abramson et al., 2024; Corso et al., 2023; Yim et al., 2023), and post-training control tasks such as editing, inpainting, and spatial motion constraints (Avrahami et al., 2022; Brooks et al., 2023; Meng et al., 2022; Lugmayr et al., 2022; Zhang et al., 2023; Bar-Tal et al., 2023; Garipov et al., 2023; Watanabe et al., 2026).

In these settings, inference-time control is paramount. Once a large-scale generator is trained, it is typically steered toward a desired condition using Classifier-Free Guidance (CFG), which usually relies on a constant scale or a hand-crafted schedule (Dhariwal & Nichol, 2021b; Ho & Salimans, 2022; 2021; Zheng et al., 2023; Lipman et al., 2024). While effective at improving conditional alignment, misconceptions regarding the resulting sample law persist (Bradley & Nakkiran, 2024; Chidambaram et al., 2024), revealing deep path-wise inconsistencies: different segments of the trajectory often require varying guidance strengths (Kynkäänniemi et al., 2024; Wang et al., 2024;

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.

Castillo et al., 2023; Jin et al., 2026). Excessive guidance scales can reduce diversity, trigger mode collapse and push samples away from the learned manifold (Chung et al., 2025; Jin et al., 2025; Wang et al., 2025; Saini et al., 2025; Cai et al., 2026; Fan et al., 2025; Sadat et al., 2025; He et al., 2024; Chidambaram et al., 2024). Consequently, guidance selection should be treated as a path-optimization problem rather than a simple hyperparameter tuning exercise.

Existing methods attempt to mitigate these failures by heuristically adapting the guidance scale, training auxiliary networks for path correction, or imposing geometric constraints to counteract faulty interactions between the guidance scale and the model (Kynkäänniemi et al., 2024; Wang et al., 2024; Sadat et al., 2023; Xia et al., 2024; Yehezkel et al., 2025; Galashov et al., 2026; Koulischer et al., 2025; Chung et al., 2025; Jin et al., 2025; Fan et al., 2025; Wang et al., 2025; Saini et al., 2025; Cai et al., 2026; Watanabe et al., 2026; Sadat et al., 2025; He et al., 2024; Guo et al., 2024). While useful, these approaches do not address a fundamental transport problem: which scalar CFG value ensures that the sampler's actual rollout remains faithful to the exact conditional probability path.

We introduce PathGuide, a framework that reformulates scalar CFG selection as an on-policy transport problem. Rather than treating guidance as a heuristic scaling factor, we derive a principled criterion using the weak form of the continuity equation (Villani, 2009; Ambrosio et al., 2005; Santambrogio, 2015) to establish a theoretical foundation for path correctness. Specifically, we derive a local objective with a direct path-correctness interpretation: if the guided field is weakly equivalent to the exact conditional field under the generated rollout, then - under the uniqueness of the weak continuity equation (DiPerna & Lions, 1989; Ambrosio, 2004) - the generated rollout coincides with the exact conditional probability path. By evaluating this objective directly on the sampler's empirical rollout and leveraging the endpoint-conditioned structure of flow matching (Lipman et al., 2023; Tong et al., 2024; Albergo et al., 2025), we translate this theory into a practical framework through the following contributions:

- We derive a rigorous local objective providing a principled measure of compatibility between the sampler's realized rollout and the target conditional law at each time step.
- We prove that for the scalar CFG family, this objective is *strictly quadratic* and admits an efficient closed-form for the optimal guidance scale.
- We introduce a practical algorithm that supports both *real-time online adaptation* during sampling and the offline generation of *reusable*, piecewise-constant guidance schedules.
- Using controlled Gaussian-mixture flow-matching experimental validation across multiple flow families, we demonstrate consistent improvements in path alignment and endpoint fidelity over existing training-free baselines under matched protocols.

2 Related Work

The emergence of Flow Matching and Rectified Flow has shifted the generative modeling paradigm from discrete diffusion steps to continuous-time transport along learned velocity fields (Lipman et al., 2023; Liu et al., 2022; Tong et al., 2024; Albergo et al., 2025). While these works utilize the decomposition of marginal dynamics into endpoint-conditioned paths as a training objective, we repurpose this decomposition as an inference-time diagnostic tool to evaluate whether a guided velocity field remains locally consistent with the ground-truth transport law.

Ensuring this consistency is non-trivial because the optimal guidance scale is not a global constant. In diffusion and flow models alike, studies show that different intervals of the generative trajectory exhibit varying sensitivities to conditioning (Kynkäänniemi et al., 2024; Wang et al., 2024). While this observation has motivated adaptive, annealed, and feedback-based schedulers that modulate guidance via heuristic triggers or learned policies (Castillo et al., 2023; Jin et al., 2026; Yehezkel et al., 2025; Galashov et al., 2026; Koulischer et al., 2025), such methods often lack a formal distributional guarantee. In contrast, we derive a variational criterion directly from the weak continuity equation (DiPerna & Lions, 1989; Ambrosio, 2004; Ambrosio et al., 2005; Santambrogio, 2015), providing a first-principles derivation for guidance schedules that preserves the integrity of the probability path without the need for manual tuning or auxiliary training.

Beyond merely scheduling the guidance scale, a concurrent line of research focuses on "repairing" the classifier-free guidance update itself to mitigate known failure modes like mode collapse (Dhariwal & Nichol, 2021b; Ho & Salimans, 2022). These manifold-aware corrections (Chung et al., 2025; Jin et al., 2025; Sadat et al., 2025; He et al., 2024; Guo et al., 2024) and solver-level refinements (Fan et al., 2025; Wang et al., 2025; Cai et al., 2026; Saini et al., 2025) aim to stabilize the solver by optimizing the *form* of the update. Our work, however, optimizes the scalar *input* to that update.

This shift toward dynamic selection reflects a broader trend where generative models are evaluated not just by terminal samples, but by the alignment of the entire induced probability path (Huang et al., 2026; Albergo et al., 2025; Watanabe et al., 2026). Our work further advances this frontier by treating guidance selection as an on-policy transport problem; unlike existing schedules defined *a priori* on idealized paths, our framework optimizes the guidance scale directly on the sampler's realized rollout to ensure the trajectory remains faithful to the target conditional distribution.

The remainder of this paper is organized as follows: Section 3 establishes the foundations of continuous-time flows and guidance, Section 4 introduces the PathGuide framework and our on-policy consistency objective, and Section 5 demonstrates that our framework is robust across various continuous-time flow constructions, ranging from Optimal Transport to Variance-Preserving dynamics.

3 Continuous-Time Flows and Guidance

Continuous-time generative models have recently emerged as a state-of-the-art paradigm for high-dimensional distribution modeling (Chen et al., 2018; Song et al., 2021b). We consider a target data distribution $q(x_1)$ on $\mathbb{R}^d$. A flow model describes a probability path p_t that evolves over the normalized time interval $t \in [0, 1]$, such that p_0 is a tractable reference distribution, e.g., a standard Gaussian, and $p_1 \approx q$.

A time-dependent vector field (or velocity field) $u_t : \mathbb{R}^d \to \mathbb{R}^d$, defines the dynamics of the generative process. This field induces a flow map $\phi_t : \mathbb{R}^d \to \mathbb{R}^d$ through the ordinary differential equation (ODE)

$$\frac{d}{dt}\phi_t(x) = u_t(\phi_t(x)), \qquad \phi_0(x) = x. \tag{1}$$

The probability density p_t at any time t is the pushforward of the initial law under this map, denoted $p_t = (\phi_t)_{\#}p_0$. This relationship implies that the pair (p_t, u_t) satisfies the continuity equation (Villani, 2009; Ambrosio et al., 2005), which describes the local conservation of probability mass:

$$\frac{\partial}{\partial t}p_t(x) + \nabla \cdot \big(p_t(x)u_t(x)\big) = 0. \tag{2}$$

For a smooth compactly supported test function ψ, the same equation can be written in weak form as (Santambrogio, 2015):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathbb{R}^d} \psi(x)p_t(x)\ \mathrm{d}x = \int_{\mathbb{R}^d} \nabla\psi(x) \cdot u_t(x)\, p_t(x)\ \mathrm{d}x. \tag{3}$$

Classifier-free guidance for flow models. In conditional generation, we aim to sample from $q(x_1 \mid y)$ by steering the flow toward a specific condition y. Classifier-free guidance (CFG) (Ho & Salimans, 2021) was originally developed for score-based diffusion, where a Bayes' rule decomposition of the conditional score replaces an explicitly trained classifier (Dhariwal & Nichol, 2021a); since the same marginal path is generated by a probability-flow ODE whose field depends on the score, guiding the score induces a guided velocity field. Appendix A gives this equivalence in full.

In modern flow-based models, guidance is therefore usually implemented directly at the level of learned velocity fields (Zheng et al., 2023). Let $v_t^\theta(\cdot \mid \varnothing)$ denote the learned unconditional field, and let $v_t^\theta(\cdot \mid y)$ denote the learned conditional field. Classifier-free guidance forms the guided field

$$v_t^\theta(x \mid y; \omega_t) = v_t^\theta(x \mid \varnothing) + \omega_t\Big(v_t^\theta(x \mid y) - v_t^\theta(x \mid \varnothing)\Big), \tag{4}$$

where ω_t is a scalar guidance value. The guidance scale ω_t serves as an extrapolation parameter: $\omega_t = 1$ recovers the nominal learned conditional field, while $\omega_t > 1$ is conventionally used to

amplify the discrepancy between the conditional and unconditional dynamics. This inference-time interface has become a cornerstone of state-of-the-art flow systems (Esser et al., 2024; Black Forest Labs, 2024; StabilityAI, 2024).

Flow Matching. Flow matching (Lipman et al., 2023) provides a framework to learn the conditional field $u_t(\cdot \mid y)$ by marginalizing over endpoint-conditioned paths. For a fixed sample $x_1 \sim q(x_1 \mid y)$, we define an endpoint-conditioned probability path $p_t(\cdot \mid x_1, y)$ and its corresponding vector field $u_t(\cdot \mid x_1, y)$. The exact marginal conditional field $u_t(x \mid y)$ is then defined as:

$$u_t(x \mid y) = \int_{\mathbb{R}^d} u_t(x \mid x_1, y) \frac{p_t(x \mid x_1, y) q(x_1 \mid y)}{p_t(x \mid y)} dx_1. \tag{5}$$

By construction, this marginal field is the unique vector field that generates the conditional marginal path

$$p_t(x \mid y) = \int p_t(x \mid x_1, y) q(x_1 \mid y) dx_1 \tag{6}$$

and satisfies the continuity equation in (2) (Lipman et al., 2023). In practice, we approximate this exact field $u_t(x \mid y)$ with a neural network $v_t^\theta(x \mid y)$ via a regression objective, which uses the known form of $u_t(x \mid x_1, y)$ determined by the flow. At inference time, one can sample $x_t \sim p_t(x \mid y)$ using the learned vector field by sampling $x_0 \sim p_0$ and integrating: $x_t = x_0 + \int_0^t v_s^\theta(x_s \mid y) ds$.

In the idealized setting where the vector fields are exact, i.e., if $v_t^\theta(\cdot \mid y) \equiv u_t(\cdot \mid y)$ for every y, the value $\omega_t \equiv 1$ would be sufficient to recover the exact marginal path $p_t(\cdot \mid y)$. However, in practice, v_t^θ is a learned estimator, and in this regime, ω_t can be interpreted as a corrective parameter that calibrates the guided flow to account for approximation errors in the neural network. In this paper, we develop a principled, theoretically grounded framework for selecting the schedule ω_t to optimally compensate for the mismatch between the frozen learned field and the true underlying transport law.

4 Dynamic Classifier-Free Guidance via On-Policy Transport Alignment

The learned velocity field $v_t^\theta(x \mid y; \omega_t)$, defined in Equation (4), is an imperfect estimator of the true marginal vector field $u_t(x \mid y)$. Because the generative process is sequential, any approximation error introduced at an earlier time $s < t$ propagates through the ODE solver, causing the realized probability path $\hat{p}_t(\cdot \mid y)$ to drift away from the target conditional distribution $p_t(\cdot \mid y)$.

We propose to treat the guidance scale ω_t as a dynamic control variable that targets these accumulated learning errors. At each time, ω_t is chosen to minimize a weak-form mismatch between the guided field and the exact conditional field, evaluated *under the rollout law realized so far*. This on-policy criterion is a tractable surrogate for path correctness: it is exact under the ideal conditions of Proposition 4.1, and Section 5 measures what it achieves on a rollout that has already drifted. We denote the guidance history on the continuous interval $s \in [0, t)$ as $\omega_{[0,t)} = (\omega_s)_{s \in [0,t)}$. With this notation, our goal is to use ω_t to adaptively recalibrate the realized probability path, $\hat{p}_t(\cdot \mid y; \omega_{[0,t)})$, toward the target conditional distribution $p_t(\cdot \mid y)$.

Proposition 4.1 (Ideal weak equivalence implies path correctness)**.** *Fix condition y. Assume that the generated rollout distribution $\hat{p}_s(\cdot \mid y; \omega_{[0,s)})$ is weakly continuous in time and satisfies the weak continuity identity with the guided field v_s^θ (per Assumption B.3):*

$$\frac{\mathrm{d}}{\mathrm{d}s} \int_{\mathbb{R}^d} \psi(x)\, \hat{p}_s(x \mid y; \omega_{[0,s)})\, \mathrm{d}x = \int_{\mathbb{R}^d} \nabla\psi(x) \cdot v_s^\theta(x \mid y; \omega_s)\, \hat{p}_s(x \mid y; \omega_{[0,s)})\, \mathrm{d}x \tag{7}$$

for every $\psi \in C_c^\infty(\mathbb{R}^d)$ and almost every $s \in [0, t]$.

Assume further that the guided field and the exact velocity field $u_s(\cdot \mid y)$ are weakly equivalent under the generated rollout:

$$\int_{\mathbb{R}^d} \nabla\psi(x) \cdot v_s^\theta(x \mid y; \omega_s)\, \hat{p}_s(x \mid y; \omega_{[0,s)})\, \mathrm{d}x = \int_{\mathbb{R}^d} \nabla\psi(x) \cdot u_s(x \mid y)\, \hat{p}_s(x \mid y; \omega_{[0,s)})\, \mathrm{d}x. \tag{8}$$

Since the exact conditional path $p_s(\cdot \mid y)$ is the unique weak solution to the continuity equation driven by $u_s(\cdot \mid y)$ (under Assumption B.4) with the same initial condition $\hat{p}_0 = p_0$, then

$$\hat{p}_s(\cdot \mid y; \omega_{[0,s)}) = p_s(\cdot \mid y), \qquad \forall s \in [0, t]. \tag{9}$$

Proof sketch. Because the generated rollout law satisfies the weak continuity identity under the guided field, and the guided field is weakly equivalent to the exact conditional field under that law, the rollout also satisfies the weak continuity equation driven by the exact conditional field. By the uniqueness of weak solutions (per Assumption B.4) with the same initial condition, the generated rollout path must coincide with the exact conditional path. Full proof is in Appendix B.4. □

While a theoretical continuous-time selector would choose a value at every instant t, a numerical solver typically utilizes a single value per integration interval. Let $0 = t_0 < t_1 < \cdots < t_T = 1$ be the solver grid. We approximate the continuous guidance schedule with a step-wise sequence of scales. At each discrete step $i \in \{0, \ldots, T-1\}$, given the history of previous guidance scales $\vec{\omega}_i(y) = (\omega_0, \ldots, \omega_{i-1})$, abbreviated $\vec{\omega}_i$, we denote by $\hat{p}_{t_i}(\cdot \mid y; \vec{\omega}_i)$ the distribution of the samples generated up to that point. Using the guided vector field $v^{\theta}_{t_i}(x \mid y; \omega_i)$, our goal is to adaptively determine the optimal ω_i for the interval $[t_i, t_{i+1})$. Following Proposition 4.1, we choose ω_i to approximate the ideal weak equivalence condition. Applied iteratively from the first solver step, each update minimizes the local residual *under the rollout realized by the already committed schedule*, so every scale is selected against the state the sampler is actually in rather than against an idealized path:

$$\mathcal{L}_i(\omega_i \mid y; \vec{\omega}_i, \psi) = \left(\int_{\mathbb{R}^d} \Big(u_{t_i}(x \mid y) - v^{\theta}_{t_i}(x \mid y; \omega_i) \Big) \cdot \nabla_x \psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x \right)^2 . \tag{10}$$

4.1 Closed-form local selector under scalar CFG

We now specialize to scalar classifier-free guidance. At this point, the key observation is that under scalar CFG, the objective is affine in the current local control value. See Appendix B.7.

