INFERENCE-TIME ATTRIBUTE DISTRIBUTION ALIGNMENT FOR UNCONDITIONAL DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time controllable generation is essential for real-world applications of unconditional diffusion models. However, most existing techniques focus on individual samples, struggling in applications that require the sample *population* to follow specific *attribute* distributions (*e.g.*, demographic balance or semantic proportions). We formalize this setting as the inference-time attribute distributional alignment problem for pretrained unconditional diffusion models. To address this, we cast inference-time attribute distributional alignment as an optimal control problem over the reverse diffusion process, viewing the process as the rollout of a dynamical system and augmenting it with additive, time-dependent perturbations as control. We solve for the perturbations using an optimal-control-based algorithm to optimize a differentiable distribution-matching objective while penalizing control effort to preserve data fidelity. Experiment results in image generation demonstrate that our proposed plug-and-play approach can better align attribute distributions to diverse and flexible test-time targets compared to baselines, without retraining or fine-tuning the pretrained diffusion model.[1]

1 INTRODUCTION

Diffusion models are powerful generative models that excel in representing complex data distributions and have been applied in a wide range of domains, including images (Ho et al., 2020; Rombach et al., 2022), videos (Ho et al., 2022; Bar-Tal et al., 2024; Hayakawa et al., 2025), graphs (Niu et al., 2020; Madeira et al., 2024; Luan et al., 2025), and robotics (Janner et al., 2022; Chi et al., 2025; Feng et al., 2025; Carvalho et al., 2025), among others. It is common that many user requirements are not known *a priori* during training and may shift over time during deployment. This makes it impractical to retrain or finetune the model for every new requirement, and in turn motivates a plethora of *inference-time* techniques that do not necessitate modifying the trained parameters of unconditional diffusion models for controllable generation, *e.g.*, guidance (Dhariwal & Nichol, 2021; Chung et al., 2023; Ye et al., 2024; Feng et al., 2024; Jiang et al., 2026), projection (Fishman et al., 2023; Zampini et al., 2025; Liang et al., 2025), joint correlation (Ruan et al., 2023; Zeng et al., 2024; Luan et al., 2026; Hao et al., 2025), *etc.*

In many controllable generation applications, however, the goal is not merely for each *individual* sample to satisfy a condition, but for the *sample population* to follow a desired *distribution* over *oracle-induced abstract attributes*, *i.e.*, the outputs obtained by applying an oracle function to generated samples. Such attributes may include styles or semantic categories in images, as identified by off-the-shelf classifiers, or behavior patterns exhibited by robotic trajectories, as determined by evaluation models. Controlling the *distribution* of such induced attributes is essential in many real-world scenarios. For example, fairness objectives in image generation involving human subjects naturally require balancing demographic attributes toward a uniform *distribution* (Choi et al., 2020; Parihar et al., 2024; Kang et al., 2025).

We refer to this goal as the inference-time diffusion *attribute distributional alignment* (DADA) problem: given a pretrained diffusion model, an *attribute oracle*, and a target attribute *distribution* specified only *at test time*, generate samples whose attribute distribution matches the target. This

[1] Implementation code and additional sample visualization are available at `https://anonymous.4open.science/r/diffusion-attribute-distribution-alignment/`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

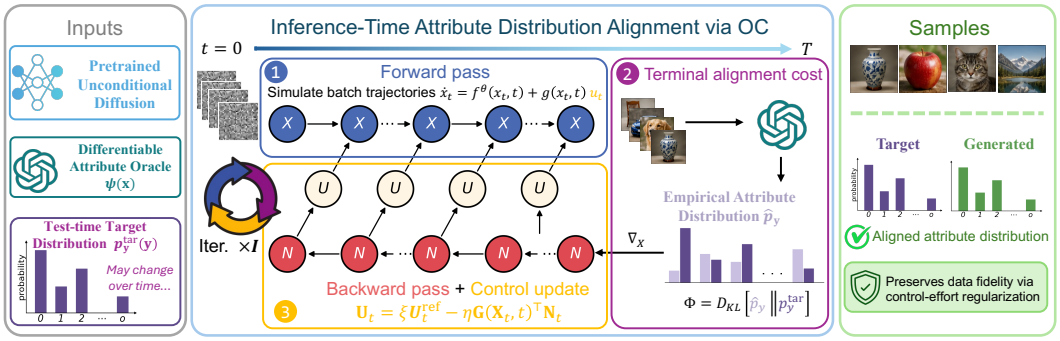


Figure 1: An overview of our proposed method for aligning attribute distribution.

problem is distinct from most common diffusion alignment and conditional generation settings (Wallace et al., 2024; Liu et al., 2024; Li et al., 2024; Uehara et al., 2025), because distributional alignment is a *population-level* objective and thus generally cannot be evaluated by a reward function $r(\mathbf{x})$ defined over a *single sample*. The most conceptually related topics include fairness-aware generation and sample diversity promotion in diffusion- and flow-based models. However, existing approaches (Kang et al., 2025; He et al., 2024; Choi et al., 2024; Friedrich et al., 2025; Jiang et al., 2025; Li et al., 2025; Morshed & Boddeti, 2025) are often tailored to text-to-image (T2I) diffusion models, require retraining or fine-tuning for new targets, or lack flexibility when the target attribute distribution changes at test time. To our knowledge, there are few generic methods that can align pretrained diffusion models to different test-time target attribute distributions in a plug-and-play fashion, *without retraining or fine-tuning the diffusion models or training extra components*.

In this work, we formulate attribute distributional alignment as an optimal control (OC) problem over the reverse diffusion process. Concretely, we view sampling from a pretrained diffusion model as rolling out a dynamical system that defines a prior over realistic samples, and augment its learned dynamics with an additive, time-dependent perturbation (see Figure 1). An OC-based algorithm then solves for perturbations that optimize a differentiable distribution-alignment objective. This OC perspective is appealing for several reasons: (i) It provides a *principled* mechanism for balancing distributional alignment and data fidelity through an explicit control-effort penalty. (ii) It is inherently *target-flexible* and naturally suited to *inference-time* use, since changing the target attribute distribution only modifies the objective while leaving the pretrained model intact. (iii) By optimizing the reverse trajectory as a whole, it avoids the single-step clean-sample approximation $\hat{\mathbf{x}}_0$ used by most inference-time guidance methods, which can be unreliable in the high-noise stages of the reverse process. (iv) It replaces hand-tuned, step-wise guidance-strength schedules, which often play a critical role in guidance methods, with control updates derived from OC optimality conditions.