Corollary 4.1 (Exact local selector)**.** *Fix y, and $\psi \in C^{\infty}_c(\mathbb{R}^d)$, and a committed past schedule $\vec{\omega}_i$. Then $\mathcal{L}_i(\omega_i \mid y; \vec{\omega}_i, \psi)$, defined in Equation* (10) *is minimized for*

$$\omega^{\star}_i(y; \vec{\omega}_i, \psi) = \frac{\displaystyle\int_{\mathbb{R}^d} \Big(u_{t_i}(x \mid y) - v^{\theta}_{t_i}(x \mid \varnothing) \Big) \cdot \nabla_x \psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x}{\displaystyle\int_{\mathbb{R}^d} \Big(v^{\theta}_{t_i}(x \mid y) - v^{\theta}_{t_i}(x \mid \varnothing) \Big) \cdot \nabla_x \psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x}, \tag{11}$$

provided the denominator is nonzero. If the denominator vanishes, the objective is constant in ω_i, so every admissible value is optimal.

Proof sketch. Since the guided field is affine in the scalar guidance value ω_i, the local objective is a one-dimensional quadratic. Setting the derivative with respect to ω_i to zero yields $\omega^{\star}_i$. The full proof is provided in Appendix B.8. □

Notably, if the learned conditional field is exact ($u_{t_i} = v^{\theta}_{t_i}$), the numerator and denominator in (11) coincide, yielding $\omega^{\star}_i = 1$ (or making any value optimal if the denominator is zero).

4.2 Endpoint-conditioned representation of the exact objective

The adaptive loss is written in terms of the exact marginal field $u_t(\cdot \mid y)$. However, in flow matching the marginal field is not accessed directly; it is obtained by averaging endpoint-conditioned vector fields with posterior weights, as described in Equation (5). The following equivalence is what makes the objective estimable: it removes $u_{t_i}(\cdot \mid y)$ in favor of endpoint-conditioned quantities that the flow construction supplies in closed form.

Theorem 4.1 (Endpoint-conditioned representation of the local objective)**.** *Fix y, $\psi \in C_c^\infty(\mathbb{R}^d)$, and a committed past schedule $\vec{\omega}_i$. Under Assumptions B.1 and B.2, the exact local objective* (10) *admits the endpoint-conditioned form $\mathcal{L}_i(\omega_i \mid y; \vec{\omega}_i, \psi) =$*

$$\left(\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} (u_{t_i}(x \mid y, x_1) - v^\theta_{t_i}(x \mid y; \omega_{t_i})) \frac{p_{t_i}(x \mid y, x_1)\, q(x_1 \mid y)}{p_{t_i}(x \mid y)} \, \mathrm{d}x_1 \cdot \nabla\psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i) \, \mathrm{d}x \right)^2, \tag{12}$$

an identity that is exact at the population level.

Proof sketch. The posterior weights integrate to one in x_1, so the guided field - which carries no endpoint dependence - may be moved inside the endpoint integral. Writing the marginal field $u_{t_i}(\cdot \mid y)$ as its posterior average over x_1 via Equation (5) and merging the two integrals gives the displayed form. The full proof is in Appendix B.6. □

To estimate Equation (12) we use a Monte Carlo approximation of these integrals: At time t_i, we have N rollout particles $x_i^{(n)} \sim \hat{p}_{t_i}(\cdot \mid y; \vec{\omega}_i)$, and we sample M endpoint samples $x_1^{(m)} \sim q(\cdot \mid y)$.

The practical procedure follows a recursive logic, progressing interval by interval through the solver grid. At each step i, we estimate the local coefficients of the quadratic objective $\tilde{\mathcal{L}}_i$ using the current generated rollout particles $\{x_i^{(n)}\}_{n=1}^N$. We then compute the optimal local guidance scale $\omega_i^\star$ via Equation (11), commit this value for the duration of the interval $[t_i, t_{i+1})$, and advance the ODE solver one step to obtain the particles at t_{i+1}. This process continues until the terminal time $t_T = 1$ is reached. An implementation of this procedure is detailed in Appendix C.

Note that the backbone θ is never modified, but calibration is a *model-owner* operation: it needs endpoint samples $x_1 \sim q(\cdot \mid y)$ and a known path construction that makes $p_t(x \mid x_1, y)$ and $u_t(x \mid x_1, y)$ evaluable (Appendix D). Once fitted, none of these quantities is needed at deployment: the stored schedule is a list of T scalars consumed exactly like a constant CFG scale.

Moreover, Proposition 4.1 characterizes the *ideal* target: weak equivalence for every $\psi \in C_c^\infty(\mathbb{R}^d)$ and almost every s. An implemented schedule realizes a finite counterpart of this target, and its relation to the real probability-path discrepancy is quantified empirically in Appendix F.1.

4.3 Usage modes

This framework introduces an **on-line selector**: the guidance value applied to each interval is adaptively chosen based on the actual distribution $\hat{p}_{t_i}$ realized by the frozen model and solver up to that point. This approach leads to two primary deployment modes:

- **Online Adaptive Selection:** In this mode, the guidance scale is recalibrated dynamically during every sampling run. This is particularly effective for high-fidelity generation where the specific drift of a sample batch must be corrected in real-time.
- **Offline Calibrated Scheduling:** For a fixed backbone model, solver, and grid, the selector can be run once on a representative set of initial samples $x_0 \sim p_0$. The resulting sequence of optimal scales $\vec{\omega}^\star$ is stored as a piecewise-constant schedule. This schedule can then be reused across all subsequent deployments, providing a *pay-once, use-many-times* solution that enjoys the corrective benefits of our method without the overhead of computing the selector at every inference step.

By decoupling the estimation of the drift from the generative inference, we provide a flexible mechanism to stabilize probability paths in both compute-constrained and quality-critical settings.

5 Experimental Validation

We evaluate the selector on a class-conditional Gaussian-mixture testbed. This setting is deliberately controlled: the endpoint law, intermediate conditional marginals, and posterior weights are available in closed form. It therefore lets us test the main claim of Section 4: the selected guidance schedule should improve the generated rollout $\hat{p}_{t_i}(\cdot \mid y; \vec{\omega}_i)$, not only the terminal samples at $t = 1$.

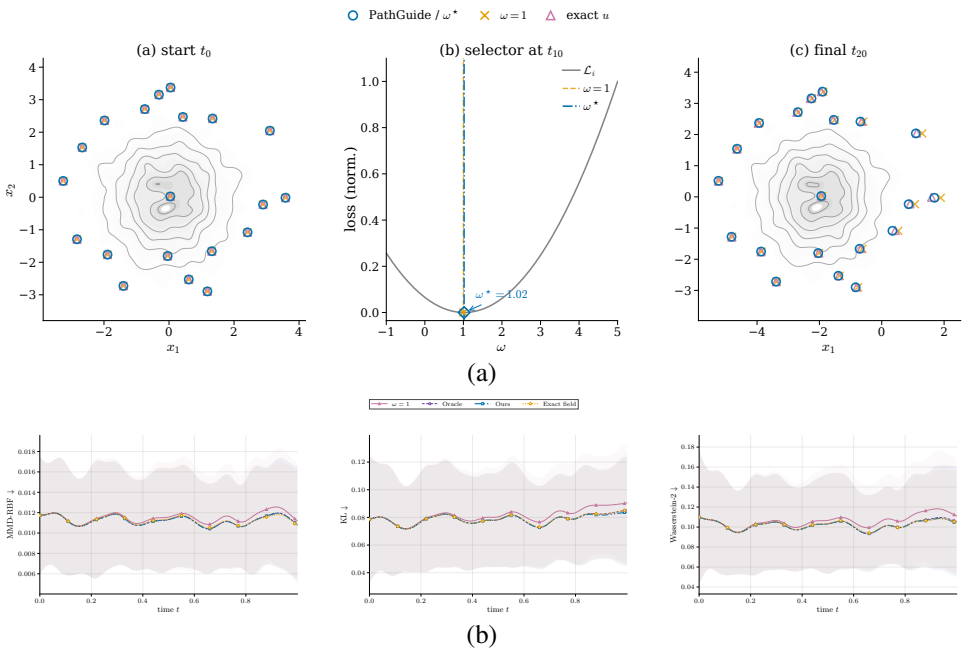


Figure 1: Comparison of rollout trajectories against the exact conditional marginal $p_{t_i}(\cdot \mid y)$. (a) Online samples at the start (t_0) and near the end (t_{T-1}) of a $T = 20$ step rollout, together with the quadratic local objective and the selected value at the midpoint time $t_{T/2}$. (b) Quantitative discrepancy measured by MMD, KL, and W_2. The true-velocity rollout serves as a discretization reference, since finite-step integration can deviate from the analytic marginal path even when the exact conditional velocity is used. While the plain conditional learned field ($\omega \equiv 1$) accumulates drift, our practical selector effectively calibrates the rollout to track the true path more closely.

All schedules are fitted separately for each class, and evaluation metrics are aggregated over classes using the class frequencies. Unless stated otherwise, each method is evaluated over three inference seeds, with matched prior latents across methods within each seed. Schedules are fitted using three independent fitting seeds. Downstream results report the mean and sample standard deviation over inference seeds. Full experimental details, including flow variants, test-function families, stabilization, metrics, and implementation defaults, are provided in Appendix E. Ablations are provided in Appendix F.

Online path alignment. We first test whether the selected guidance values improve the path followed by the sampler. Since the exact conditional marginal $p_{t_i}(\cdot \mid y)$ is known at every solver time, we measure the discrepancy between $p_{t_i}(\cdot \mid y)$ and the generated rollout $\hat{p}_{t_i}(\cdot \mid y; \vec{\omega}_i)$ along the full trajectory. Figure 1 compares three rollouts: one generated with the true conditional velocity $u_t(\cdot \mid y)$, one generated with the learned conditional field $v_t^\theta(\cdot \mid y)$, which corresponds to $\omega \equiv 1$, and one generated by the practical selector. Endpoint discrepancies at $t \approx 1$ are reported in Table 17.

Endpoint quality against guidance baselines. We next test whether improved path alignment translates into better terminal samples. Table 1 compares the final generated distribution at $t = 1$ against plain CFG (Ho & Salimans, 2022; Zheng et al., 2023), CFG-Zero* (Fan et al., 2025), CFG-MP (Cai et al., 2026), and Rectified-CFG++ (Saini et al., 2025). Appendix F.2 adds a control that isolates the weak-form criterion itself, selecting the scale from the same endpoint-conditioned estimates by direct pointwise projection instead. The schedule for our method is fitted once on a reference solver grid and then reused on fresh inference seeds, matching the offline calibrated scheduling mode described in Section 4, isolating the benefit of selecting $\vec{\omega}$. All methods in a comparison are run under identical conditions: the same frozen velocity field, endpoint law, solver grid, initial latents, and inference seeds, with each baseline tuned over its own hyperparameter sweep (Appendix E.5). We also evaluate the same protocol on MNIST handwritten digits (LeCun et al., 1998). Table 2 shows that the fitted schedules remain effective in pixel space for both Optimal Transport (OT) and Rectified Flow (RF) variants.

Table 1: Generation quality against guidance baselines for all flow variants. Each metric is reported for $T = 200$ and $T = 500$; lower is better. All methods are evaluated using 2^{14} generated samples per seed. Our method is fitted once using $M = N = 2^{14}$, independently of the evaluation samples. Each row reports the best configuration selected from that method's hyperparameter sweep, using the selection rule described in Appendix E.5. Full sweeps including standard deviations are in Appendix H.

		KL↓		W_2 ↓		MMD↓	
Variant	Method	$T = 200$	$T = 500$	$T = 200$	$T = 500$	$T = 200$	$T = 500$
RF (Liu et al., 2022)	Plain CFG	0.0044	0.0044	0.0060	0.0060	0.0006	0.0006
	CFG-Zero*	0.0046	0.0045	0.0064	0.0062	0.0006	0.0006
	CFG-MP	6.2232	1.8740	9.5319	2.9592	0.6234	0.2987
	R-CFG++	0.0191	0.0185	0.0248	0.0243	0.0020	0.0019
	PathGuide (Ours)	**0.0039**	**0.0038**	**0.0056**	**0.0055**	**0.0005**	**0.0005**
I-CFM (Tong et al., 2024)	Plain CFG	0.0084	0.0083	0.0112	0.0111	0.0012	0.0012
	CFG-Zero*	0.0084	0.0083	0.0114	0.0111	0.0012	0.0012
	CFG-MP	0.1197	0.1191	0.1853	0.1859	0.0168	0.0167
	R-CFG++	0.0192	0.0185	0.0252	0.0246	0.0024	0.0023
	PathGuide (Ours)	**0.0076**	**0.0073**	**0.0099**	**0.0096**	**0.0012**	**0.0012**
OT (Lipman et al., 2023)	Plain CFG	0.0082	0.0080	0.0111	0.0110	0.0012	0.0012
	CFG-Zero*	0.0081	0.0080	0.0112	0.0110	**0.0012**	0.0011
	CFG-MP	0.1215	0.1208	0.1889	0.1895	0.0171	0.0170
	R-CFG++	0.0199	0.0192	0.0266	0.0260	0.0024	0.0024
	PathGuide (Ours)	**0.0076**	**0.0073**	**0.0102**	**0.0099**	0.0012	**0.0011**
VP (Song et al., 2021b)	Plain CFG	0.0030	0.0030	0.0052	0.0052	0.0005	0.0005
	CFG-Zero*	0.0013	0.0013	0.0019	0.0019	0.0003	0.0003
	CFG-MP	2.9991	1.1612	4.7929	1.6279	0.4827	0.2012
	R-CFG++	0.0290	0.0289	0.0388	0.0387	0.0029	0.0028
	PathGuide (Ours)	**0.0011**	**0.0011**	**0.0015**	**0.0015**	**0.0003**	**0.0003**

Table 2: MNIST image-generation quality for RF and OT flow variants. We report FID computed with `torch-fidelity==0.4.0`. Lower is better. All methods are evaluated over three evaluation seeds using 2^{13} generated samples per seed, with $M = 2^{11}$, $N = 2^{12}$, and $T = 50$. Each row reports the configuration with the lowest mean FID within that method's sweep. Full sweeps are reported in Table 16. On matched inference latents, the paired PathGuide-minus-tuned-CFG FID difference is -0.103 (95% CI $[-0.186, -0.021]$) for RF and -0.175 ($[-0.250, -0.101]$) for OT.

RF (Liu et al., 2022)		OT (Lipman et al., 2023)	
Method	FID↓	Method	FID↓
Plain CFG	15.802	Plain CFG	7.578
CFG-Zero*	26.534	CFG-Zero*	11.327
CFG-MP	30.951	CFG-MP	24.475
R-CFG++	16.979	R-CFG++	15.038
PathGuide (Ours)	**15.699**	PathGuide (Ours)	**7.375**

Reuse under coarser inference. The offline schedule is more useful if it can be reused under cheaper inference. We therefore fit schedules on a reference grid T, compress them to coarser grids $T^- \leq T$ aggregating with an average over ω_i included in each corresponding coarse interval, and evaluate the resulting terminal distributions using matched latents. Figure 2 reports the degradation as the inference grid is coarsened. The fitted schedules degrade gracefully ($\leq 4\%$), supporting the "fit once, reuse later" interpretation of the method.

Schedule diagnostics. We then inspect whether the fitted schedules follow the local weak-form objective derived in Section 4. Figure 3 shows the $T = 500$ analytic local-objective landscape over (t_i, ω) for a representative VP run, with fitted schedules overlaid for different solver grids. The fitted schedules follow the low-objective region and preserve the same global shape across fitting resolutions. The *oracle schedule* uses the same interval-wise selector, but replaces the estimated posterior weights with the analytic posterior weights available in this controlled setting. Finer grids reduce estimated schedule variance and better track the oracle schedule.

Ablation. Appendix F ablates the selector's components. Table 7 shows that a first- and second-order test-function mix beats either order alone. Figure 5 shows downstream metrics and schedule

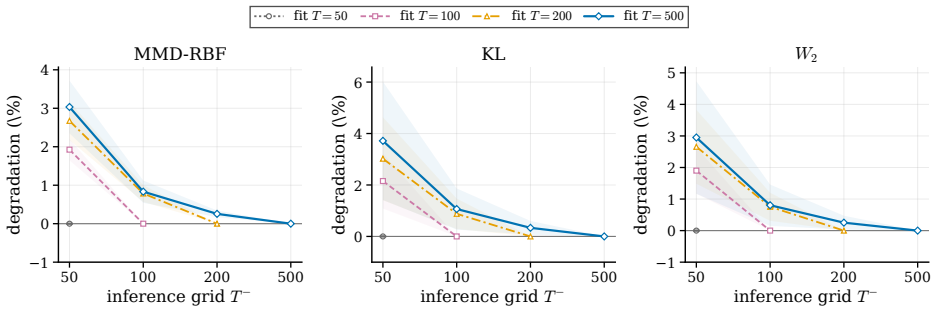


Figure 2: Reusing a schedule fitted at grid T on coarser inference grids $T^- \leq T$. Each curve fixes the fitting grid and varies the deployed inference grid. $\text{degradation} = 100 \cdot (\text{coarse result} - \text{full result})/\text{full result}$, each result is averaged over the fitting seeds. Shaded regions show one standard deviation over the inference seeds. Lower is better for all metrics.

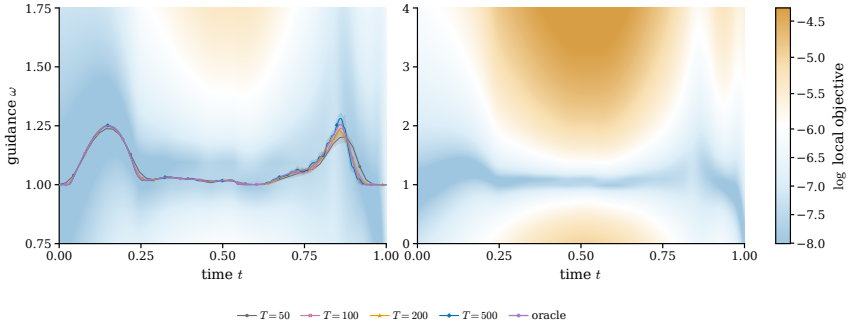


Figure 3: Schedule diagnostic for the VP flow, class 0. The background shows the analytic local-objective landscape over (t_i, ω). Curves show fitted schedules for different solver grids together with the oracle schedule. The practical schedules preserve the same global shape across resolutions and better follow low-objective regions as the grid is refined.

distance stabilizing at moderate budgets. Appendix E.6 shows low sensitivity to the probe draw: at the default setting, with all other randomness fixed, independent draws give a mean schedule standard deviation of 0.00207 for the default first- and second-order mix. Table 6 shows that shifting the same budget toward third-order probes keeps schedules close to, and slightly better than, the default, while an all-third-order family fails outright, supporting the principled mixed default. Table 8 shows that the held-out residual tracks the resulting path discrepancy (mean Spearman 0.46–0.66).

6 Conclusion and Future Work

We introduced *PathGuide*, a framework that reformulates scalar CFG selection as an on-policy transport problem grounded in the weak form of the continuity equation. By deriving a strictly quadratic local objective, we provide a closed-form selector that encourages generative rollouts to remain faithful to the target conditional law. Despite its theoretical and empirical advantages, our method has several limitations. Once a schedule is fitted and amortized, inference has the same backbone-evaluation cost as standard CFG, however, online selection adds estimator overhead during sampling. Furthermore, the selector is local in time, and future work could extend it to a global optimal-control formulation. Moreover, the framework's sensitivity to the test-function family $\mathcal{H}$ introduces a trade-off between discrepancy detection and computational variance (Appendix E.6), suggesting the use of adaptive or learned kernels as an extension. While we focused on scalar CFG, the transport-based objective is naturally extensible to more expressive, non-linear control signals. Finally, our empirical scope is a two-class Gaussian mixture and MNIST. We therefore make no claim of generalization to large backbones, higher dimensions, or open-ended conditional generation; evaluating the weak-form objective on large-scale text-to-image and video benchmarks - particularly for reward-tilted or tempered distributions - remains a critical next step.

AI USE STATEMENT

We used generative AI tools in the preparation of this manuscript. Their use was confined to writing, formatting, and auxiliary tooling. The research itself - the theoretical framework, the mathematical claims and their proofs, the method, its implementation, and the experiments - is the authors' own work.

We did *not* use generative AI tools to generate synthetic datasets, to develop the theoretical model or conceptual framework, to formulate mathematical claims, to provide critical ingredients for proving mathematical claims, to assist in the writing of proofs, to propose or refine hypotheses, to design or provide feedback on the research methodology or experiments, to support qualitative or thematic data analysis, or to interpret results. The continuity-equation criterion, the closed-form selector, the proofs in Appendix B, the experimental design, and the reported interpretations are the authors' own. Regarding the implementation of methods: the PathGuide method implementation and the experimental pipeline were written by the authors, and generative AI assisted only with auxiliary helper and analysis scripts that are not part of the method. Translation and the cleaning or reformatting of datasets are not applicable to this work.

We used generative AI tools for: drafting and revising portions of the manuscript text, editing for readability, creating and editing auxiliary software and analysis scripts, document formatting (including figure sizing and layout), compiling and formatting the bibliography, sourcing and verifying reference metadata, suggesting document structure, and brainstorming presentation alternatives. No figure in this paper was generated or modified in content by AI. All figures were produced by the authors' own code, and AI assistance was limited to their sizing within the document.

We have reviewed all AI-assisted work. All mathematical claims and proofs were derived and checked by the authors. All reported numbers were produced by the authors' code and verified against the logged experimental outputs. All AI-assisted text was read and edited by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Assumptions and proofs. The standing assumptions used throughout are collected in Appendix B.1, and complete proofs of all stated results, including the path-correctness proposition and the closed-form selector, are given in Appendix B.

Method and algorithm. The practical estimator, including the endpoint-conditioned representation, the test-function construction, and the numerical safeguards, is specified in Appendix C, with the full procedure stated as Algorithm 1.

Experimental protocol. Appendix E reports the reproducibility summary, the data-generating law, the flow variants and backbone architectures, the baseline sweeps and their grids, the test-function families, and the numerical stabilisation used by the selector. Metric definitions are given in the same appendix. Complete per-configuration results, beyond the summary tables in Section 5, are tabulated in Appendix H.

Compute and cost. Hardware, software environment, wall-clock time, peak memory and backbone function evaluations for every configuration are reported in Appendix F.4.

Assets and licences. Licences and terms for all external code and data assets are summarised in Appendix G.

Code. Anonymised source code, configurations, and commands reproducing the reported experiments are provided in the supplementary material.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 2024.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making?, 2022.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. URL https://www.jmlr.org/papers/v26/23-1605.html.

L. Ambrosio, N. Gigli, and G. Savare. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics. ETH Zürich. Birkhäuser Basel, 2005. ISBN 9783764324285. URL https://books.google.co.il/books?id=HZqhWIq1-jgC.

Luigi Ambrosio. Transport equation and Cauchy problem for BV vector fields. *Inventiones Mathematicae*, 158(2):227–260, 2004. doi: 10.1007/s00222-004-0367-2.

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. In *International Conference on Machine Learning*, 2023.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, et al. Improving image generation with better captions, 2023.

Black Forest Labs. Announcing black forest labs, 2024. URL https://blackforestlabs.ai/announcing-black-forest-labs/. Launch post describing FLUX.1 model family, including FLUX.1 [dev] as a guidance-distilled model.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023a.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models, 2023b.

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector, 2024. URL https://arxiv.org/abs/2408.09000.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023.

Jian-Feng Cai, Haixia Liu, Zhengyi Su, and Chao Wang. Improving classifier-free guidance of flow matching via manifold projection, 2026. URL https://arxiv.org/abs/2601.21892.

Angela Castillo, Jonas Kohler, Juan C. Pérez, Juan Pablo Pérez, Albert Pumarola, Bernard Ghanem, Pablo Arbeláez, and Ali Thabet. Adaptive guidance: Training-free acceleration of conditional diffusion models, 2023. URL https://arxiv.org/abs/2312.12487.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, et al. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 2021.

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g7ohDlTITL.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. *CoRR*, abs/1806.07366, 2018. URL http://arxiv.org.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? a fine-grained analysis in a simple setting. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://openreview.net/forum?id=AdS3H8SaPi.

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. CFG++: Manifold-constrained classifier free guidance for diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=E77uvbOTtp.

Gabriele Corso, Hannes Stark, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations*, 2023.

Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Junnan Wang, et al. Emu: Enhancing image generation models using photogenic needles in a haystack, 2023.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021a. URL https://arxiv.org/abs/2105.05233.

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

R. J. DiPerna and P. L. Lions. Ordinary differential equations, transport theory and Sobolev spaces. *Inventiones mathematicae*, 98(3):511–547, 1989. doi: 10.1007/BF01393835. URL https://eudml.org/doc/143741.

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, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Zach Evans, CJ Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion, 2024.

Weichen Fan, Amber Yijia Zheng, Raymond A. Yeh, and Ziwei Liu. CFG-Zero*: Improved classifier-free guidance for flow matching models, 2025. URL https://arxiv.org/abs/2503.18886.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Alexandre Galashov, Ashwini Pokle, Arnaud Doucet, Arthur Gretton, Mauricio Delbracio, and Valentin De Bortoli. Learn to guide your diffusion model. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=l8XOk4ylBH.

Timur Garipov, Sebastiaan De Peuter, Ge Yang, Vikas Garg, Samuel Kaski, and Tommi Jaakkola. Compositional sculpting of iterative generative processes. In *Advances in Neural Information Processing Systems*, 2023.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://openreview.net/forum?id=X1QeUYBXke.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J. Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=o3BxOLoxm1.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://neurips.cc/virtual/2021/33972.

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

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

Yuhao Huang, Taos Transue, Shih-Hsin Wang, William Feldman, Hong Zhang, and Bao Wang. Improving flow matching by aligning flow divergence, 2026. URL https://arxiv.org/abs/2602.00869.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2021.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022.

Cheng Jin, Zhenyu Xiao, Chutao Liu, and Yuantao Gu. Angle domain guidance: Latent diffusion requires rotation rather than extrapolation. In *International Conference on Machine Learning*, 2025.

Cheng Jin, Qitan Shi, and Yuantao Gu. Stage-wise dynamics of classifier-free guidance in diffusion models. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fP0s1TEow3.

Kacper Kapuśniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael M. Bronstein, Avishek Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://openreview.net/forum?id=fE3RqiF4Nx.

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 35*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/a98846e9d9cc01cfb87eb694d946ce6b-Abstract-Conference.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015. URL https://arxiv.org/abs/1412.6980.

Félix Koulischer, Florian Handke, Johannes Deleu, Thomas Demeester, and Luca Ambrogioni. Feedback guidance of diffusion models. In *Advances in Neural Information Processing Systems 38*, 2025. URL https://openreview.net/forum?id=8ySOcf7UpM.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural Information Processing Systems 37*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/dd540e1c8d26687d56d296e64d35949f-Paper-Conference.pdf.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, et al. Voicebox: Text-guided multilingual universal speech generation at scale, 2023.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. doi: 10.1109/5.726791.

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

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022.

Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particle solutions of fokker–planck equations through gradient–log–density estimation. *Entropy*, 22(8):802, 2020.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. URL https://github.com/toshas/torch-fidelity. Version: 0.4.0, DOI: 10.5281/zenodo.3786539.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://neurips.cc.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. URL https://arxiv.org/abs/1709.07871.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

Kedar Potdar, Taher S Pardawala, and Chinmay D Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175(4):7–9, 2017. doi: 10.5120/ijca2017915495.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

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

Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. CADS: Unleashing the diversity of diffusion models through condition-annealed sampling, 2023. URL https://arxiv.org/abs/2310.17347.

Seyedmorteza Sadat, Otmar Hilliges, and Romann M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e2ONKX6qzJ.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, et al. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

Shreshth Saini, Shashank Gupta, and Alan C. Bovik. Rectified-cfg++ for flow based models, 2025. URL https://arxiv.org/abs/2510.07631.

Filippo Santambrogio. Optimal transport for applied mathematicians: Calculus of variations, pdes, and modeling. 2015. URL https://api.semanticscholar.org/CorpusID:124181096.

Uriel Singer, Adam Polyak, Thomas Hayes, Xiaoyue Yin, Jie An, et al. Make-a-video: Text-to-video generation without text-video data, 2022.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

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

Yang Song, 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.

StabilityAI. Introducing stable diffusion 3.5, 2024. URL https://stability.ai/news-updates/introducing-stable-diffusion-3-5. Accessed: 2026-04-17.

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. URL https://openreview.net/forum?id=CD9Snc73AW.

Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer, 2009.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, et al. Neural codec language models are zero-shot text to speech synthesizers, 2023.

Kaibo Wang, Jianda Mao, Tong Wu, and Yang Xiang. Towards a golden classifier-free guidance path via foresight fixed point iterations. In *Advances in Neural Information Processing Systems 38*, 2025. URL https://openreview.net/forum?id=yf8O4xEB4T.

Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernandez Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=SUMtDJqicd.

Akihisa Watanabe, Qing Yu, Edgar Simo-Serra, and Kent Fujiwara. Projflow: Projection sampling with flow matching for zero-shot exact spatial motion control, 2026. URL https://arxiv.org/abs/2602.22742.

Mengfei Xia, Yujun Shen, Changsong Lei, Yu Zhou, Deli Zhao, Ran Yi, Wenping Wang, and Yong-Jin Liu. Towards more accurate diffusion model acceleration with a timestep tuner. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Xia_Towards_More_Accurate_Diffusion_Model_Acceleration_with_A_Timestep_Tuner_CVPR_2024_paper.html.

Shai Yehezkel, Omer Dahary, Andrey Voynov, and Daniel Cohen-Or. Navigating with annealing guidance scale in diffusion space. In *SIGGRAPH Asia 2025 Conference Papers*, 2025. doi: 10.1145/3757377.3763830. URL https://dl.acm.org/doi/10.1145/3757377.3763830.

Jason Yim et al. Se(3) diffusion model with application to protein backbone generation, 2023.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky T. Q. Chen. Guided flows for generative modeling and decision making, 2023. URL https://arxiv.org/abs/2311.13443.

A Additional background: Unifying Gaussian Flow-Matching and Diffusion Probability-Flows

This section establishes the formal equivalence between the velocity fields used in flow-matching and the score-based representations in diffusion ODEs referred to in Section 3.

Steering a generative process toward a condition y can be understood through the lens of a time-dependent classifier $p_t(y \mid x)$ (Dhariwal & Nichol, 2021a). To avoid training such a classifier explicitly, classifier-free guidance (Ho & Salimans, 2021) uses the Bayes' rule decomposition

$$\nabla_x \log p_t(x \mid y) = \nabla_x \log p_t(x) + \nabla_x \log p_t(y \mid x), \tag{13}$$

so the classifier gradient is approximated by the difference between the conditional and unconditional score fields. The guided score is then used inside the diffusion sampling dynamics. Equivalently, the same marginal probability path is generated by a deterministic probability-flow ODE whose vector field depends on the score, so applying CFG to the score induces a corresponding guided probability-flow vector field. This is the bridge from score-based diffusion to flow matching (Maoutsa et al., 2020; Lipman et al., 2023; Song et al., 2021b).

Consider an endpoint-conditioned Gaussian path

$$p_t(x \mid x_1, y) = \mathcal{N}\big(x \mid \alpha_t x_1, \sigma_t^2 \mathbf{I}\big), \qquad t \in [0,1], \tag{14}$$

with differentiable scalar schedules α_t and $\sigma_t > 0$. Its endpoint-conditioned score is

$$\nabla_x \log p_t(x \mid x_1, y) = -\frac{x - \alpha_t x_1}{\sigma_t^2}. \tag{15}$$

The corresponding endpoint-conditioned velocity field is

$$u_t(x \mid x_1, y) = \dot{\alpha}_t x_1 + \frac{\dot{\sigma}_t}{\sigma_t}\big(x - \alpha_t x_1\big). \tag{16}$$

Equivalently, when $\alpha_t \neq 0$,

$$u_t(x \mid x_1, y) = \frac{\dot{\alpha}_t}{\alpha_t} x - \sigma_t^2 \left(\frac{\dot{\sigma}_t}{\sigma_t} - \frac{\dot{\alpha}_t}{\alpha_t}\right) \nabla_x \log p_t(x \mid x_1, y). \tag{17}$$

Thus Gaussian endpoint-conditioned flow-matching paths admit a drift-plus-score representation.

Crucially, because the relationship between the velocity field and the score is affine in x and the score term, this identity extends directly to the marginal laws via the linearity of expectation (Albergo et al., 2025; Zheng et al., 2023). Specifically, by integrating the conditioned field over the posterior $p(x_1 \mid x, y)$, we obtain the marginal conditional field:

$$u_t(x \mid y) = \frac{\dot{\alpha}_t}{\alpha_t} x - \sigma_t^2 \left(\frac{\dot{\sigma}_t}{\sigma_t} - \frac{\dot{\alpha}_t}{\alpha_t}\right) \nabla_x \log p_t(x \mid y). \tag{18}$$

The same identity holds for the unconditional field $u_t(x)$ by marginalizing over y.