We make the following contributions in this paper: (i) We formulate the attribute distributional alignment problem for diffusion models as an OC problem. (ii) We propose a practical inference-time, training-free algorithm for pretrained unconditional diffusion to align the *attribute distribution* with flexible target distributions. (iii) We demonstrate the empirical effectiveness of our method in aligning attribute distributions with test-time targets in image generation.

2 Preliminaries and Formulation

Notations. Let $n, m, o \in \mathbb{Z}^+$ be positive integers and $t \in \mathbb{R}_{\geq 0}$ be the time variable. Let $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^o$ be vector random variables. Vector constants are in bold symbol as $\boldsymbol{x} \in \mathbb{R}^n$. $\boldsymbol{I}_n$ stands for an $n \times n$ identity matrix. The Euclidean norm is denoted by $\|\cdot\|$. We note the probability density function of $\mathbf{x}$ (or probability mass function for discrete variables) by $p_{\mathbf{x}}(\mathbf{x})$, and may, for notational brevity, omit the subscript of random variable when it is clear from the context. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ stands for a Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$. Let $\mathbb{D}_{\text{KL}}\left[\cdot \,\|\, \cdot\right]$ denote the Kullback-Leibler (KL) divergence.

2.1 Preliminaries in Diffusion Models

Song et al. (2021b) describe a forward diffusion process with stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \quad t \in [0, T],$$

where $\boldsymbol{f}(\mathbf{x}_t, t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient, and $\mathbf{w}_t$ is the standard Wiener process. The reverse diffusion process, in which noises are transformed into data, is given by Anderson (1982); Song et al. (2021b) as

$$\mathrm{d}\mathbf{x}_t = [\boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t, \tag{1}$$

wherein $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is the *(Stein) score function* of the marginal distribution over $\mathbf{x}_t$ at time t, and $\bar{\mathbf{w}}_t$ is the standard Wiener process in reverse time. Song et al. (2021a) also identify the probability flow ordinary differential equations (PF-ODE) that yields the same marginal distribution over $\mathbf{x}_t$ at t as SDE Eq. (1):

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right] \mathrm{d}t. \tag{2}$$

For complex, high-dimensional distributions, the score function is analytically intractable in general, so it is often approximated by a neural network $\mathbf{s}^\theta(\mathbf{x}, t)$ parameterized by θ and learned via score matching (Song & Ermon, 2019). Equivalent to learning the score, the DDPM (Ho et al., 2020) and DDIM (Song et al., 2021a) formulations, in which the forward diffusion process is constructed as sequentially adding noise to the initial data distribution, learn a noise prediction neural network $\boldsymbol{\epsilon}^\theta(\mathbf{x}_t, t)$ for removing the noise during the reverse process to reconstruct the data distribution.

In this work, we focus on the *PF-ODE of the reverse diffusion process* and identify two instances; the details are deferred to Appendix **B.1**. Without loss of generality, we denote both in a unified form

$$\dot{\mathbf{x}}_t = \mathbf{f}^\theta(\mathbf{x}_t, t), \quad \mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n), \quad t \in [0, T]. \tag{3}$$

Remark 1 (Flow matching ODE). Eq. (3) is similar to the ODE in flow matching (FM) (Lipman et al., 2023; Liu et al., 2023). We also cover the details in Appendix **B.1**.

Steering diffusion at inference. A common paradigm of inference-time controllable generation is through classifier guidance (CG) (Dhariwal & Nichol, 2021), wherein the key is to obtain the conditional score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t \mid \mathbf{y})$ via adding a likelihood term $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t)$ to the pretrained score $\mathbf{s}^\theta(\mathbf{x}_t, t)$. Obtaining this term requires either additionally *training a noise-aware classifier*, or, as methods such as (Chung et al., 2023; Song et al., 2023), factorizing it over $\mathbf{x}_0$ $\nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) \approx \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \hat{\mathbf{x}}_0(\mathbf{x}_t))$ and approximating $\mathbf{x}_0$ by the posterior mean $\hat{\mathbf{x}}_0(\mathbf{x}_t) := \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$ via Tweedie's approach (Efron, 2011; Kim & Ye, 2021). The latter is training-free and plug-and-play, but the posterior mean is observed to be unreliable at the early stage of the reverse process (Li & Pereira, 2024). Therefore, the guidance strength schedule across time steps requires careful and often case-by-case tuning with handcrafted heuristics.

2.2 Problem Formulation: Diffusion Attribute Distribution Alignment

Suppose that there is a pretrained diffusion model $\mathbf{f}^\theta(\mathbf{x}, t)$ and the reverse diffusion process is as Eq. (3). Following Domingo-Enrich et al. (2024; 2025), we treat the diffusion PF-ODE as a dynamical system and further introduce a time-dependent, additive perturbation $\mathbf{u}_t \in \mathbb{R}^m$ called control into it:

$$\dot{\mathbf{x}}_t := \mathbf{f}^\theta(\mathbf{x}_t, t) + \mathbf{g}(\mathbf{x}_t, t)\mathbf{u}_t, \tag{4}$$

for $t \in [0, T]$, with $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^{n \times m}$ as an actuation field. In this work, we instantiate Eq. (4) under the EDM (Karras et al., 2022) and DDIM (Song et al., 2021a) formulations; details are deferred to Appendix **B.1**. We further let $p^u_{\mathbf{x}_T}$ denote the distribution of $\mathbf{x}_T$ sampled from process Eq. (4).

Remark 2 (Time Direction). From this point on, we follow the convention in dynamical systems and take the time direction as from 0 to T to describe *reverse* diffusion.

Remark 3 (Connection with diffusion guidance). The perturbed PF-ODE Eq. (4) can conceptually encompass classifier guidance: Let $\mathbf{g}(\mathbf{x}_t, t) \propto \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t)$ and let the control $\mathbf{u}_t := w_t$ be a scheduled scalar weight.

Given a continuously differentiable function $\Psi : \mathbb{R}^n \to \mathbb{R}^o$ as an *attribute oracle*[2], we note the attribute of interest as $\mathbf{y} = \Psi(\mathbf{x}_T)$. The distribution of $\mathbf{y}$ for samples generated by the perturbed

[2]The attribute oracle is a known function as part of the problem setting, akin to the setup in general *inverse problems* in diffusion generation (Song et al., 2022; Kawar et al., 2022; Chung et al., 2023).

process is the pushforward of $p^u_{\mathbf{x}_T}(\mathbf{x})$ by Ψ:

$$p^u_{\mathbf{y}}(\mathbf{y}) = \int_{\mathbb{R}^n} \delta\big(\mathbf{y} - \Psi(\mathbf{x})\big)\, p^u_{\mathbf{x}_T}(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

where $\delta(\cdot)$ is the Dirac-delta function.

Inference-time Attribute Distribution Alignment. Given an attribute oracle $\Psi(\mathbf{x})$ and a target attribute *distribution* $p^{\text{tar}}_{\mathbf{y}}$ at *test-time*, our goal is to find control $\mathbf{u}_t$ for $t \in [0, T]$ such that $p^u_{\mathbf{y}}(\mathbf{y})$ aligns with the target *without retraining or fine-tuning* the pretrained generative model $\mathbf{f}^\theta(\mathbf{x}, t)$, *i.e.*,

$$\min_{\mathbf{u}_t, t\in[0,T]} \mathbb{D}[p^u_{\mathbf{y}} || p^{\text{tar}}_{\mathbf{y}}] \qquad \text{s.t. Eq. (4)}, \tag{5}$$

where $\mathbb{D}[\cdot\,||\,\cdot]$ is a statistical distance or divergence between two distributions with the same support.

3 Optimal Control for Distribution Alignment

3.1 Review of Sample-wise Optimal Control

Let $\mathbf{x}$ denote the state and $\mathbf{u}$ the *control input* for a dynamical system. Let $L : \mathbb{R}^n \to \mathbb{R}$ and $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be continuously differentiable functions. Let $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ be a continuous vector function with continuous first partial derivatives w.r.t. the first argument. An OC problem is formulated as follows:

$$\min_{\mathbf{u}_t, t\in[0,T]} J(\mathbf{x}, \mathbf{u}) = L(\mathbf{x}_T) + \int_0^T \ell(\mathbf{x}_t, \mathbf{u}_t, t)\, \mathrm{d}t \tag{6a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, t) \tag{6b}$$

$$\mathbf{x}_0 = \boldsymbol{x}_{\text{init}}, \quad \mathbf{u}_t \in \mathcal{U}, \quad t \in [0, T], \tag{6c}$$

where $\mathcal{U} \subseteq \mathbb{R}^m$ is a nonempty closed set called the admissible control set. L is the terminal cost function, ℓ is the running cost or transient cost, and J is the cost functional. The constraint Eq. (6b) is the equation of motion of a dynamical system. See Fleming & Rishel (2012) for more rigorous definitions. The formulation Eq. (6) is explicitly minimizing a scalar total cost that encodes some desired effects at the terminal time and the control efforts exerted to the system, while simultaneously respecting the system dynamics, initial state conditions, and admissible control constraints.

3.2 Optimal Control Formulation for DADA

Finite-sample Batched OC. Different from previous work that applies OC to diffusion generation to achieve sample-wise objectives, our alignment objective in Eq. (5) naturally depends on multiple samples. As such, we stack the M-sample batches of states, controls, and costates (adjoints) into vectors: $\mathbf{X} := [{\mathbf{x}^{[1]}}^\top, \ldots, {\mathbf{x}^{[M]}}^\top]^\top \in \mathbb{R}^{Mn}$, $\mathbf{U} := [{\mathbf{u}^{[1]}}^\top, \ldots, {\mathbf{u}^{[M]}}^\top]^\top \in \mathbb{R}^{Mm}$, and define the concatenated dynamics $\mathbf{F}^\theta(\mathbf{X}, \mathbf{U}, t) := [{\mathbf{f}^\theta(\mathbf{x}^{[1]}, \mathbf{u}^{[1]}, t)}^\top, \ldots, {\mathbf{f}^\theta(\mathbf{x}^{[M]}, \mathbf{u}^{[M]}, t)}^\top]^\top \in \mathbb{R}^{Mn}$ and concatenated actuation $\mathbf{G}(\mathbf{X}, t) = \text{blkdiag}\left(\mathbf{g}(\mathbf{x}^{[1]}, t), \ldots, \mathbf{g}(\mathbf{x}^{[M]}, t)\right) \in \mathbb{R}^{Mn\times Mm}$. We stack the i.i.d. initial states $\mathbf{x}^{[i]}_{\text{init}} \sim \mathcal{N}(\mathbf{0},\, \boldsymbol{I}_n)$ as $\mathbf{X}_{\text{init}} \in \mathbb{R}^{Mn}$. We also write $\mathcal{U}^M := \mathcal{U} \times \cdots \times \mathcal{U} \subseteq \mathbb{R}^{Mm}$ and use $\Pi_{\mathcal{U}^M}$ for the component-wise projection onto $\mathcal{U}^M$.

With the above notations, we formulate the DADA problem as a finite-sample batched OC problem:

$$\min_{\mathbf{U}_t\in\mathcal{U}^M,\ t\in[0,T]} \Phi\left(\hat{p}^u_{\mathbf{y}}(\mathbf{X}_T)\right) + \frac{\rho}{2}\int_0^T \|\mathbf{U}_t\|^2\, \mathrm{d}t \tag{7a}$$

$$\text{s.t.} \qquad \dot{\mathbf{X}}_t = \mathbf{F}^\theta(\mathbf{X}_t, \mathbf{U}_t, t),\ \mathbf{X}_0 = \mathbf{X}_{\text{init}}, \quad \mathbf{U}_t \in \mathcal{U}^M \tag{7b}$$

where $\hat{p}^u_{\mathbf{y}}(\mathbf{X}_T)$ denotes attribute distribution estimated by the batched samples $\mathbf{X}_T$, and we adopt the reverse KL divergence against the target as the terminal cost:

$$\Phi\big(\hat{p}^u_{\mathbf{y}}(\mathbf{X}_T)\big) := \mathbb{D}_{\text{KL}}\left[\hat{p}^u_{\mathbf{y}}(\mathbf{X}_T)\,\|\,p^{\text{tar}}_{\mathbf{y}}\right]. \tag{8}$$

While the dynamics Eq. (7b) are separable across samples, Eq. (8) couples the batch through $\hat{p}^u_{\mathbf{y}}(\mathbf{X}_T)$.