This resembles the deterministic probability-flow ODE notation used for score-based diffusion, where a marginal path is generated by a field of the form

$$\frac{\mathrm{d}x_t}{\mathrm{d}t} = f(x_t, t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x_t \mid y). \tag{19}$$

B Proofs

This appendix collects the proof details used in Section 4. All statements are written for a fixed condition y. We distinguish the exact continuous-time objects from the empirical estimators used by the practical algorithm.

B.1 STANDING ASSUMPTIONS

We use the following assumptions throughout the proofs.

Assumption B.1 (Regularity of conditional paths)**.** For $t \in [0,1]$, the endpoint-conditioned path $p_t(\cdot \mid x_1, y)$ and its vector field $u_t(\cdot \mid x_1, y)$ satisfy the continuity equation

$$\partial_t p_t(x \mid x_1, y) + \operatorname{div}\big(p_t(x \mid x_1, y) u_t(x \mid x_1, y)\big) = 0 \tag{20}$$

in the weak sense. Moreover, the integrability and regularity conditions needed to exchange differentiation, divergence, and integration over x_1 hold.

Assumption B.2 (Endpoint posterior weights)**.** The marginal conditional density (6) is finite and positive on the region where the generated rollout law is evaluated. We define

$$\pi_t(x_1 \mid x, y) := \frac{p_t(x \mid x_1, y) q(x_1 \mid y)}{p_t(x \mid y)}. \tag{21}$$

Outside the support where $p_t(x \mid y) > 0$, the value of $\pi_t(\cdot \mid x, y)$ is immaterial for the objectives below.

Assumption B.3 (Generated rollout law)**.** For every admissible guidance schedule, the generated rollout law $\hat{p}_t(\cdot \mid y; \omega_{[0,t)})$ is weakly continuous in t and satisfies the weak continuity equation driven by the guided field $v_t^\theta(\cdot \mid y; \omega_t)$. The law at time t depends only on the already committed past schedule $\omega_{[0,t)}$.

Assumption B.4 (Uniqueness)**.** The exact conditional path $p_t(\cdot \mid y)$ is the unique weak solution of the continuity equation driven by the exact marginal conditional field $u_t(\cdot \mid y)$, with initial law p_0. Specifically, following the theory of DiPerna & Lions (1989) and Ambrosio (2004), we assume:

1. $u_t(\cdot \mid y) \in L^1([0,1]; BV_{loc}(\mathbb{R}^d))$ with at most linear growth in x.
2. $\operatorname{div} u_t(\cdot \mid y) \in L^1([0,1]; L^\infty(\mathbb{R}^d))$.
3. $p_t(\cdot \mid y) \in L^\infty([0,1] \times \mathbb{R}^d)$.

B.2 ENDPOINT-CONDITIONED PATHS IMPLY THE MARGINAL CONTINUITY EQUATION

Proposition B.1 (Marginal conditional continuity equation)**.** *Under Assumptions B.1 and B.2,*

$$\partial_t p_t(x \mid y) + \operatorname{div}\big(p_t(x \mid y) u_t(x \mid y)\big) = 0. \tag{22}$$

Proof. By differentiation under the endpoint integral,

$$\partial_t p_t(x \mid y) = \int_{\mathbb{R}^d} \partial_t p_t(x \mid x_1, y) q(x_1 \mid y)\, \mathrm{d}x_1. \tag{23}$$

Using the endpoint-conditioned continuity equation,

$$\partial_t p_t(x \mid y) = -\int_{\mathbb{R}^d} \operatorname{div}\big(p_t(x \mid x_1, y) u_t(x \mid x_1, y)\big) q(x_1 \mid y)\, \mathrm{d}x_1. \tag{24}$$

The divergence acts on x, so by the regularity assumption it may be moved outside the endpoint integral:

$$\partial_t p_t(x \mid y) = -\operatorname{div}\left(\int_{\mathbb{R}^d} p_t(x \mid x_1, y) u_t(x \mid x_1, y) q(x_1 \mid y)\, \mathrm{d}x_1\right). \tag{25}$$

Using the definition of $\pi_t(x_1 \mid x, y)$,

$$\int_{\mathbb{R}^d} p_t(x \mid x_1, y) u_t(x \mid x_1, y) q(x_1 \mid y)\, \mathrm{d}x_1 = p_t(x \mid y) u_t(x \mid y). \tag{26}$$

This recovers the marginal continuity equation described in the Flow Matching paragraph of Section 3, a result previously established by Lipman et al. (2023). □

B.3 WEAK FORM USED BY THE LOCAL OBJECTIVE

Proposition B.2 (Weak continuity identity). *Let ρ_t and b_t satisfy*

$$\partial_t \rho_t + \operatorname{div}(\rho_t b_t) = 0 \tag{27}$$

in the distributional sense on $[0,1] \times \mathbb{R}^d$. Then, for every $\psi \in C_c^1(\mathbb{R}^d)$, the map

$$t \mapsto \int_{\mathbb{R}^d} \psi(x)\rho_t(x)\, \mathrm{d}x \tag{28}$$

is absolutely continuous and, for almost every t,

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\mathbb{R}^d} \psi(x)\rho_t(x)\, \mathrm{d}x = \int_{\mathbb{R}^d} \nabla\psi(x) \cdot b_t(x)\rho_t(x)\, \mathrm{d}x. \tag{29}$$

Proof. Test the distributional continuity equation with $\varphi(t,x) = \xi(t)\psi(x)$, where $\xi \in C_c^1((0,1))$ and $\psi \in C_c^1(\mathbb{R}^d)$. This yields

$$\int_0^1 \xi'(t) \left(\int_{\mathbb{R}^d} \psi(x)\rho_t(x)\, \mathrm{d}x \right) \mathrm{d}t + \int_0^1 \xi(t) \left(\int_{\mathbb{R}^d} \nabla\psi(x) \cdot b_t(x)\rho_t(x)\, \mathrm{d}x \right) \mathrm{d}t = 0. \tag{30}$$

Hence the scalar map $t \mapsto \int \psi\, \rho_t$ has weak derivative $t \mapsto \int \nabla\psi \cdot b_t\, \rho_t$, giving the desired absolutely continuous representative and pointwise identity for almost every t. For a complete treatment, see Ambrosio et al. (2005); Santambrogio (2015). □

B.4 IDEAL WEAK EQUIVALENCE IMPLIES PATH CORRECTNESS

Proof of Proposition 4.1. Fix $\psi \in C_c^\infty(\mathbb{R}^d)$. By Assumption B.3, the generated law satisfies

$$\frac{\mathrm{d}}{\mathrm{d}s} \int_{\mathbb{R}^d} \psi(x)\hat{p}_s(x)\, \mathrm{d}x = \int_{\mathbb{R}^d} \nabla\psi(x) \cdot v_s^\theta(x \mid y; \omega_s)\hat{p}_s(x)\, \mathrm{d}x \tag{31}$$

for almost every $s \in [0,t]$. The assumed weak equivalence in Proposition 4.1 replaces the right-hand side by

$$\int_{\mathbb{R}^d} \nabla\psi(x) \cdot u_s(x \mid y)\hat{p}_s(x)\, \mathrm{d}x. \tag{32}$$

Therefore $\hat{p}_s$ is a weak solution of the continuity equation driven by $u_s(\cdot \mid y)$. Since $\hat{p}_0 = p_0$, Assumption B.4 implies

$$\hat{p}_s(\cdot \mid y; \omega_{[0,s)}) = p_s(\cdot \mid y), \qquad s \in [0,t]. \tag{33}$$

□

B.5 EXACT LOCAL WEAK-FORM OBJECTIVE

The body defines the local objective at solver time t_i, weighted by the current generated rollout law $\hat{p}_{t_i}(\cdot \mid y; \vec{\omega}_i)$, where $\vec{\omega}_i = (\omega_0, \ldots, \omega_{i-1})$ is fixed before choosing ω_i. For a test function ψ, define the signed weak-form mismatch

$$G_i(\omega_i \mid y; \vec{\omega}_i, \psi) := \int_{\mathbb{R}^d} \Big(u_{t_i}(x \mid y) - v_{t_i}^\theta(x \mid y; \omega_i) \Big) \cdot \nabla\psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x. \tag{34}$$

The exact local objective is

$$\mathcal{L}_i(\omega_i \mid y; \vec{\omega}_i, \psi) := \big| G_i(\omega_i \mid y; \vec{\omega}_i, \psi) \big|^2. \tag{35}$$

This objective is on-policy: its weighting distribution is the rollout produced by the already committed past guidance values, not the ideal path.

B.6 Endpoint-conditioned representation of the exact objective

Proposition B.3 (Endpoint-conditioned objective)**.** *Under Assumptions B.1 and B.2, the signed quantity in* (35) *admits the endpoint-conditioned form*

$$G_i(\omega_i \mid y; \vec{\omega}_i, \psi) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \Big(u_{t_i}(x \mid x_1, y) - v^{\theta}_{t_i}(x \mid y; \omega_i) \Big) \cdot \nabla \psi(x) \times \pi_{t_i}(x_1 \mid x, y)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x_1\, \mathrm{d}x. \tag{36}$$

Consequently, $\mathcal{L}_i$ can be evaluated from endpoint-conditioned velocities and posterior endpoint weights, which proves Theorem 4.1.

Proof. We proceed in four steps.

(i) Posterior normalization. By Assumption B.2, $\pi_{t_i}(x_1 \mid x, y) = p_{t_i}(x \mid x_1, y) q(x_1 \mid y) / p_{t_i}(x \mid y)$ is a probability density in x_1 wherever $p_{t_i}(x \mid y) > 0$, so $\int_{\mathbb{R}^d} \pi_{t_i}(x_1 \mid x, y)\, \mathrm{d}x_1 = 1$.

(ii) Marginal field as a posterior average. Eq. (5) reads $u_{t_i}(x \mid y) = \int_{\mathbb{R}^d} u_{t_i}(x \mid x_1, y)\, \pi_{t_i}(x_1 \mid x, y)\, \mathrm{d}x_1$.

(iii) The guided field carries no endpoint dependence. $v^{\theta}_{t_i}(x \mid y; \omega_i)$ does not depend on x_1, so by (i) it may be written as $\int_{\mathbb{R}^d} v^{\theta}_{t_i}(x \mid y; \omega_i)\, \pi_{t_i}(x_1 \mid x, y)\, \mathrm{d}x_1$.

(iv) Combine. Substituting (ii) and (iii) into the definition of G_i and merging the two x_1-integrals - permitted by Assumption B.1 - gives the displayed formula. Squaring the signed quantity gives the endpoint-conditioned representation of $\mathcal{L}_i$.

The identity is exact at the population level. In the estimator of Appendix C, both π_{t_i} and the outer integral are replaced by Monte Carlo averages. □

B.7 Affine form under scalar CFG

Proposition B.4 (Quadratic local objective under scalar CFG)**.** *Assume scalar CFG:*

$$v^{\theta}_t(x \mid y; \omega) = v^{\theta}_t(x \mid \varnothing) + \omega \Big(v^{\theta}_t(x \mid y) - v^{\theta}_t(x \mid \varnothing) \Big). \tag{37}$$

For fixed $y, \vec{\omega}_i, \psi$, define

$$A_i(\psi) := \int_{\mathbb{R}^d} \Big(u_{t_i}(x \mid y) - v^{\theta}_{t_i}(x \mid \varnothing) \Big) \cdot \nabla \psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x \tag{38}$$

and

$$B_i(\psi) := \int_{\mathbb{R}^d} \Big(v^{\theta}_{t_i}(x \mid y) - v^{\theta}_{t_i}(x \mid \varnothing) \Big) \cdot \nabla \psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x. \tag{39}$$

Then

$$\mathcal{L}_i(\omega_i \mid y; \vec{\omega}_i, \psi) = \big| A_i(\psi) - \omega_i B_i(\psi) \big|^2. \tag{40}$$

In particular, the exact local objective is quadratic in ω_i.

Proof. Insert the scalar CFG field into G_i:

$$G_i(\omega_i \mid y; \vec{\omega}_i, \psi) = \int_{\mathbb{R}^d} \Big(u_{t_i}(x \mid y) - v^{\theta}_{t_i}(x \mid \varnothing) - \omega_i \big(v^{\theta}_{t_i}(x \mid y) - v^{\theta}_{t_i}(x \mid \varnothing) \big) \Big) \cdot \nabla \psi(x)\, \hat{p}_{t_i}(x \mid y; \vec{\omega}_i)\, \mathrm{d}x. \tag{41}$$

Collecting the terms independent of ω_i and those multiplying ω_i gives $G_i = A_i - \omega_i B_i$. Squaring gives the claim. □

B.8 EXACT LOCAL SELECTOR

Proof of Corollary 4.1. By Proposition B.4,

$$\mathcal{L}_i(\omega_i) = \left|A_i - \omega_i B_i\right|^2. \tag{42}$$

For a single test function, $A_i, B_i \in \mathbb{R}$, so

$$\mathcal{L}_i(\omega_i) = A_i^2 - 2\omega_i A_i B_i + \omega_i^2 B_i^2. \tag{43}$$

If $B_i \neq 0$, differentiating and setting the derivative to zero gives

$$\omega_i^\star = \frac{A_i B_i}{B_i^2} = \frac{A_i}{B_i}. \tag{44}$$

Substituting the definitions of A_i and B_i gives Eq. (11) in the body.

If $B_i = 0$, then the objective is independent of ω_i. Hence every admissible guidance value is optimal. In practical use, this degenerate case is handled by a denominator floor as specified in E.7. □

B.9 FROM THE LOCAL OBJECTIVE TO THE INTERVAL-WISE IMPLEMENTATION

The selector in Corollary 4.1 is written at a fixed time t_i. In numerical sampling, the guidance value is held fixed over the solver interval $[t_i, t_{i+1})$. We therefore use the left-endpoint discretization of the local weak-form objective on each interval.

C PRACTICAL ALGORITHM

The algorithm can be used in two modes: *online selection*, where the guidance value is selected during the same sampling run, and *offline schedule fitting*, where a schedule is fitted once and then reused for later inference with the same backbone, solver, grid, and condition.

Here T is the number of solver intervals, M is the number of endpoint samples, N is the number of rollout particles, and L is the number of weak test functions.

Posterior weights. At interval i, suppose the current rollout particles are

$$x_i^{(n)} \sim \hat{p}_{t_i}(\cdot \mid y; \vec{\omega}_i), \qquad n = 1, \dots, N, \tag{45}$$

where $\vec{\omega}_i = (\omega_0, \dots, \omega_{i-1})$ is the already committed schedule. Draw endpoint samples

$$x_{1,i}^{(m)} \sim q(\cdot \mid y), \qquad m = 1, \dots, M. \tag{46}$$

For each rollout particle $x_i^{(n)}$, define the empirical posterior weights

$$\pi_i^{(n,m)} := \frac{p_{t_i}\left(x_i^{(n)} \mid x_{1,i}^{(m)}, y\right)}{\sum_{r=1}^{M} p_{t_i}\left(x_i^{(n)} \mid x_{1,i}^{(r)}, y\right)}. \tag{47}$$

Multi-test coefficient estimator. Let

$$\Psi = \{\psi_\ell\}_{\ell=1}^{L} \tag{48}$$

be the chosen test family, and let $\Delta t_i = t_{i+1} - t_i$. For each ψ_ℓ, define

$$\widetilde{A}_i^{(\ell)} := \Delta t_i \frac{1}{N} \sum_{n=1}^{N} \left[\sum_{m=1}^{M} \pi_i^{(n,m)} \Big(u_{t_i}(x_i^{(n)} \mid x_{1,i}^{(m)}, y) - v_{t_i}^\theta(x_i^{(n)} \mid \varnothing) \Big) \right] \cdot \nabla \psi_\ell(x_i^{(n)}), \tag{49}$$

and

$$\widetilde{B}_i^{(\ell)} := \Delta t_i \frac{1}{N} \sum_{n=1}^{N} \Big(v_{t_i}^\theta(x_i^{(n)} \mid y) - v_{t_i}^\theta(x_i^{(n)} \mid \varnothing) \Big) \cdot \nabla \psi_\ell(x_i^{(n)}). \tag{50}$$