3.3 Optimal Controller for DADA OC

Problem Eq. (7) is an OC problem in $\mathbb{R}^{Mn}$. Define the Hamiltonian associated with Eq. (7) as

$$\widetilde{H}(\mathbf{X}_t, \mathbf{N}_t, \mathbf{U}_t, t) := \frac{\rho}{2}\|\mathbf{U}_t\|^2 + \mathbf{N}_t^\top \mathbf{F}^\theta(\mathbf{X}_t, \mathbf{U}_t, t)$$

where $\mathbf{N}_t = [{\boldsymbol{\nu}_t^{[1]}}^\top, \ldots, {\boldsymbol{\nu}_t^{[M]}}^\top]^\top \in \mathbb{R}^{Mn}$ stacks the adjoint (costate) vectors. By Pontryagin's Maximum Principle (PMP) (Fleming & Rishel, 2012; Kirk, 2004), the necessary conditions for an optimal trajectory $(\tilde{\mathbf{X}}_t^*, \tilde{\mathbf{N}}_t^*, \tilde{\mathbf{U}}_t^*)$ of Eq. (7) are:

$$\dot{\tilde{\mathbf{X}}}_t^* = \mathbf{F}^\theta(\tilde{\mathbf{X}}_t^*, \tilde{\mathbf{U}}_t^*, t), \tag{9a}$$

$$\dot{\tilde{\mathbf{N}}}_t^* = -\nabla_{\mathbf{X}}\mathbf{F}^\theta(\tilde{\mathbf{X}}_t^*, \tilde{\mathbf{U}}_t^*, t)^\top \tilde{\mathbf{N}}_t^*, \qquad \tilde{\mathbf{N}}_T^* = \nabla_{\mathbf{X}}\Phi\left(\hat{p}_{\mathbf{y}}^u(\tilde{\mathbf{X}}_T^*)\right), \tag{9b}$$

$$\tilde{\mathbf{U}}_t^* \in \underset{\mathbf{U}\in\mathcal{U}^M}{\arg\min}\ \widetilde{H}(\tilde{\mathbf{X}}_t^*, \tilde{\mathbf{N}}_t^*, \mathbf{U}, t). \tag{9c}$$

Since $\mathbf{F}^\theta$ concatenates per-sample dynamics Eq. (6b), its Jacobian is block-diagonal:

$$\nabla_{\mathbf{X}}\mathbf{F}^\theta(\mathbf{X}, \mathbf{U}, t) = \mathrm{blkdiag}\big(\mathbf{J}_t^{[1]}, \ldots, \mathbf{J}_t^{[M]}\big) \in \mathbb{R}^{Mn\times Mn}, \tag{10}$$

where $\mathbf{J}_t^{[i]} := \nabla_{\mathbf{x}}\mathbf{f}^\theta(\mathbf{x}_t^{[i]}, \mathbf{u}_t^{[i]}, t) \in \mathbb{R}^{n\times n}$ is the standard Jacobian of the per-sample vector field w.r.t. the state. Consequently, the adjoint dynamics Eq. (9b) decouple across samples: $\dot{\boldsymbol{\nu}}_t^{*,[i]} = -(J_t^{[i]})^\top \boldsymbol{\nu}_t^{*,[i]}$ for each i. The only cross-sample coupling enters through the terminal condition, which depends on all M terminal states via the batch cost Φ.

However, jointly solving all the conditions in Eq. (9) is essentially a two-point boundary value problem; for general nonlinear, nonconvex dynamics, it is challenging to solve. Rather than directly solving them in joint, we follow Li et al. (2018); Wang et al. (2025) and employ the Extended Method of Successive Approximations (E-MSA) to solve a *proximalized* OC subproblem in an iterative fashion for stability. Specifically, given a reference control trajectory $\boldsymbol{U}_t^{\mathrm{ref}}$, we consider a subproblem

$$\min_{\mathbf{U}_t\in\mathcal{U}^M}\ \Phi\big(\hat{p}_{\mathbf{y}}^u(\mathbf{X}_T)\big) + \frac{1}{2}\int_0^T \left(\rho\|\mathbf{U}_t\|^2 + \gamma\|\mathbf{U}_t - \boldsymbol{U}_t^{\mathrm{ref}}\|^2\right)\mathrm{d}t, \quad \text{s.t. Eq. (7b)} \tag{11}$$

where γ is a hyperparameter. Applying PMP to this *proximalized* subproblem yields the *necessary* conditions akin to Eq. (9) but in terms of the extended Hamiltonian $H(\mathbf{X}_t, \mathbf{N}_t, \mathbf{U}_t, t) := \widetilde{H}(\mathbf{X}_t, \mathbf{N}_t, \mathbf{U}_t, t) + \frac{\gamma}{2}\|\mathbf{U}_t - \boldsymbol{U}_t^{\mathrm{ref}}\|^2$ and the optimal trajectories $(\mathbf{X}_t^*, \mathbf{N}_t^*, \mathbf{U}_t^*)$ that solves the *proximalized* problem. Accordingly,

$$\mathbf{U}_t^* \in \underset{\mathbf{U}\in\mathcal{U}^M}{\arg\min}\ H(\mathbf{X}_t^*, \mathbf{N}_t^*, \mathbf{U}, t). \tag{12}$$

Closed-form Control Update. Since the running cost is quadratic in $\mathbf{U}$ and the dynamics are control-affine, Eq. (12) admits a *closed-form* minimizer:

$$\mathbf{U}_t^* = \Pi_{\mathcal{U}^M}\left(\xi\boldsymbol{U}_t^{\mathrm{ref}} - \eta\,\mathbf{G}\left(\mathbf{X}_t^*, t\right)^\top \mathbf{N}_t^*\right), \tag{13}$$

where $\xi := \frac{\gamma}{\rho+\gamma}$ and $\eta := \frac{1}{\rho+\gamma}$.

Proposition 3.1. *For any fixed $t \in [0, T]$ and fixed $(\boldsymbol{x}_t^{[i]}, \boldsymbol{\nu}_t^{[i]})$, with $\rho + \gamma > 0$, (1) the sample-wise optimal control $\mathbf{u}_t^{*,[i]}$ is given by*

$$\mathbf{u}_t^{*,[i]} \in \Pi_{\mathcal{U}}\left(\bar{\boldsymbol{u}}_t^{[i]}\right) := \underset{\mathbf{u}\in\mathcal{U}}{\arg\min}\left\|\mathbf{u} - \bar{\boldsymbol{u}}_t^{[i]}\right\|^2, \text{ where } \bar{\boldsymbol{u}}_t^{[i]} := \frac{1}{\rho+\gamma}\left(\gamma\,\boldsymbol{u}_t^{\mathrm{ref,[i]}} - \mathbf{g}\left(\boldsymbol{x}_t^{[i]}, t\right)^\top \boldsymbol{\nu}_t^{[i]}\right),$$

and (2) $\Pi_{\mathcal{U}}\left(\bar{\boldsymbol{u}}_t^{[i]}\right)$ is nonempty. Further, $\mathbf{u}_t^{*,[i]}$ is unique if $\mathcal{U}$ is convex.

The proof of Proposition 3.1 is in Appendix **C.1**. Proposition 3.1 indicates that the optimal control at each time is a function of the state and adjoint at that time. The resulting *discrete-time* algorithm (see Algorithm 1 in Appendix **B.2**) performs the following three steps iteratively: (i) Forward pass: Simulate Eq. (7b) in forward time. (ii) Backward pass: Evaluate the distribution-alignment cost, take the gradients, and simulate the adjoint dynamics in backward time. (iii) Control update with Eq. (13). An illustration is in Figure 1. In practice, we adopt Euler discretizations for the forward and backward passes and leverage the vector-Jacobian product (VJP) at backward pass without materializing the gigantic dynamics Jacobian. We analyze the computational complexity in Appendix **B.3**.

Algorithmic Stationarity. The closed-form update Eq. (13) can be interpreted as projected gradient descent on a time-weighted Euler discretization of Eq. (7), yielding a non-asymptotic first-order stationarity rate. Let J_h denote the Euler-discretized version of the objective Eq. (7a) on a time grid with step sizes $\{h_k\}$, and let $\mathbf{G}_\eta(\mathbf{U})$ be the projected-gradient mapping for J_h on $\mathcal{C} := (\mathcal{U}^M)^K$ in the time-weighted norm $\|\mathbf{U}\|_h^2 := \sum_k h_k \|\mathbf{U}_k\|^2$. Formal definitions are deferred to Appendix **C.2**.

Proposition 3.2 (Algorithmic stationarity)**.** *Under the regularity conditions detailed in Appendix* **C.2***, there exists a finite $L > 0$, such that whenever $\eta := (1-\xi)/\rho \le 1/L$ with $\xi \in [0,1)$, the iterates $\{\mathbf{U}^{(r)}\}_{r\ge 0}$ produced by Algorithm 1 satisfy*

$$\min_{0\le r<I} \left\|\mathbf{G}_\eta(\mathbf{U}^{(r)})\right\|_h^2 \;\le\; \frac{2\big(J_h(\mathbf{U}^{(0)}) - J_h^{\inf}\big)}{\eta\, I}, \qquad \forall\, I \ge 1, \tag{14}$$

where $J_h^{\inf} := \inf_{\mathbf{U}\in\mathcal{C}} J_h(\mathbf{U})$. Moreover, $\mathbf{G}_\eta(\mathbf{U}^{(r)}) \to 0$ along the full sequence, and every cluster point $\mathbf{U}^*$ satisfies $\mathbf{G}_\eta(\mathbf{U}^*) = 0$.

The proof of Proposition 3.2 is in Appendix **C.2**.

Differentiable Approximations for Terminal Cost. For discrete attributes in practice, the attribute oracle is often a continuously differentiable neural network that maps an input $\mathbf{x}$ to an attribute-logit vector. To maintain differentiability, we *estimate* the empirical distribution $\hat{p}^u_{\mathbf{y}}$ over a batch of M generated samples by averaging their softmax probabilities across the batch: $\hat{p}^u_{\mathbf{y}} = \frac{1}{M}\sum_{i=1}^M \text{softmax}\left(\Psi\left(\mathbf{x}_T^{[i]}\right)\right)$. The terminal cost is then computed as the element-wise KL divergence: $\Phi\left(\hat{p}^u_{\mathbf{y}}\right) = \mathbb{D}_{\text{KL}}\left[\hat{p}^u_{\mathbf{y}} \,\|\, p^{\text{tar}}_{\mathbf{y}}\right] = \sum_{j=1}^o \hat{p}^{u,(j)}_{\mathbf{y}} \log(\hat{p}^{u,(j)}_{\mathbf{y}} / p^{\text{tar},(j)}_{\mathbf{y}})$, where o is the total number of attribute classes, and the superscript (j) indexes the probability mass associated with the j-th class.

Population Alignment. We provide theoretical guarantees on attribute distribution alignment. The following proposition relates the realized terminal cost to the expected *soft* attribute distribution induced by the algorithm. Let $\hat{p} := \hat{p}^u_{\mathbf{y}}(\mathbf{X}_T)$ be the *empirical soft* attribute distribution of the returned sample batch, $\hat{\varepsilon} := \mathbb{D}_{\text{KL}}\left[\hat{p} \,\|\, p^{\text{tar}}_{\mathbf{y}}\right]$, and $\bar{p}_M := \mathbb{E}[\hat{p}]$. Detailed definitions are in Appendix **C.3.3**.

Proposition 3.3 (Soft attribute alignment)**.** *Under the assumptions detailed in Appendix* **C.3.2***, run* ***Algorithm 1*** *with projected controls for a fixed iteration I. Then, with probability at least $1-\delta$, $\|\hat{p} - \bar{p}_M\|_1 \le \varepsilon_{\text{rep}}(M,\delta)$, $\text{TV}(\bar{p}_M, p^{\text{tar}}_{\mathbf{y}}) \le \sqrt{\frac{\hat{\varepsilon}}{2}} + \frac{1}{2}\varepsilon_{\text{rep}}(M,\delta)$ where $\varepsilon_{\text{rep}}(M,\delta)$ is a constant depending on δ and batch size M.*

The proof of Proposition 3.3 is in Appendix **C.3**. For produced sample batches, the terminal cost also gives a deterministic bound for the *hard-label* attribute histogram; see Appendix **C.3.6** for details.

4 Experiments

We address the following research questions through experiments in image generation:

RQ 1: How effective is our method at aligning attribute distributions toward test-time targets?

RQ 2: Can our method better preserve sample quality while achieving distributional alignment?

We evaluate our method in two settings: (i) CIFAR-100 tests whether the method can match flexible target distributions over semantic labels with different support sizes; (ii) Human face generation tests fairness-motivated attribute alignment over age, gender, race, and their joint distributions, and further examines whether the same formulation and our method applies to flow matching in latent space.

4.1 CIFAR-100 with Hierarchical Semantic Classes

We first test on generating low-resolution images across semantic classes. We treat the semantic class of an image as attribute and consider flexible test-time target distributions with different support sizes.