Algorithm 1 Practical interval-wise guidance selection

Input: condition y; frozen fields $v_t^\theta(\cdot \mid y)$ and $v_t^\theta(\cdot \mid \varnothing)$; endpoint-conditioned field $u_t(\cdot \mid x_1, y)$; endpoint-conditioned density $p_t(\cdot \mid x_1, y)$; solver grid $0 = t_0 < \cdots < t_T = 1$; rollout budget N; endpoint budget M; test family $\Psi = \{\psi_\ell\}_{\ell=1}^L$; min and max constraint $\omega_{\min}, \omega_{\max}$; mode $\in \{\text{online}, \text{offline}\}$; random seeds.
Draw initial rollout particles

$$x_0^{(n)} \sim p_0, \qquad n = 1, \dots, N. \tag{54}$$

Initialize the selected schedule as an empty list.
for $i = 0, \dots, T-1$ **do**
Set $\Delta t_i = t_{i+1} - t_i$.
Draw endpoint samples

$$x_{1,i}^{(m)} \sim q(\cdot \mid y), \qquad m = 1, \dots, M. \tag{55}$$

Compute empirical posterior weights $\{\pi_i^{(n,m)}\}_{n,m}$ using Eq. (47).
Compute the multi-test coefficients $\widetilde{A}_i, \widetilde{B}_i$ using Eqs. (49)-(50).
Select the interval guidance value using the stabilized quotient of Appendix E.7, with $\widetilde{\beta}_i = \left\|\widetilde{B}_i\right\|_2^2$ and $\widetilde{\beta}_i^\star = \max_{0 \le j < i} \widetilde{\beta}_j$:

$$\widehat{\omega}_i \leftarrow \min\left\{\omega_{\max}, \max\left\{\omega_{\min}, \frac{\left\langle \widetilde{A}_i, \widetilde{B}_i \right\rangle}{\max\left\{\widetilde{\beta}_i,\, \eta\, \widetilde{\beta}_i^\star\right\}}\right\}\right\}. \tag{56}$$

Append $\widehat{\omega}_i$ to the schedule.
Advance all rollout particles by one solver step:

$$x_{i+1}^{(n)} \leftarrow \mathrm{ODEStep}\left(x_i^{(n)}, t_i, t_{i+1}; v^\theta(\cdot \mid y; \widehat{\omega}_i)\right), \qquad n = 1, \dots, N. \tag{57}$$

end for
if mode is online **then**
Return: selected schedule $(\widehat{\omega}_0, \dots, \widehat{\omega}_{T-1})$ and generated particles $\{x_T^{(n)}\}_{n=1}^N$.
else
Return: fitted schedule $(\widehat{\omega}_0, \dots, \widehat{\omega}_{T-1})$.
end if

Stack these coefficients as

$$\widetilde{A}_i = \left(\widetilde{A}_i^{(1)}, \dots, \widetilde{A}_i^{(L)}\right), \qquad \widetilde{B}_i = \left(\widetilde{B}_i^{(1)}, \dots, \widetilde{B}_i^{(L)}\right). \tag{51}$$

The empirical interval-wise objective is

$$\widetilde{\mathcal{L}}_i(\omega_i \mid y; \vec{\omega}_i, \Psi) = \frac{1}{L}\left\|\widetilde{A}_i - \omega_i \widetilde{B}_i\right\|_2^2. \tag{52}$$

The factor $1/L$ does not affect the minimizer. The closed-form multi-test selector is

$$\widehat{\omega}_i = \frac{\left\langle \widetilde{A}_i, \widetilde{B}_i \right\rangle}{\left\|\widetilde{B}_i\right\|_2^2} \tag{53}$$

Offline reuse. In offline mode, Algorithm 1 is run once using fitting seeds and representative initial particles. The resulting schedule is stored as a piecewise-constant function on the solver grid. Later inference runs draw fresh initial particles and use the stored values $(\widehat{\omega}_0, \dots, \widehat{\omega}_{T-1})$ directly, without endpoint sampling or posterior-weight computation.

D CONDITIONAL PATHS USED FOR EXPERIMENTAL VALIDATION

This appendix defines the endpoint-conditioned paths and velocity labels used for experimental validation. Throughout, $t \in [0, 1]$ denotes generation time: $t = 0$ corresponds to the Gaussian

source and $t = 1$ corresponds to the conditional endpoint law. We do not use a separate diffusion time variable. For a class label y, endpoint samples are drawn from $x_1 \sim q(\cdot \mid y)$, and source samples are drawn from $x_0 \sim \mathcal{N}(0, \mathbf{I})$.

Affine Gaussian path template. Several paths below are special cases of

$$x_t = a_t x_0 + b_t x_1, \qquad x_0 \sim \mathcal{N}(0, \mathbf{I}), \tag{58}$$

where $a_t > 0$ for $t < 1$. The endpoint-conditioned law is

$$p_t(x \mid x_1, y) = \mathcal{N}\big(x \mid b_t x_1, a_t^2 \mathbf{I}\big)\,. \tag{59}$$

The associated endpoint-conditioned velocity field is

$$u_t(x \mid x_1, y) = \dot{b}_t x_1 + \frac{\dot{a}_t}{a_t}\,(x - b_t x_1)\,, \qquad t < 1. \tag{60}$$

Equivalently, along a sampled path $x_t = a_t x_0 + b_t x_1$,

$$u_t(x_t \mid x_1, y) = \dot{a}_t x_0 + \dot{b}_t x_1. \tag{61}$$

The along-sample label is the quantity used whenever the paired source sample is available.

Rectified flow (RF) (Liu et al., 2022). The RF path uses the deterministic straight-line interpolation

$$x_t = (1 - t)x_0 + t x_1. \tag{62}$$

Thus $a_t = 1 - t$ and $b_t = t$. The along-sample velocity label is

$$u_t(x_t \mid x_1, y) = x_1 - x_0. \tag{63}$$

Optimal-transport Gaussian path (OT) (Lipman et al., 2023). The OT flow-matching path uses

$$x_t = \big(1 - (1 - \sigma_{\min})t\big)x_0 + t x_1, \qquad \sigma_{\min} > 0. \tag{64}$$

Thus

$$a_t = 1 - (1 - \sigma_{\min})t, \qquad b_t = t. \tag{65}$$

The endpoint-conditioned law is

$$p_t(x \mid x_1, y) = \mathcal{N}\Big(x \mid t x_1, \big(1 - (1 - \sigma_{\min})t\big)^2 \mathbf{I}\Big)\,, \tag{66}$$

and the explicit conditional velocity is

$$u_t(x \mid x_1, y) = \frac{x_1 - (1 - \sigma_{\min})x}{1 - (1 - \sigma_{\min})t}. \tag{67}$$

Along sampled paths, the velocity label is

$$u_t(x_t \mid x_1, y) = x_1 - (1 - \sigma_{\min})x_0. \tag{68}$$

Independent conditional flow matching (I-CFM) (Tong et al., 2024). For I-CFM, the conditioning variable is the pair

$$z = (x_0, x_1), \qquad x_0 \sim \mathcal{N}(0, \mathbf{I}), \qquad x_1 \sim q(\cdot \mid y), \tag{69}$$

sampled independently. The pair-conditioned path is

$$p_t(x \mid z, y) = \mathcal{N}\big(x \mid (1 - t)x_0 + t x_1, \sigma_{\mathrm{icfm}}^2 \mathbf{I}\big)\,, \tag{70}$$

or equivalently

$$x_t = (1 - t)x_0 + t x_1 + \sigma_{\mathrm{icfm}}\varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{71}$$

Since the standard deviation is constant in time, the velocity label is

$$u_t(x \mid z, y) = x_1 - x_0. \tag{72}$$

Table 3: Conditional paths used for experimental validation. The time variable is always generation time $t \in [0, 1]$, from prior to endpoint.

Variant	Conditioning	Path sample x_t	Velocity label
RF (Liu et al., 2022)	x_1, y	$(1-t)x_0 + tx_1$	$x_1 - x_0$
OT (Lipman et al., 2023)	x_1, y	$\big(1-(1-\sigma_{\min})t\big)x_0 + tx_1$	$x_1 - (1-\sigma_{\min})x_0$
I-CFM (Tong et al., 2024)	$z = (x_0, x_1), y$	$(1-t)x_0 + tx_1 + \sigma_{\text{icfm}}\varepsilon$	$x_1 - x_0$
VP (Song et al., 2021b)	x_1, y	$\bar{\alpha}_t x_1 + \sqrt{1-\bar{\alpha}_t^2}\, x_0$	$\partial_t x_t$

Table 4: Reproducibility parameters for the Gaussian-mixture experimental validation.

Item	Value / convention
Endpoint law	Eq. (81)
Source law	$\mathcal{N}(0, \mathbf{I})$
Dimension	$d = 2$
Class schedules	fitted separately for each $y \in \mathcal{Y}$
Main solver	fixed-grid Euler
Solver grid	$0 = t_0 < t_1 < \cdots < t_T = 1$
Reference fitting grid	$T =$
Generated samples per inference seed	N_{gen}
Reference samples per inference seed	N_{ref}
Rollout particles for fitting	N
Endpoint samples for fitting	M
Number of test functions	L
Guidance clipping interval	$[\omega_{\min}, \omega_{\max}]$
Seed aggregation	mean $\pm$ sample std over inference seeds

Variance-preserving probability-flow path (VP) (Song et al., 2021b). For the VP variant, we write the signal coefficient directly in generation time as $\bar{\alpha}_t$, with $\bar{\alpha}_0 \approx 0$ and $\bar{\alpha}_1 = 1$. The endpoint-conditioned law is

$$p_t(x \mid x_1, y) = \mathcal{N}\big(x \mid \bar{\alpha}_t x_1, \big(1-\bar{\alpha}_t^2\big)\mathbf{I}\big)\,. \tag{73}$$

Applying Eq. (60) with $a_t = \sqrt{1-\bar{\alpha}_t^2}$ and $b_t = \bar{\alpha}_t$ gives

$$u_t(x \mid x_1, y) = \frac{\dot{\bar{\alpha}}_t}{1-\bar{\alpha}_t^2}\,(x_1 - \bar{\alpha}_t x)\,, \qquad t < 1. \tag{74}$$

The VP field is evaluated only at solver times $t_i < 1$. Endpoint quantities at $t = 1$ are evaluated from the target law rather than by querying Eq. (74).

E Experimental details

Here we provide the details for the experimental validation in Section 5. All schedules are fitted per class unless explicitly stated otherwise. All methods in a comparison use the same endpoint law, solver grid, initial latents, and inference seeds. We report mean $\pm$ sample standard deviation, where $\text{std} = \sqrt{\Big(\sum_{i=1}^{|\text{seeds}|}(\text{metric}_i - \text{mean})^2\Big)/(|\text{seeds}| - 1)}$.

E.1 Reproducibility summary

Unless otherwise stated, the experimental validation uses

$$T = 200, \qquad M = 2^{14}, \qquad N = 2^{14}, \qquad L = 2^{12}, \qquad \omega_{\min} = 1, \qquad \omega_{\max} = \infty. \tag{75}$$

The default test family contains 2^{11} linear tests and 2^{11} quadratic tests. We use three fitting seeds and three inference seeds, with matched prior latents across compared methods within each inference seed. The denominator stabilizer is specified in Appendix E.7.

Randomness is split into fitting randomness and inference randomness. Fitting seeds determine the rollout particles, endpoint samples, and test-function draws used to estimate the schedule. Inference

seeds determine the evaluation latents. Within each inference seed and class, all compared methods receive the same initial latents x_0.

For a piecewise-constant schedule $\bar{\omega}_i$ on $[t_i, t_{i+1})$, sampling uses

$$x_{i+1} = x_i + (t_{i+1} - t_i) v^{\theta}_{t_i}(x_i \mid y; \bar{\omega}_i). \tag{76}$$

For FM paths, the grid ends at $t = 1$. For VP paths, the grid ends at $1 - 10^{-5}$. All compared methods use matched initial latents within each class and inference seed.

E.2 Metric definitions

Let $A = \{\hat{x}_i\}_{i=1}^{N_A}$ denote generated samples and $B = \{x_j\}_{j=1}^{N_B}$ denote reference samples. Metrics are computed per class and then averaged using the class prior ρ_y. For every reported table or curve, the error bar is the sample standard deviation over inference seeds.

Gaussian-fit KL. Let $(\hat{\mu}, \hat{\Sigma})$ be the empirical mean and covariance of A, and let (μ, Σ) be the reference mean and covariance. We report

$$\mathrm{KL}(A \| B) = \frac{1}{2}\left[\mathrm{tr}(\Sigma^{-1}\hat{\Sigma}) + (\mu - \hat{\mu})^{\top}\Sigma^{-1}(\mu - \hat{\mu}) - d + \log\frac{\det\Sigma}{\det\hat{\Sigma}}\right]. \tag{77}$$

Gaussian W_2. We report the Gaussian Wasserstein distance

$$\mathrm{W}_2^2(A, B) = \|\hat{\mu} - \mu\|_2^2 + \mathrm{tr}\left(\hat{\Sigma} + \Sigma - 2(\Sigma^{1/2}\hat{\Sigma}\Sigma^{1/2})^{1/2}\right). \tag{78}$$

MMD-RBF. Let $\{\hat{x}^{(n)}\}_{n=1}^{N_{\mathrm{gen}}}$ denote the generated endpoint samples for a fixed inference seed, and let $\{x^{(m)}\}_{m=1}^{N_{\mathrm{ref}}}$ denote the corresponding reference endpoint samples. We report

$$\mathrm{MMD}^2_{\mathrm{RBF}} = \frac{\sum_{n,n'=1}^{N_{\mathrm{gen}}} k(\hat{x}^{(n)}, \hat{x}^{(n')})}{N_{\mathrm{gen}}^2} + \frac{\sum_{m,m'=1}^{N_{\mathrm{ref}}} k(x^{(m)}, x^{(m')})}{N_{\mathrm{ref}}^2} - \frac{2\sum_{n=1}^{N_{\mathrm{gen}}}\sum_{m=1}^{N_{\mathrm{ref}}} k(\hat{x}^{(n)}, x^{(m)})}{N_{\mathrm{gen}} N_{\mathrm{ref}}}, \tag{79}$$

where k is the RBF kernel. The default kernel is the multi-bandwidth RBF kernel

$$k(u, v) = \frac{1}{3}\sum_{\alpha \in \{0.25, 0.5, 1.0\}} \exp\left(-\frac{\|u - v\|_2^2}{2(\alpha\sigma_y)^2}\right), \tag{80}$$

where σ_y is computed once per class from reference samples using the median heuristic. The same bandwidth is used for all methods, guidance scales, and inference seeds within a class.

E.3 Data-generating law

We use the analytic two-class Gaussian-mixture. The endpoint law is

$$q(x_1) = \sum_{y \in \{0,1\}} \rho_y q(x_1 \mid y), \qquad q(x_1 \mid y) = \mathcal{N}(\mu_y, \Sigma_y), \tag{81}$$

with

$$\rho_0 = \rho_1 = \frac{1}{2}, \qquad \mu_0 = (-2, 0), \qquad \mu_1 = (2, 0), \qquad \Sigma_0 = \Sigma_1 = I_2. \tag{82}$$

The source law is $p_0 = \mathcal{N}(0, I_2)$, independent of y. Samples are drawn directly from the analytic law. No saved dataset files or train/validation splits are used. Fitting and evaluation samples are generated with disjoint random seeds.

E.4 Flow variants and backbone

We evaluate the path variants defined in Appendix D. I-CFM uses $\sigma_{\mathrm{icfm}} = 10^{-3}$. VP uses $\beta_{\min} = 0.1$, $\beta_{\max} = 20.0$, and endpoint truncation $t_{\max} = 1 - 10^{-5}$. All models were trained using

Table 5: Baseline sweep grids.

Method	Sweep grid
Plain CFG (Zheng et al., 2023)	$\omega \in \{1.0, 1.25, 1.5, 2.0, 3.0\}$
CFG-Zero* (Fan et al., 2025)	$\omega \in \{1.0, 1.25, 1.5, 2.0, 3.0\}$, $K_{\text{zero}} \in \{1, 2\}$
CFG-MP (Cai et al., 2026)	$\omega \in \{1.0, 1.25, 1.5, 2.0, 3.0\}$, $K_{\text{proj}} \in \{1, 2\}$ (GM); $K_{\text{proj}} = 3$ (MNIST)
Rectified-CFG++ (Saini et al., 2025)	$\lambda_{\max} \in \{0.5, 1.0, 2.0, 3.0\}$, $\sigma \in \{0.0, 0.05\}$

implementations adapted from the torchcfm (Tong et al., 2024) library; see Appendix G. We make only the minimal modification needed to pass the class label as an additional network input. All reported methods use frozen trained velocity fields. The backbone is a conditional MLP in $d = 2$, with hidden width 64, 3 layers, time embedding dimension 2, and class conditioning using one hot encoding (Potdar et al., 2017) of the input label. Conditional and unconditional predictions share one network trained with null-label dropout probability $p_{\text{uncond}} = 0.2$. Training minimizes the flow-matching regression loss (Zheng et al., 2023; Lipman et al., 2023) $\|v_\theta(t, x, y) - u_t(x \mid y)\|_2^2$. We use Adam optimizer (Kingma & Ba, 2015) with learning rate 10^{-3}, batch size 256, gradient clipping at norm 1.0, no EMA. All models are trained for $20{,}000$ iterations.