Setup: We adopt an unconditional EDM (Karras et al., 2022) trained on the CIFAR-100 dataset (Krizhevsky et al., 2009) as the base diffusion model and the attribute oracle Ψ is instantiated as ResNet (He et al., 2016) image classifiers. We test our method on three different levels of class labels with a coarse-to-fine hierarchy: `meta5`, `coarse`, and `fine`, with 5, 20, and 100 classes, respectively. We choose three different attribute distributions as test-time targets for each level: Uniform, ZigZag, and Gaussian. Figure 3 in Appendix **D.1** shows instances of the target distributions.

Table 1: Quantitative evaluation metrics on `CIFAR-100`. Best in **bold**, second-best underlined.

METHOD	meta5 TV↓	meta5 JS↓	meta5 χ^2↓	meta5 FID↓	coarse TV↓	coarse JS↓	coarse χ^2↓	coarse FID↓	fine TV↓	fine JS↓	fine χ^2↓	fine FID↓
Gaussian												
EDM	0.252	0.196	0.0749	17.4	0.274	0.227	0.0988	17.4	0.248	0.231	0.101	16.1
PG-DPS	0.154	0.133	0.0350	21.3	0.195	0.165	0.0528	16.0	0.348	0.299	0.165	33.1
SMPL+DPS	0.140	0.130	0.0334	32.5	0.228	0.204	0.0793	20.3	0.248	0.229	0.0992	16.7
OURS	**0.136**	**0.117**	**0.0272**	**16.5**	**0.176**	**0.146**	**0.0421**	**15.7**	**0.184**	**0.173**	**0.0573**	**13.7**
Zigzag												
EDM	0.067	0.0574	0.00658	15.3	0.185	0.148	0.0433	16.1	0.24	0.205	0.0812	16.1
PG-DPS	0.113	0.1	0.0199	22	0.133	0.116	0.0267	15.7	0.341	0.288	0.156	33.8
SMPL+DPS	0.0681	0.0573	0.00655	15.6	0.155	0.129	0.0331	16.4	0.230	0.197	0.0755	16.7
OURS	**0.0544**	**0.0495**	**0.00489**	**14**	**0.103**	**0.0927**	**0.0171**	**14.8**	**0.171**	**0.15**	**0.0442**	**12.6**
Uniform												
EDM	0.083	0.0651	0.00845	15.5	0.132	0.114	0.0255	15.5	0.186	0.159	0.0495	15.5
PG-DPS	0.0692	0.0529	0.00559	22.4	0.0805	0.0742	0.0109	15.2	0.287	0.255	0.123	30.7
SMPL+DPS	0.0750	0.0598	0.00714	16.8	0.106	0.100	0.0199	15.4	0.177	0.155	0.0471	16.1
OURS	**0.0281**	**0.0292**	**0.00171**	**13.1**	**0.069**	**0.0655**	**0.00853**	**12.6**	**0.141**	**0.12**	**0.0284**	**12.6**

Baselines: We compare our method with vanilla EDM, which corresponds to i.i.d. sampling from the learned distribution, and two guidance-based methods: Particle Guidance (PG) (Corso et al., 2024), and a naive two-stage method (SMPL+DPS)[3]. For PG, the guidance potential is the same terminal cost in our method but applied on the estimated terminal sample $\hat{\mathbf{x}}_T$ via Tweedie's formula (Efron, 2011), akin to DPS. See all implementation details in Appendix **D.1**.

Evaluation Metrics: We use several statistical distances/divergences to quantify semantic-class distribution alignment: Total Variation (TV), Jensen–Shannon divergence (JS), and the χ^2 distance[4]. We use the Fréchet Inception Distance (FID) to evaluate image quality.[5]

Results: Table 1 presents comprehensive quantitative comparisons across methods and target attribute distributions with different support sizes. Across all tested settings, our method achieves the best performance in terms of both attribute-distribution alignment and sample quality. Surprisingly, PG performs even worse than vanilla EDM in some cases (*e.g.*, ZigZag-`meta5` and Gaussian-`fine`). This suggests that naively applying guidance-style approaches with a batch-wise distributional loss may be insufficient for attribute distribution alignment. Qualitative results and several ablation studies regarding oracle architectures and sampling steps are in Appendix **E.1**.

4.2 HUMAN FACE GENERATION

We aim to achieve human-face image generation with *fairness and controllable ratios* across ages, genders, and races, mitigating potential biases introduced by pretrained diffusion models *without retraining or fine-tuning them*. Beyond the standard diffusion paradigm, we also empirically demonstrate the applicability of our method in latent flow matching.

Setup: We use a DDIM (Song et al., 2021a) pretrained by Choi et al. (2022) and a pretrained Latent Flow Matching (LFM) (Dao et al., 2023) model; both models were trained on the FFHQ-256 dataset (Karras et al., 2019). We consider three attributes in this task: gender, race, and age. For gender, we consider {`Female`, `Male`}; for race, we adopt the 4-way classification by Karkkainen & Joo (2021): {`Asian`, `Black`, `Indian`, `WMELH`}[6]; for age, we partition it into three groups: {`Junior`, `Middle`, `Senior`}[7]. We test with both fairness targets and customized skewed targets for each single attribute and *joint* attributes. The attribute oracle is also instantiated as an image classifier. We implement our method under both diffusion and latent flow matching settings. See implementation details in Appendix **D.2**.

[3]It constructs target-label counts matching the target distribution up to rounding, randomly permutes the labels across initial noises, and independently guides each sample to its target label by DPS (Chung et al., 2023).

[4]We adopt the symmetric χ^2 distance (Markatou et al., 2018): $d_{\chi^2}(P,Q) = \frac{1}{2}\sum_{i=1}^{K}(P_i - Q_i)^2/(P_i + Q_i)$ for two discrete distributions P and Q with the same finite support.

[5]FID references are computed via sampling from the dataset according to each target attribute distribution.

[6]`WMELH` stands for the merge of White, Middle Eastern, and Latino Hispanic.

[7]`Junior`: 0–19 years old; `Middle`: 20–50; `Senior`: 50–120.