The backbone for the MNIST is a conditional convolutional UNet on grayscale MNIST (LeCun et al., 1998) images $x \in \mathbb{R}^{1\times 28\times 28}$ with base channel width 64, two down/up stages plus a middle block, sinusoidal time embedding of dimension 128 processed by a small time MLP, and class conditioning via a learned embedding of dimension 32 concatenated to the time features for FiLM-style modulation in residual blocks (Perez et al., 2018). We use two networks: a class-conditional UNet as above, and a second UNet with identical width and stage layout but no class input, trained separately to match the same flow target without conditioning. Training minimizes the flow-matching regression loss with the ℓ_2 norm taken over the pixel/channel dimensions of the velocity. All models are trained for $100{,}000$ iterations.

E.5 Baseline sweeps

All baseline comparisons use the same reference grid, matched latents, and inference seeds as the proposed schedule. For each method, the baseline values reported in the sweep tables correspond to the best configurations for that method. The best configuration is selected by the lowest average rank over the evaluated metrics; thus, only the best configurations for each method are included in these tables, with respect to the corresponding Cartesian product in Table 5.

For the proposed method, every pair of fitting seed and test-function seed gives one fitted per-class schedule. The Gaussian-mixture tables use the schedule fitted with the first fitting seed. For MNIST (Table 2) we apply to our method the same selection rule used for every baseline, reporting the fitting seed with the lowest mean FID over the three evaluation seeds; all three fitting seeds are listed in Table 16, and each of them is below the best baseline. Schedule-variance figures report variability over all fitting seeds.

E.6 Test functions

The default weak-form objective uses polynomial test functions up to degree two,

$$\Psi_{\text{poly},2} = \left\{x \mapsto a^\top x\right\} \cup \left\{x \mapsto \frac{1}{2} x^\top A x\right\}, \tag{83}$$

so that the test family in Eq (48) is formed by sampling $a \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d\times d}$ from Gaussian ensembles L times. To align with the theoretical requirement for compact support, each function is multiplied by a smooth C^∞ bump $\eta_R(x)$ that vanishes outside a ball of radius $R + 1$. In practice, we set $R = 10^4$ so that $\eta_R \equiv 1$ for all samples; thus, the truncation is a mathematical formality that does not affect numerical results. The same test-function family and seed convention are used for all compared methods that require a fitted schedule. Figure 4 isolates the effect of varying L, while the ablation in Appendix F (Table 7) decomposes $\Psi_{\text{poly},2}$.

The family is interpretable and its blind spots are explicit: linear tests detect mean-velocity mismatch, and quadratic tests additionally target the symmetric first spatial moment governing second-

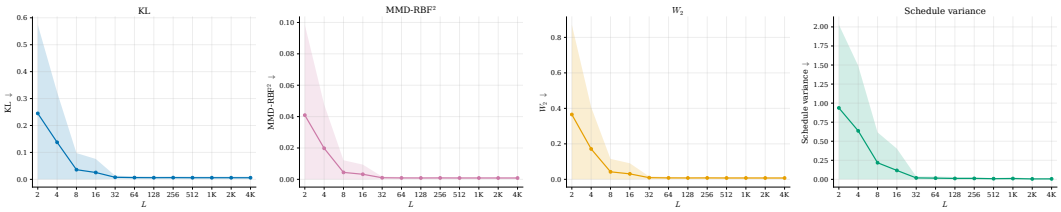


Figure 4: Sensitivity analysis of L, the number of test functions used to estimate the selector. This graph aggregates over all flow variants, classes, fitting and inference seeds. Shaded regions show one standard deviation over fitting seeds. Downstream metrics and schedule variance stabilize at moderate L.

moment evolution. In a Gaussian detectability check, injected mean and symmetric-linear mismatches are detected 2-4 orders of magnitude above Monte Carlo error, whereas an orthogonal third-Hermite mismatch $(x_1^3 - 3x_1)e_1$ is nearly invisible. Test functions should therefore be chosen for the data modality: low-degree polynomials are appropriate where low-order moments are informative, and richer families are needed for higher-order or structured discrepancies.

Sensitivity to the random probe draw is low. Holding the fitting latents, rollout noise, and inference latents bit-identical, independent draws at $L = 2^{12}$ give mean schedule standard deviations 0.00033 (linear), 0.00207 (mixed), and 0.00660 (quadratic-only), so the default mixed family is highly reproducible.

Table 6 reallocates the same budget $L = 2^{12}$ across linear, quadratic, and cubic probes on MNIST, paired against the default $(2048, 2048, 0)$ mixed family. All runs use identical checkpoints, fitting particles, endpoint-sampling streams, solver grids, and evaluation latents; only the probe allocation differs. Adding cubic probes yields at most modest gains while keeping the fitted schedules close to the default, whereas an all-cubic family fails outright, supporting the mixed default.

Table 6: Equal-budget probe allocations on MNIST. Paired ΔFID is variant minus default; mean schedule difference is the mean absolute scale difference over the 50 solver intervals.

$(L_{\text{lin}}, L_{\text{quad}}, L_{\text{cubic}})$	Flow	Paired ΔFID	95% CI	Mean schedule diff.
$(2048, 1024, 1024)$	RF	-0.0011	$[-0.0047, 0.0038]$	0.0033 ± 0.0038
$(2048, 1024, 1024)$	OT	-0.0220	$[-0.0321, -0.0133]$	0.0045 ± 0.0057
$(1024, 2048, 1024)$	RF	-0.2499	$[-0.3860, -0.0626]$	0.0103 ± 0.0142
$(1024, 2048, 1024)$	OT	-0.5663	$[-1.1632, 0.1931]$	0.0123 ± 0.0156
$(1024, 1024, 2048)$	RF	-0.2544	$[-0.3980, -0.0702]$	0.0100 ± 0.0131
$(1024, 1024, 2048)$	OT	-0.5924	$[-1.1851, 0.1669]$	0.0144 ± 0.0176
$(0, 0, 4096)$	RF	$+64.8805$	$[63.6820, 66.1250]$	1.7277 ± 0.2654
$(0, 0, 4096)$	OT	$+33.5259$	$[31.8922, 35.6230]$	1.8590 ± 0.7304

E.7 Numerical stabilization

The interval-wise selector solves a one-dimensional least-squares problem with empirical coefficients $\widetilde{A}_i, \widetilde{B}_i$. The solution quotient (53) can become unstable when the empirical CFG direction is nearly zero. In that case, changing ω_i has little effect on the local objective, so the minimizer is poorly identified and small Monte Carlo errors in $\widetilde{A}_i$ may produce large jumps in the selected scale.

To stabilize only this numerical division, we define

$$\widetilde{\beta}_i := \|\widetilde{B}_i\|_2^2, \qquad \widetilde{\beta}_i^\star := \max_{0 \le j < i} \widetilde{\beta}_j, \tag{84}$$

and replace the denominator by the running-max floor so that the implemented update is

$$\widehat{\omega}_i = \frac{\left\langle \widetilde{A}_i, \widetilde{B}_i \right\rangle}{\max\left\{ \widetilde{\beta}_i,\, \eta\, \widetilde{\beta}_i^\star \right\}} \tag{85}$$

In all experiments using the beta floor, we set $\eta = 10^{-2}$. The floor is relative rather than absolute, so it adapts to the scale of the fitted objective coefficients. When inactive, the original closed-form

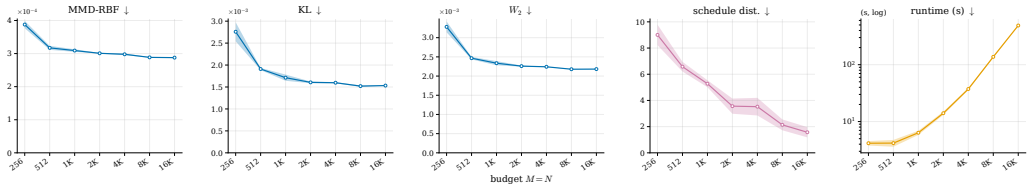


Figure 5: Monte Carlo scaling on the diagonal $M = N$. Shaded regions show one standard deviation over fitting seeds. Downstream metrics and schedule distance stabilize at moderate budgets.

selector is recovered exactly, when active, it only shrinks the raw quotient before projection. The denominator-stabilization ablation in Appendix F (Table 7) isolates the effect of varying η.

Over the 19,800 selector updates in the reported runs, the floor is active in 24.10% of updates and the projection returns the lower bound in 53.34%. Floor activation is 0% on MNIST, 6.6-9.3% on VP, and $\approx 35\%$ on the remaining Gaussian-mixture flows. Because $\omega_{\min} = 1$ is the conditional-only field, an interval projected to the bound applies *no* CFG extrapolation: on those intervals the selector determines that the conditional field alone best matches the target transport. We adopt $\omega_{\min} = 1$ as an empirical stabilization constraint; Appendix F.3 measures the effect of relaxing it.

F Ablation Study

We evaluate the core components of our selector using a one-factor-at-a-time protocol across all flow variants. The default configuration uses $T = 100$ intervals, $M = 2^{14}$ endpoint samples, $N = 2^{14}$ particles, and $L_{\text{test}} = 2^{12}$ mixed linear-quadratic test functions. Table 7 summarizes the aggregate results over all flow variants. We ablate four primary axes to justify our design choices:

- **Schedule Reusability:** We compare *online* selection against *offline* reuse on fresh particles. This validates our method as an efficient "fit-once, use-many" procedure.
- **Endpoint Weighting:** We isolate the impact of *posterior weighting* against naive uniform averaging, demonstrating that accounting for the endpoint posterior structure significantly improves guidance fidelity.
- **Denominator Stabilization:** We test the impact of the stability floor η. Table 7 shows that while the floor has minimal effect in well-conditioned regimes, it is essential for preventing guidance spikes when the denominator vanishes.
- **Test-Function Family:** We evaluate linear, quadratic, and mixed families at a fixed computational budget. The mixed family provides the best balance, capturing both first- and second-moment mismatches.

Finally, we perform a sensitivity analysis on the number of test functions $L \in \{2^i\}_{i=1}^{12}$, visualized in Figure 4. We further analyze the sensitivity to the number of endpoint samples M and rollout particles N used by the schedule, with results shown in the heatmaps in Figure 6.

Monte Carlo scaling. Finally, we vary the estimator budget used by the practical selector. Figure 5 shows the diagonal sweep $M = N$, keeping the trained model and solver grid fixed. Downstream metrics and schedule distance stabilize at moderate budgets. The full (M, N) sweep in Figure 6 further shows that endpoint samples are especially important, as they affect both the posterior weights and the endpoint-conditioned velocity estimates.

F.1 Does the finite residual track path discrepancy?

In Table 8 we empirically test the link between the approximated residual and the real path discrepancy. We hold the interval t_i, the rollout history, and the particle state fixed, vary only ω over 26 candidates in $[1, 5]$, evaluate the held-out residual $J_{r,i}(\omega)$, propagate one matched Euler step, and measure the next-step discrepancy $D_{r,i+1}(\omega)$. We report $\rho_{r,i} = \text{Spearman}(J_{r,i}(\omega), D_{r,i+1}(\omega))$, averaged over $18 \times 20 = 360$ cells per flow, with 95% CIs from a run-clustered bootstrap.

Table 7: Aggregate ablation results over all flow variants. For each ablation variant and metric, we pool all raw measurements across the four flow backbones (RF, I-CFM, OT, and VP), fitting seeds, evaluation seeds, and classes. Each cell reports mean ± sample standard deviation. Lower is better.

Ablation	Variant	KL ↓	W_2 ↓	MMD ↓	Time↓
reuse	online	0.0059 ± 0.0041	0.0075 ± 0.0048	0.0009 ± 0.0005	13.58 ± 5.67
	offline, same T	0.0060 ± 0.0042	0.0076 ± 0.0049	0.0009 ± 0.0005	13.58 ± 5.66
weights	posterior	0.0059 ± 0.0041	0.0074 ± 0.0048	0.0008 ± 0.0005	13.51 ± 5.72
	uniform	0.0060 ± 0.0042	0.0075 ± 0.0049	0.0009 ± 0.0005	13.42 ± 5.61
stability	floor off	0.0065 ± 0.0048	0.0079 ± 0.0053	0.0009 ± 0.0006	13.52 ± 5.73
	floor on, $\eta = 10^{-2}$	0.0059 ± 0.0041	0.0075 ± 0.0048	0.0009 ± 0.0005	13.57 ± 5.60
	floor on, $\eta = 10^{-3}$	0.0061 ± 0.0044	0.0075 ± 0.0049	0.0009 ± 0.0005	13.53 ± 5.63
tests	linear	0.0064 ± 0.0041	0.0081 ± 0.0051	0.0009 ± 0.0005	13.25 ± 5.64
	quadratic	0.3476 ± 0.3205	0.5253 ± 0.4899	0.0591 ± 0.0554	13.26 ± 5.63
	linear + quadratic	0.0059 ± 0.0041	0.0074 ± 0.0048	0.0008 ± 0.0005	13.51 ± 5.72

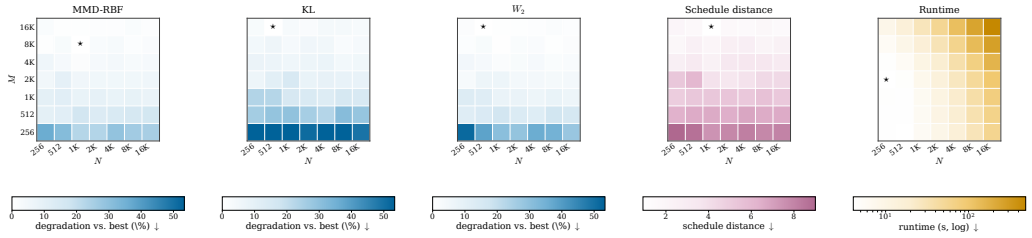


Figure 6: Full (M, N) sweep for downstream metrics. Each cell shows the percentage gap from the best observed budget for that metric; lighter is better and the star marks the best cell. The heatmaps show that increasing M gives the main early improvement, while very large N has smaller marginal effect once M is moderate.

Table 8: Correlation between the held-out finite residual and the resulting path discrepancy, at fixed time and fixed rollout state. Mean Spearman correlation [95% CI].

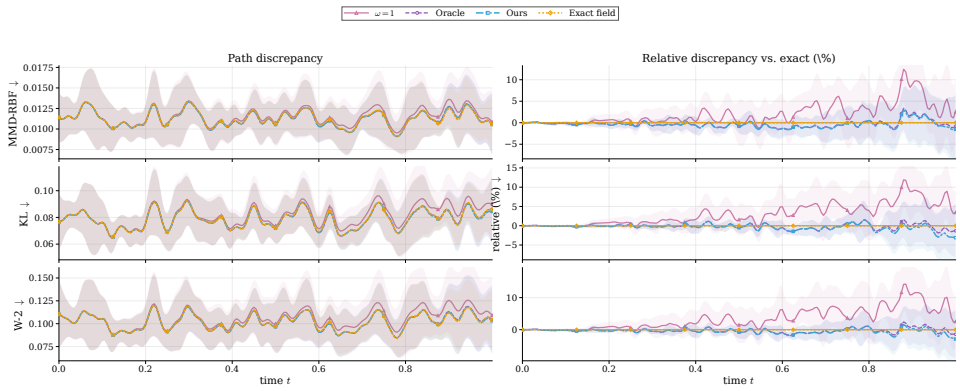
Flow	MMD	W_2	KL	Mean error	Cov. error
RF	0.57 [0.49, 0.65]	0.59 [0.52, 0.65]	0.55 [0.48, 0.62]	0.61 [0.47, 0.74]	0.47 [0.40, 0.54]
I-CFM	0.66 [0.56, 0.77]	0.65 [0.55, 0.74]	0.64 [0.53, 0.75]	0.62 [0.47, 0.76]	0.58 [0.48, 0.68]
OT	0.50 [0.38, 0.61]	0.55 [0.46, 0.65]	0.56 [0.45, 0.66]	0.59 [0.47, 0.71]	0.41 [0.30, 0.53]
VP	0.46 [0.23, 0.67]	0.50 [0.28, 0.70]	0.54 [0.33, 0.72]	0.16 [−0.13, 0.45]	0.44 [0.26, 0.60]

F.2 What the weak form contributes: pointwise projection control

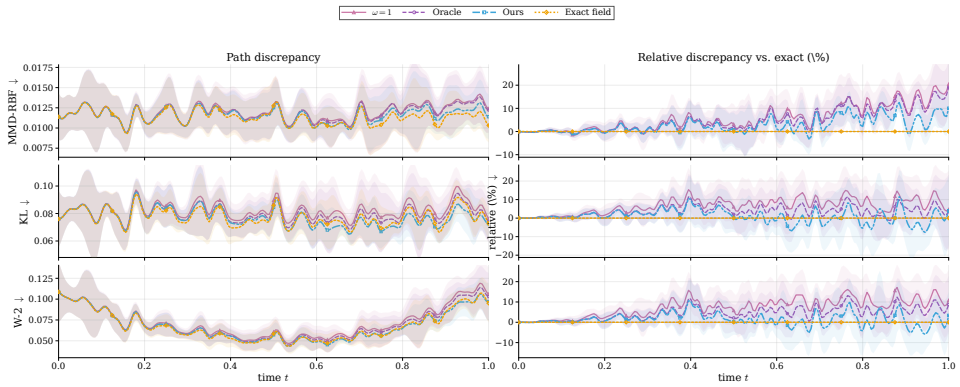
Given the same estimated endpoint-conditioned information, seeds, and stabilization, we compare PathGuide against selecting the scalar by a direct pointwise projection of the estimated conditional velocity onto the CFG direction, See Table 9. With identical endpoint information, the on-policy weak-form selector wins on RF, I-CFM, and OT across all three metrics and ties on VP. The gain is therefore attributable to the criterion, not to the endpoint-conditioned estimate itself.

Table 9: Pointwise projection versus the weak-form selector under identical endpoint information. Mean ± std over 18 paired settings; lower is better.

Flow	Pointwise projection: W_2^2 / KL / MMD^2	PathGuide: W_2^2 / KL / MMD^2
RF	.0592±.0034 / .0444±.0050 / .00532±.00071	.0135±.0056 / .0109±.0040 / .00180±.00065
I-CFM	.0580±.0034 / .0440±.0045 / .00539±.00069	.0130±.0055 / .0108±.0039 / .00178±.00065
OT	.0441±.0214 / .0329±.0178 / .00433±.00134	.0069±.0046 / .0053±.0028 / .00100±.00040
VP	.0056±.0031 / .0040±.0022 / .00066±.00019	.0058±.0031 / .0040±.0024 / .00067±.00020



(a) VP



(b) I-CFM

Figure 7: Online path discrepancy to the exact conditional marginal $p_{t_i}(\cdot \mid y)$, measured along $\hat{p}_{t_i}(\cdot \mid y; \vec{\omega}_i)$ by MMD-RBF^2, KL, and W_2. Lower is better. Panel (a) shows VP, and panel (b) shows I-CFM. The true-velocity rollout provides a discretization reference. In both cases, the practical/oracle selector stays closer to this reference than the learned conditional rollout $\omega \equiv 1$, indicating improved path alignment.

F.3 Relaxing the guidance lower bound

We rerun the complete fitting procedure with the admissible lower bound relaxed from $\omega_{\min} = 1$ to 0, using the same backbones, fitting and evaluation seeds, latents, solver grids, and estimator budgets. In Table 10 we report Δ = relaxed − default, so positive values indicate worse performance under relaxation. Results average all matched fitting/evaluation-seed pairs.

Table 10: Effect of relaxing the admissible lower bound from $\omega_{\min} = 1$ to 0. Positive value means $\omega_{\min} = 1$ is better.

Flow	ΔKL T=200/500	ΔW_2 T=200/500	ΔMMD^2 T=200/500
RF	$+.03264/+.03446$	$+.04097/+.04310$	$+.001005/+.001104$
I-CFM	$+.01161/+.01247$	$+.01303/+.01397$	$+.000593/+.000650$
OT	$+.00950/+.01033$	$+.01060/+.01148$	$+.000471/+.000528$
VP	$+.000001/+.000006$	$+.000139/+.000145$	$+.000004/+.000004$

MNIST flow	ΔFID [95% CI]
RF	$-0.940\,[-1.028, -0.853]$
OT	$+1.408\,[1.316, 1.501]$

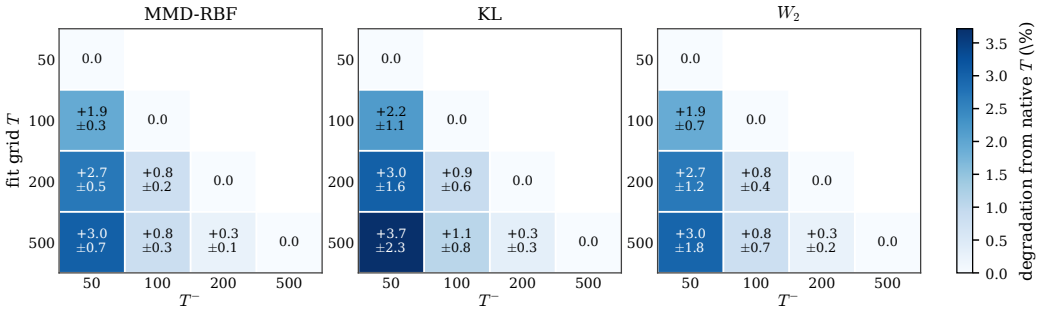


Figure 8: Relative degradation under coarser inference. Each entry reports the percentage increase in the metric when a schedule fitted at T is deployed at T^-, relative to evaluating the same schedule at its native grid T. Lower is better; 0% corresponds to native-grid evaluation.

F.4 Computational cost

The method has two modes. In online selection, objective coefficients are estimated during sampling. In offline schedule fitting, the schedule is fitted once for a fixed backbone, class, solver, and test-function family, then reused. The main experimental validation uses the offline mode unless explicitly stated.

Let C_{cond} and C_{uncond} denote the cost of one conditional and unconditional backbone evaluation, and let C_{path} denote the cost of evaluating endpoint-conditioned path densities and velocity labels for posterior weighting. For T solver intervals, N rollout particles, M endpoint samples, and L test functions, the dominant offline fitting cost scales as

$$\mathcal{O}(TN\left[C_{\mathrm{cond}} + C_{\mathrm{uncond}} + MC_{\mathrm{path}} + Ld\right]). \tag{86}$$

After a schedule is fitted, deployment has the same backbone-evaluation cost as standard CFG on the same solver grid:

$$\mathcal{O}(TN_{\mathrm{gen}}\left[C_{\mathrm{cond}} + C_{\mathrm{uncond}}\right]). \tag{87}$$

Table 11: Cost accounting for fitted schedules. Offline fitting is amortized across future deployments; deployment uses the stored schedule and therefore has the same NFE-equivalent as standard CFG.

Method	Offline fit cost	Sampling cost	Memory	NFE-equivalent
Conditional-only	none	TC_{cond}	$\mathcal{O}(N_{\mathrm{gen}}d)$	T
Constant CFG	sweep over Ω_{CFG}	$T(C_{\mathrm{cond}} + C_{\mathrm{uncond}})$	$\mathcal{O}(N_{\mathrm{gen}}d)$	$2T$
Proposed, offline	Eq. (86)	$T(C_{\mathrm{cond}} + C_{\mathrm{uncond}})$	$\mathcal{O}((N+M)d)$	$2T$
Proposed, online	none	Eq. (86) during sampling	$\mathcal{O}((N+M)d)$	$2T$ plus estimator work

Table 12: Measured cost per workload. Each cell reports wall-clock time / peak memory / backbone NFE. Deployment with a stored schedule matches plain CFG on all three; only fitting is additional, and it is amortized across later deployments.

Workload	$\omega \equiv 1$	CFG / stored PathGuide	PathGuide fitting
GMM, $N = M = 2^{14}$, $L = 2^{12}$, $T = 500$	10.0 s / 34 MB / T	19.7 s / 34 MB / $2T$	137 s / 333 MB / $2T$
MNIST, $N = 2^{12}$, $M = 2^{11}$, $L = 2^{12}$, $T = 50$	29.9 s / 315 MB / T	59.6 s / 327 MB / $2T$	230 s / 578 MB / $2T$

The estimator reuses the conditional and unconditional backbone evaluations already required by the ODE step, so it adds no NFE; its overhead is posterior and test-function estimation. One fit costs ≈ 7 GMM or ≈ 4 MNIST CFG generations, and amortizes against repeated online selection after two stored-schedule deployments.

Experiments compute resources. All experiments were run on a single workstation equipped with six NVIDIA RTX 2080 Ti GPUs, using CUDA 12.4, PyTorch 2.6.0+cu124, and Python 3.9.20.

Table 13 summarizes the aggregate compute budget across all completed experiment runs used in the paper.

Table 13: Compute resources for the experiment families reported in the paper. All experiments were run on the same workstation. Runtime is reported as accumulated single-GPU time.

Experiment family	Runs	Total time	Mean time
Baseline sweeps	163	226.9h	1.4h
Correction-field diagnostics	173	27.8h	9.6min
Online path alignment	11	22.8h	2.1h
Fit-at-T, infer-at-coarser-T	136	145.3h	1.1h
Monte Carlo scaling	87	236.5h	2.7h
Schedule heatmap overlays	86	60.3h	42.1min
Fitting-resolution sweep	81	70.9h	52.5min
MNIST velocity training	2	8.8h	4.4h
Gaussian Mixture velocity training	4	8.0min	2.0min
Total reported	**743**	**799.4h**	**1.1h**

G LICENSES

This project utilizes the following third-party libraries and datasets:

- **torchcfm** (v1.0.7): Tong et al. (2023; 2024), MIT License. Used for flow matching and OT-CFM training. `https://github.com`
- **MNIST Dataset**: LeCun et al. (1998), CC BY-SA 3.0 License. Used for image experiments. `http://lecun.com`
- **PyTorch**: Paszke et al. (2019), BSD-3-Clause License. Base framework. `https://pytorch.org`
- **POT (Python Optimal Transport)**: Flamary et al. (2021), MIT License. Used for W_2 computation. `https://github.io`
- **torch-fidelity**: Obukhov et al. (2020), Apache-2.0 License. Used for FID computation. `https://github.com`

H FULL TABLES

Table 14: Appendix sweep results for baseline configurations at $T = 200$. For each flow variant and method, we report up to three configurations from the hyperparameter sweep. Lower is better for all metrics. All entries are evaluated over 3 inference seeds using 2^{14} generated samples per seed, and report mean $\pm$ standard deviation. The mean is averaged over classes per evaluation seed; the standard deviation is the sample std over the 3 per-seed averages.

Variant	Method	Configuration	KL↓	W_2 ↓	MMD↓
RF	Plain CFG	$\omega = 1$	$0.0044_{0.0013}$	$0.0060_{0.0016}$	$0.0006_{0.0001}$
		$\omega = 1.25$	$0.0441_{0.0005}$	$0.0657_{0.0025}$	$0.0054_{0.0003}$
		$\omega = 1.5$	$0.1298_{0.0015}$	$0.1961_{0.0052}$	$0.0175_{0.0005}$
	CFG-Zero*	$\omega = 1, K_{\text{zero}} = 1$	$0.0046_{0.0013}$	$0.0064_{0.0018}$	$0.0006_{0.0001}$
		$\omega = 1, K_{\text{zero}} = 2$	$0.0050_{0.0014}$	$0.0072_{0.0019}$	$0.0006_{0.0001}$
		$\omega = 1.25, K_{\text{zero}} = 2$	$0.0080_{0.0005}$	$0.0127_{0.0013}$	$0.0013_{0.0002}$
	CFG-MP	$\omega = 3, K_{\text{proj}} = 2$	$6.2232_{0.0225}$	$9.5319_{0.0356}$	$0.6234_{0.0018}$
		$\omega = 2, K_{\text{proj}} = 2$	$6.1985_{0.0204}$	$9.5446_{0.0354}$	$0.6355_{0.0018}$
		$\omega = 1.25, K_{\text{proj}} = 2$	$6.0096_{0.2804}$	$9.5237_{0.1140}$	$0.6443_{0.0016}$
	R-CFG++	$\lambda = 0.5, \sigma = 0$	$0.0191_{0.0006}$	$0.0248_{0.0006}$	$0.0020_{0.0001}$
		$\lambda = 0.5, \sigma = 0.05$	$0.0192_{0.0006}$	$0.0248_{0.0006}$	$0.0020_{0.0001}$
		$\lambda = 1, \sigma = 0$	$0.0534_{0.0003}$	$0.0688_{0.0017}$	$0.0057_{0.0002}$
	PathGuide (Ours)	seed_1	$\mathbf{0.0039_{0.0012}}$	$\mathbf{0.0056_{0.0013}}$	$\mathbf{0.0005_{0.0001}}$
		seed_2	$0.0039_{0.0011}$	$0.0056_{0.0012}$	$0.0005_{0.0001}$
		seed_3	$0.0040_{0.0011}$	$0.0058_{0.0011}$	$0.0005_{0.0001}$
I-CFM	Plain CFG	$\omega = 1$	$0.0084_{0.0021}$	$0.0112_{0.0029}$	$0.0012_{0.0003}$
		$\omega = 1.25$	$0.0423_{0.0004}$	$0.0645_{0.0018}$	$0.0056_{0.0002}$
		$\omega = 1.5$	$0.1220_{0.0008}$	$0.1898_{0.0043}$	$0.0172_{0.0004}$
	CFG-Zero*	$\omega = 1, K_{\text{zero}} = 1$	$0.0084_{0.0022}$	$0.0114_{0.0030}$	$0.0012_{0.0003}$
		$\omega = 1, K_{\text{zero}} = 2$	$0.0086_{0.0022}$	$0.0119_{0.0031}$	$0.0012_{0.0003}$
		$\omega = 1.25, K_{\text{zero}} = 2$	$0.0101_{0.0010}$	$0.0157_{0.0016}$	$0.0018_{0.0003}$
	CFG-MP	$\omega = 1, K_{\text{proj}} = 1$	$0.1197_{0.0007}$	$0.1853_{0.0042}$	$0.0168_{0.0004}$
		$\omega = 1.25, K_{\text{proj}} = 1$	$0.2140_{0.0016}$	$0.3348_{0.0062}$	$0.0322_{0.0006}$
		$\omega = 1.5, K_{\text{proj}} = 1$	$0.3133_{0.0024}$	$0.4939_{0.0080}$	$0.0497_{0.0007}$
	R-CFG++	$\lambda = 0.5, \sigma = 0$	$0.0192_{0.0013}$	$0.0252_{0.0011}$	$0.0024_{0.0002}$
		$\lambda = 0.5, \sigma = 0.05$	$0.0193_{0.0013}$	$0.0252_{0.0011}$	$0.0024_{0.0002}$
		$\lambda = 1, \sigma = 0$	$0.0500_{0.0008}$	$0.0652_{0.0011}$	$0.0057_{0.0002}$
	PathGuide (Ours)	seed_1	$\mathbf{0.0076_{0.0019}}$	$\mathbf{0.0099_{0.0022}}$	$\mathbf{0.0012_{0.0003}}$
		seed_2	$0.0077_{0.0018}$	$0.0100_{0.0021}$	$0.0012_{0.0003}$
		seed_3	$0.0078_{0.0018}$	$0.0102_{0.0020}$	$0.0012_{0.0003}$
OT	Plain CFG	$\omega = 1$	$0.0082_{0.0020}$	$0.0111_{0.0027}$	$0.0012_{0.0003}$
		$\omega = 1.25$	$0.0434_{0.0003}$	$0.0669_{0.0020}$	$0.0058_{0.0002}$
		$\omega = 1.5$	$0.1238_{0.0010}$	$0.1935_{0.0045}$	$0.0175_{0.0004}$
	CFG-Zero*	$\omega = 1, K_{\text{zero}} = 1$	$0.0081_{0.0020}$	$0.0112_{0.0029}$	$\mathbf{0.0012_{0.0003}}$
		$\omega = 1, K_{\text{zero}} = 2$	$0.0082_{0.0021}$	$0.0115_{0.0030}$	$0.0012_{0.0003}$
		$\omega = 1.25, K_{\text{zero}} = 2$	$0.0105_{0.0009}$	$0.0169_{0.0016}$	$0.0018_{0.0003}$
	CFG-MP	$\omega = 1, K_{\text{proj}} = 1$	$0.1215_{0.0009}$	$0.1889_{0.0043}$	$0.0171_{0.0004}$
		$\omega = 1.25, K_{\text{proj}} = 1$	$0.2155_{0.0018}$	$0.3387_{0.0064}$	$0.0325_{0.0006}$
		$\omega = 1.5, K_{\text{proj}} = 1$	$0.3137_{0.0026}$	$0.4972_{0.0082}$	$0.0498_{0.0007}$
	R-CFG++	$\lambda = 0.5, \sigma = 0$	$0.0199_{0.0012}$	$0.0266_{0.0011}$	$0.0024_{0.0002}$
		$\lambda = 0.5, \sigma = 0.05$	$0.0200_{0.0012}$	$0.0266_{0.0011}$	$0.0025_{0.0002}$
		$\lambda = 1, \sigma = 0$	$0.0515_{0.0007}$	$0.0676_{0.0012}$	$0.0060_{0.0002}$
	PathGuide (Ours)	seed_1	$\mathbf{0.0076_{0.0017}}$	$\mathbf{0.0102_{0.0021}}$	$0.0012_{0.0003}$
		seed_2	$0.0077_{0.0017}$	$0.0103_{0.0020}$	$0.0012_{0.0003}$
		seed_3	$0.0078_{0.0017}$	$0.0105_{0.0020}$	$0.0012_{0.0003}$
VP	Plain CFG	$\omega = 1$	$0.0030_{0.0007}$	$0.0052_{0.0013}$	$0.0005_{0.0001}$
		$\omega = 1.25$	$0.0254_{0.0009}$	$0.0372_{0.0027}$	$0.0027_{0.0002}$
		$\omega = 1.5$	$0.0912_{0.0018}$	$0.1403_{0.0052}$	$0.0116_{0.0004}$
	CFG-Zero*	$\omega = 1.5, K_{\text{zero}} = 1$	$0.0013_{0.0003}$	$0.0019_{0.0005}$	$0.0003_{0.0000}$
		$\omega = 1.5, K_{\text{zero}} = 2$	$0.0014_{0.0003}$	$0.0020_{0.0005}$	$0.0003_{0.0000}$
		$\omega = 1.25, K_{\text{zero}} = 1$	$0.0019_{0.0004}$	$0.0031_{0.0008}$	$0.0003_{0.0000}$
	CFG-MP	$\omega = 1, K_{\text{proj}} = 2$	$2.9991_{0.0226}$	$4.7929_{0.0130}$	$0.4827_{0.0008}$
		$\omega = 1.25, K_{\text{proj}} = 2$	$3.1150_{0.0695}$	$4.9108_{0.0261}$	$0.4866_{0.0008}$
		$\omega = 1.5, K_{\text{proj}} = 2$	$3.1485_{0.0620}$	$4.9986_{0.0235}$	$0.4905_{0.0008}$
	R-CFG++	$\lambda = 0.5, \sigma = 0$	$0.0290_{0.0007}$	$0.0388_{0.0023}$	$0.0029_{0.0002}$
		$\lambda = 0.5, \sigma = 0.05$	$0.0291_{0.0007}$	$0.0389_{0.0023}$	$0.0029_{0.0002}$
		$\lambda = 1, \sigma = 0$	$0.0931_{0.0013}$	$0.1318_{0.0045}$	$0.0110_{0.0004}$
	PathGuide (Ours)	seed_1	$\mathbf{0.0011_{0.0004}}$	$0.0015_{0.0005}$	$\mathbf{0.0003_{0.0000}}$
		seed_2	$0.0011_{0.0004}$	$\mathbf{0.0015_{0.0005}}$	$0.0003_{0.0000}$
		seed_3	$0.0011_{0.0004}$	$0.0015_{0.0005}$	$0.0003_{0.0000}$