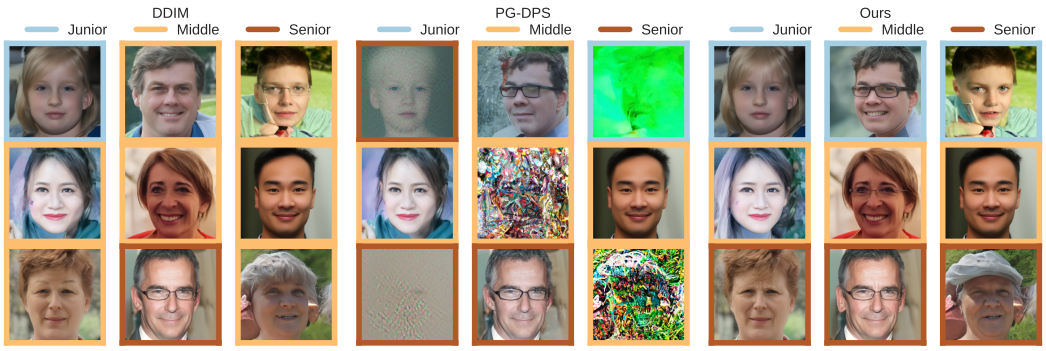


Figure 2: Qualitative samples *from a single batch* of our method (right), compared to vanilla DDIM (left) and PG (mid) for generating human faces with a *fair* target distribution over age groups (Uniform-`age`), where the ratio of faces in three age groups should be 1:1:1. The pretrained diffusion model learned a highly imbalanced age distribution from the FFHQ dataset, where most faces are classified as `Middle`. Our approach aligns the generated attribute distribution with the target by minimally editing facial details such as wrinkles and face shapes. See more samples in Appendix **E.2**.

Aligning Joint Attribute Distribution: For *joint* attributes, the alignment target is the factorized joint distribution $p^{\text{tar}}_{\mathbf{y}}(\mathbf{y}) \propto \prod_{i=1}^{N} p^{\text{tar}}_{\mathbf{y}_i}(\mathbf{y}_i)$, where we *assume independence among all attributes*[8]. Note that this independence assumption generally does *not* hold for the generated attribute distribution $p^{u}_{\mathbf{y}}(\mathbf{y})$. With independence, the terminal cost decomposes as $\mathbb{D}_{\text{KL}}\left[\hat{p}^{u}_{\mathbf{y}} \,\|\, p^{\text{tar}}_{\mathbf{y}}\right] = \sum_{i=1}^{N} \mathbb{D}_{\text{KL}}\left[\hat{p}^{u}_{\mathbf{y}_i} \,\|\, p^{\text{tar}}_{\mathbf{y}_i}\right] + \sum_{i=1}^{N} \text{H}(\hat{p}^{u}_{\mathbf{y}_i}) - \text{H}(\hat{p}^{u}_{\mathbf{y}})$, where $\hat{p}^{u}_{\mathbf{y}_i}$ is the empirical marginal, and $\text{H}(\cdot)$ denotes entropy.

Baselines: We compare our method with vanilla DDIM, LFM (Dao et al., 2023), SMPL+DPS, and PG. Details are in Appendix **D.2**.

Evaluation Metrics: As in the previous experiment, we measure attribute-distribution alignment using JS, TV, and χ^2, and we use FID and CMMD (Jayasumana et al., 2024) (see Apdx. E.2.2) to evaluate image quality. We also measure the Fairness Discrepancy (FD), following prior work on fair generation (Choi et al., 2020; Parihar et al., 2024): $\text{FD} := \left\| p^{\text{tar}}_{\mathbf{y}} - \mathbb{E}_{\mathbf{x}\sim p^{\theta,\mathbf{u}}_{\mathbf{x}}(\mathbf{x})} \hat{p}_{\mathbf{y}}(\mathbf{y} \mid \mathbf{x}) \right\|$, where $\hat{p}_{\mathbf{y}}(\mathbf{y} \mid \mathbf{x})$ is the *softmax output* of the attribute oracle.

Table 2: Quantitative results for face generation with *joint attribute control*. Best in **bold**, second-best underlined.

	`age`×`gender`×`race`				
METHOD	TV↓	JS↓	χ^2↓	FD↓	FID↓
UniformJoint					
DDIM	0.566	0.477	0.393	0.308	46.26
PG-DPS	0.444	0.404	0.281	0.238	127.0
SMPL+DPS	0.294	0.264	0.131	0.149	198.2
LFM	0.584	0.251	0.424	0.359	46.20
OURS	0.112	0.107	0.0225	0.0606	**41.28**
OURS-F	**0.0963**	**0.00783**	**.0155**	**.0504**	41.54
CustomJoint $[5:2:3] \times [3:7] \times [4:3:2:1]$					
DDIM	0.493	0.444	0.337	0.305	44.28
PG-DPS	0.531	0.442	0.335	0.287	74.77
SMPL+DPS	0.268	0.247	0.114	0.147	89.51
LFM	0.500	0.213	0.361	0.346	44.05
OURS	0.142	0.131	0.034	0.090	42.19
OURS-F	**0.0968**	**0.00674**	**0.0134**	**0.0493**	**38.65**

Results: Table 2 report alignment performance and sample quality for individual and joint target attribute distributions, respectively, with qualitative samples shown in Figure 2. For Uniform targets, pretrained DDIM and LFM show pronounced age and race biases inherited from training data. PG reduces these biases but at a large cost in image quality. In contrast, our method under both diffusion and latent flow matching paradigms achieves better distributional alignment across metrics while preserving sample quality. Under the Custom targets with strongly skewed distributions, our method performs consistently better in matching the target distributions *and* maintains sample quality, whereas PG yields inconsistent performance given different targets. Similar trends hold for joint alignment: PG trades quality for alignment and does not reliably outperform vanilla DDIM, while our method improves alignment without degrading sample quality. See more results in Appendix **E.2**.

[8]Note that this independence assumption is only for the convenience of experiment design, but *not* a restriction of our method, as one can always directly compute the joint KL.

Table 3: Quantitative evaluation metrics on face generation with controlling *single* attribute. Best in **bold**, second-best underlined. OURS-F denotes latent-flow-based version of our method.