Table 15: Appendix sweep results for baseline configurations at $T = 500$. For each flow variant and method, we report up to three configurations from the hyperparameter sweep. Lower is better for all metrics. All entries are evaluated over 3 inference seeds using 2^{14} generated samples per seed, and report mean $\pm$ standard deviation. The mean is averaged over classes per evaluation seed; the standard deviation is the sample std over the 3 per-seed averages.

Variant	Method	Configuration	KL↓	W_2 ↓	MMD↓
RF	Plain CFG	$\omega = 1$	$0.0044_{0.0012}$	$0.0060_{0.0016}$	$0.0006_{0.0001}$
		$\omega = 1.25$	$0.0429_{0.0005}$	$0.0645_{0.0026}$	$0.0053_{0.0003}$
		$\omega = 1.5$	$0.1277_{0.0016}$	$0.1939_{0.0052}$	$0.0172_{0.0005}$
	CFG-Zero*	$\omega = 1, K_{\text{zero}} = 1$	$0.0045_{0.0013}$	$0.0062_{0.0017}$	$0.0006_{0.0001}$
		$\omega = 1, K_{\text{zero}} = 2$	$0.0046_{0.0013}$	$0.0064_{0.0017}$	$0.0006_{0.0001}$
		$\omega = 1.25, K_{\text{zero}} = 2$	$0.0093_{0.0005}$	$0.0150_{0.0014}$	$0.0015_{0.0002}$
	CFG-MP	$\omega = 3, K_{\text{proj}} = 2$	$1.8740_{0.0116}$	$2.9592_{0.0236}$	$0.2987_{0.0005}$
		$\omega = 2, K_{\text{proj}} = 2$	$1.5907_{0.0101}$	$2.5036_{0.0195}$	$0.2658_{0.0005}$
		$\omega = 1.25, K_{\text{proj}} = 2$	$1.3863_{0.0253}$	$2.1654_{0.0297}$	$0.2382_{0.0006}$
	R-CFG++	$\lambda = 0.5, \sigma = 0, \gamma = 1.0$	$0.0185_{0.0005}$	$0.0243_{0.0007}$	$0.0019_{0.0001}$
		$\lambda = 0.5, \sigma = 0.05, \gamma = 1.0$	$0.0186_{0.0005}$	$0.0243_{0.0007}$	$0.0019_{0.0001}$
		$\lambda = 1, \sigma = 0, \gamma = 1.0$	$0.0525_{0.0002}$	$0.0683_{0.0018}$	$0.0055_{0.0002}$
	PathGuide (Ours)	seed_1	$\mathbf{0.0038_{0.0011}}$	$\mathbf{0.0055_{0.0013}}$	$\mathbf{0.0005_{0.0001}}$
		seed_2	$0.0038_{0.0011}$	$0.0055_{0.0012}$	$0.0005_{0.0001}$
		seed_3	$0.0039_{0.0011}$	$0.0056_{0.0011}$	$0.0005_{0.0001}$
I-CFM	Plain CFG	$\omega = 1$	$0.0083_{0.0021}$	$0.0111_{0.0028}$	$0.0012_{0.0003}$
		$\omega = 1.25$	$0.0411_{0.0003}$	$0.0634_{0.0019}$	$0.0054_{0.0003}$
		$\omega = 1.5$	$0.1200_{0.0009}$	$0.1877_{0.0043}$	$0.0169_{0.0004}$
	CFG-Zero*	$\omega = 1, K_{\text{zero}} = 1$	$0.0083_{0.0021}$	$0.0111_{0.0029}$	$0.0012_{0.0003}$
		$\omega = 1, K_{\text{zero}} = 2$	$0.0083_{0.0021}$	$0.0112_{0.0029}$	$0.0012_{0.0003}$
		$\omega = 1.25, K_{\text{zero}} = 2$	$0.0114_{0.0009}$	$0.0182_{0.0015}$	$0.0020_{0.0003}$
	CFG-MP	$\omega = 1, K_{\text{proj}} = 1$	$0.1191_{0.0008}$	$0.1859_{0.0043}$	$0.0167_{0.0004}$
		$\omega = 1.25, K_{\text{proj}} = 1$	$0.2131_{0.0017}$	$0.3355_{0.0064}$	$0.0321_{0.0006}$
		$\omega = 1.5, K_{\text{proj}} = 1$	$0.3121_{0.0025}$	$0.4944_{0.0081}$	$0.0494_{0.0007}$
	R-CFG++	$\lambda = 0.5, \sigma = 0, \gamma = 1.0$	$0.0185_{0.0012}$	$0.0246_{0.0011}$	$0.0023_{0.0002}$
		$\lambda = 0.5, \sigma = 0.05, \gamma = 1.0$	$0.0186_{0.0012}$	$0.0247_{0.0011}$	$0.0023_{0.0002}$
		$\lambda = 1, \sigma = 0, \gamma = 1.0$	$0.0491_{0.0007}$	$0.0647_{0.0012}$	$0.0056_{0.0002}$
	PathGuide (Ours)	seed_1	$\mathbf{0.0073_{0.0018}}$	$\mathbf{0.0096_{0.0022}}$	$\mathbf{0.0012_{0.0003}}$
		seed_2	$0.0073_{0.0018}$	$0.0097_{0.0021}$	$0.0012_{0.0003}$
		seed_3	$0.0075_{0.0018}$	$0.0098_{0.0020}$	$0.0012_{0.0003}$
OT	Plain CFG	$\omega = 1$	$0.0080_{0.0020}$	$0.0110_{0.0027}$	$0.0012_{0.0003}$
		$\omega = 1.25$	$0.0422_{0.0003}$	$0.0657_{0.0021}$	$0.0056_{0.0003}$
		$\omega = 1.5$	$0.1217_{0.0010}$	$0.1914_{0.0045}$	$0.0172_{0.0004}$
	CFG-Zero*	$\omega = 1, K_{\text{zero}} = 1$	$0.0080_{0.0020}$	$0.0110_{0.0028}$	$0.0011_{0.0003}$
		$\omega = 1, K_{\text{zero}} = 2$	$0.0080_{0.0020}$	$0.0111_{0.0028}$	$0.0012_{0.0003}$
		$\omega = 1.25, K_{\text{zero}} = 2$	$0.0120_{0.0009}$	$0.0195_{0.0017}$	$0.0021_{0.0003}$
	CFG-MP	$\omega = 1, K_{\text{proj}} = 1$	$0.1208_{0.0010}$	$0.1895_{0.0045}$	$0.0170_{0.0004}$
		$\omega = 1.25, K_{\text{proj}} = 1$	$0.2146_{0.0019}$	$0.3394_{0.0065}$	$0.0324_{0.0006}$
		$\omega = 1.5, K_{\text{proj}} = 1$	$0.3125_{0.0027}$	$0.4977_{0.0083}$	$0.0496_{0.0007}$
	R-CFG++	$\lambda = 0.5, \sigma = 0, \gamma = 1.0$	$0.0192_{0.0011}$	$0.0260_{0.0011}$	$0.0024_{0.0002}$
		$\lambda = 0.5, \sigma = 0.05, \gamma = 1.0$	$0.0192_{0.0011}$	$0.0260_{0.0011}$	$0.0024_{0.0002}$
		$\lambda = 1, \sigma = 0, \gamma = 1.0$	$0.0505_{0.0006}$	$0.0670_{0.0013}$	$0.0059_{0.0002}$
	PathGuide (Ours)	seed_1	$\mathbf{0.0073_{0.0017}}$	$\mathbf{0.0099_{0.0021}}$	$\mathbf{0.0011_{0.0003}}$
		seed_2	$0.0073_{0.0017}$	$0.0099_{0.0020}$	$0.0011_{0.0003}$
		seed_3	$0.0075_{0.0016}$	$0.0102_{0.0019}$	$0.0012_{0.0003}$
VP	Plain CFG	$\omega = 1$	$0.0030_{0.0007}$	$0.0052_{0.0013}$	$0.0005_{0.0001}$
		$\omega = 1.25$	$0.0252_{0.0009}$	$0.0369_{0.0027}$	$0.0027_{0.0002}$
		$\omega = 1.5$	$0.0905_{0.0018}$	$0.1393_{0.0052}$	$0.0115_{0.0004}$
	CFG-Zero*	$\omega = 1.5, K_{\text{zero}} = 1$	$0.0013_{0.0003}$	$0.0019_{0.0005}$	$0.0003_{0.0000}$
		$\omega = 1.5, K_{\text{zero}} = 2$	$0.0013_{0.0003}$	$0.0019_{0.0005}$	$0.0003_{0.0000}$
		$\omega = 1.25, K_{\text{zero}} = 1$	$0.0019_{0.0004}$	$0.0031_{0.0008}$	$0.0003_{0.0000}$
	CFG-MP	$\omega = 1, K_{\text{proj}} = 2$	$1.1612_{0.0127}$	$1.6279_{0.0133}$	$0.2012_{0.0005}$
		$\omega = 1.25, K_{\text{proj}} = 2$	$1.2188_{0.0021}$	$1.7345_{0.0131}$	$0.2098_{0.0005}$
		$\omega = 1.5, K_{\text{proj}} = 2$	$1.2585_{0.0019}$	$1.8333_{0.0139}$	$0.2186_{0.0005}$
	R-CFG++	$\lambda = 0.5, \sigma = 0, \gamma = 1.0$	$0.0289_{0.0007}$	$0.0387_{0.0023}$	$0.0028_{0.0002}$
		$\lambda = 0.5, \sigma = 0.05, \gamma = 1.0$	$0.0290_{0.0007}$	$0.0387_{0.0023}$	$0.0028_{0.0002}$
		$\lambda = 1, \sigma = 0, \gamma = 1.0$	$0.0927_{0.0013}$	$0.1313_{0.0045}$	$0.0110_{0.0004}$
	PathGuide (Ours)	seed_1	$0.0011_{0.0004}$	$0.0015_{0.0005}$	$0.0003_{0.0000}$
		seed_2	$\mathbf{0.0011_{0.0004}}$	$\mathbf{0.0015_{0.0005}}$	$\mathbf{0.0003_{0.0000}}$
		seed_3	$0.0011_{0.0004}$	$0.0015_{0.0005}$	$0.0003_{0.0000}$

Table 16: Full MNIST image-generation sweep results for RF and OT flow variants at $T = 50$. FID is computed with `torch-fidelity==0.4.0`; lower is better. All entries are evaluated over three evaluation seeds using 2^{13} generated samples per seed, with $M = 2^{11}$ and $N = 2^{12}$. Entries report mean $\pm$ sample standard deviation over three evaluation seeds.

Variant	Method	Configuration	FID↓
RF	Plain CFG	$\omega = 1.0$	15.802 ± 0.464
		$\omega = 1.25$	16.342 ± 0.506
		$\omega = 1.5$	18.567 ± 0.640
		$\omega = 2.0$	26.028 ± 0.718
	CFG-Zero*	$\omega = 1.0, K_{\text{zero}} = 1$	29.627 ± 0.494
		$\omega = 1.25, K_{\text{zero}} = 1$	26.805 ± 0.610
		$\omega = 1.5, K_{\text{zero}} = 1$	26.534 ± 0.747
		$\omega = 2.0, K_{\text{zero}} = 1$	30.908 ± 0.844
	CFG-MP	$\omega = 1.0, K_{\text{proj}} = 3$	30.951 ± 0.999
		$\omega = 1.25, K_{\text{proj}} = 3$	37.156 ± 0.869
		$\omega = 1.5, K_{\text{proj}} = 3$	43.712 ± 0.621
		$\omega = 2.0, K_{\text{proj}} = 3$	57.352 ± 0.725
	R-CFG++	$\lambda_{\max} = 1.0, \gamma = 1.0, \sigma = 0.0$	16.979 ± 0.473
	PathGuide (Ours)	seed_1	$\mathbf{15.699 \pm 0.437}$
		seed_2	15.737 ± 0.446
		seed_3	15.720 ± 0.430
OT	Plain CFG	$\omega = 1.0$	7.578 ± 0.263
		$\omega = 1.25$	8.262 ± 0.254
		$\omega = 1.5$	10.596 ± 0.215
		$\omega = 2.0$	18.230 ± 0.059
	CFG-Zero*	$\omega = 1.0, K_{\text{zero}} = 1$	11.918 ± 0.298
		$\omega = 1.25, K_{\text{zero}} = 1$	11.327 ± 0.261
		$\omega = 1.5, K_{\text{zero}} = 1$	12.666 ± 0.303
		$\omega = 2.0, K_{\text{zero}} = 1$	18.698 ± 0.032
	CFG-MP	$\omega = 1.0, K_{\text{proj}} = 3$	24.475 ± 0.128
		$\omega = 1.25, K_{\text{proj}} = 3$	28.995 ± 0.130
		$\omega = 1.5, K_{\text{proj}} = 3$	33.313 ± 0.109
		$\omega = 2.0, K_{\text{proj}} = 3$	43.200 ± 0.302
	R-CFG++	$\lambda_{\max} = 1.0, \gamma = 1.0, \sigma = 0.0$	15.038 ± 0.356
	PathGuide (Ours)	seed_1	7.403 ± 0.281
		seed_2	7.384 ± 0.282
		seed_3	$\mathbf{7.375 \pm 0.285}$

Table 17: Endpoint path discrepancy at $t_i \approx 1$ for $T = 20$, our method is evaluated online where $M = N = 2^{11}$ against the plain conditional learned field under flow type is VP. We report mean $\pm$ sample standard deviation. Lower is better for all metrics.

Method	KL ↓	MMD-RBF^2 ↓	W_2 ↓
$\omega \equiv 1$	0.01261 ± 0.00184	0.00271 ± 0.00090	0.01755 ± 0.00420
Practical schedule	0.01232 ± 0.00216	0.00260 ± 0.00073	0.01569 ± 0.00340
Exact field u	0.01147 ± 0.00127	0.00258 ± 0.00059	0.01516 ± 0.00309