METHOD	age TV↓	JS↓	χ^2↓	FD↓	FID↓	gender TV↓	JS↓	χ^2↓	FD↓	FID↓	race TV↓	JS↓	χ^2↓	FD↓	FID↓
Uniform															
DDIM	.21	.15	.044	.25	47.18	.042	.029	.0017	.052	47.28	.56	.45	.36	.65	52.38
PG-DPS	.15	.13	.033	.18	102.1	.043	.030	.0018	.047	75.67	.37	.32	.19	.45	101.8
SMPL+DPS	.028	.024	.0011	.036	130.7	.014	.0096	.00018	.037	54.97	.22	.18	.062	.27	69.56
LFM	.28	.042	.083	.34	45.34	.16	.013	.026	.22	44.07	.57	.20	.36	.66	48.41
OURS	.023	.018	6.4e-04	.028	45.18	.0052	.0037	2.7e-05	.0087	44.97	.028	.025	.0012	.038	47.59
OURS-F	**.018**	**1.8e-04**	**3.6e-04**	**.023**	**43.60**	**0.00**	**0.00**	**0.00**	**5.2e-04**	**42.67**	**.024**	**3.0e-04**	**6.0e-04**	**.026**	**46.12**
Custom-1															
	$[4:1:3]$					$[2:8]$					$[4:3:2:1]$				
DDIM	.42	.32	.20	.51	49.65	.26	.20	.076	37	48.58	.41	.35	.22	.50	48.71
PG-DPS	.38	.31	.18	.46	123.1	.24	.18	.065	.33	129.6	.27	.27	.13	.30	165.3
SMPL+DPS	.055	.056	.0062	.11	95.05	.020	.017	.00059	.020	70.64	.18	.14	.039	.23	61.74
LFM	.49	.14	.26	.60	51.95	.14	.013	.025	.20	45.87	.42	.13	.23	.51	48.80
OURS	.049	.035	.0025	.063	**46.14**	.0094	.0082	1.4e-4	.017	46.52	.043	.040	.0032	.062	**43.03**
OURS-F	**.016**	**1.4e-4**	**2.7e-4**	**.018**	48.25	**.0073**	**4.1e-5**	**8.2e-5**	**.010**	**45.76**	**.029**	**4.9e-4**	**9.8e-4**	**.034**	44.73
Custom-2															
	$[2:3:4]$					$[7:3]$					$[1:1:4:4]$				
DDIM	.23	.18	.066	.32	51.39	.24	.17	.060	.33	55.74	.70	.56	.53	.84	51.45
PG-DPS	.30	.24	.11	.36	90.79	.043	.032	.0021	.084	156.9	.31	.28	.14	.36	253.2
SMPL+DPS	.042	.038	.0029	.038	114.6	.023	.017	.00061	.059	60.62	.25	.23	.10	.34	72.94
LFM	.29	.058	.11	.40	51.50	.36	.066	.13	.51	56.16	.71	.30	.53	.84	51.12
OURS	**.013**	.012	**2.8e-4**	**.018**	**46.25**	**1.2e-8**	**1.0e-4**	**1.7e-16**	**.0011**	**48.69**	.053	.041	.0034	.075	46.53
OURS-F	.022	**2.8e-4**	5.6e-4	.031	47.83	.020	2.3e-4	4.6e-4	.029	51.22	**.016**	**1.8e-4**	**3.7e-4**	**.020**	**46.09**

5 RELATED WORK

More detailed related work is deferred to Appendix **A**.

Diffusion Models with Conditional Generation. Diffusion models define the generative process as iterative denoising (Ho et al., 2020) or score-based Langevin dynamics (Song et al., 2021b). DDIM (Song et al., 2021a) and EDM (Karras et al., 2022) improved sampling efficiency and design clarity. Dhariwal & Nichol (2021) introduced classifier guidance to steer pretrained models, while Ho & Salimans (2022) proposed classifier-free guidance for joint training with conditioning signals.

Fairness and Diversity in Diffusion. Fairness generation with diffusion is an instance of the DADA problem. Shen et al. (2024) propose a supervised fine-tuning method using distributional alignment loss for finetuning to mitigate demographic biases in T2I diffusion models. Friedrich et al. (2025) conduct a preliminary model audit to generate a lookup table of biased prompts and attributes, and then employs Semantic Guidance to promote fair attribute classes. Parihar et al. (2024) trains a predictor to map h-space features to attribute distributions during the denoising diffusion process. Corso et al. (2024) propose a training-free guidance-based method for promoting sample diversity in diffusion models. However, most of the above methods are specifically designed for T2I conditional models and leverage text conditioning mechanisms that are baked inside the T2I models, thus not generalizable to unconditional models nor to other domains.

Optimal Control in Diffusion- and Flow-based Models. Theoretical connections between Stochastic Optimal Control (SOC) and diffusion were early established in (Berner et al., 2024). Domingo-Enrich et al. (2024; 2025) propose Adjoint Matching and SOCM for fine-tuning diffusion- and flow-based models by learning optimal control fields via regression objectives. In contrast, training-free methods such as OC-Flow (Wang et al., 2025) and DTM (Pandey et al., 2025) steer pre-trained diffusion or flow models by differentiating through the generative ODE or SDE to minimize control costs without updating model parameters. While our method resonates with these works in terms of methodology, none of them are addressing the attribute distribution alignment problem.

6 DISCUSSIONS AND CONCLUSION

We formulate and study the attribute distribution alignment problem for pretrained unconditional diffusion models. We propose an inference-time, plug-and-play method that does not require any extra training or fine-tuning. Our results show that the proposed method is effective in aligning the generation attribute distribution to flexible test-time targets, while better preserving sample quality compared to training-free baselines. Limitations of our method include extra computational overheads and the need for a moderate batch size for attribute distribution estimation. Detailed discussions are in Appendix **F**. Future work includes extensions to finding optimal guidance weights in conditional diffusion models and applying our approach in other domains, *e.g.*, robotics, graphs, *etc.*

Reproducibility Statement

An anonymous link to code implementations of the proposed method on partial experiments are provided in the supplementary materials. Details of hyperparameters are described in the appendix and reflected in the code. We will open-source all code and relevant trained model weights upon acceptance.

AI Use Statement

We used generative AI tools for proof checking and language polishing, with further human verification and edition.

References

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

Iskander Azangulov, Peter Potaptchik, Qinyu Li, Eddie Aamari, George Deligiannidis, and Judith Rousseau. Adaptive diffusion guidance via stochastic optimal control. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4087–4095. PMLR, 02–05 May 2026. URL https://proceedings.mlr.press/v300/azangulov26a.html.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia*, pp. 94:1–94:11, 2024. URL https://doi.org/10.1145/3680528.3687614.

R.G. Bartle and D.R. Sherbert. *Introduction to Real Analysis*. Wiley, 2011. ISBN 9780471433316. URL https://books.google.com.sg/books?id=ulveQwAACAAJ.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics. Springer Cham, 2 edition, 2017. ISBN 978-3-319-48310-8. doi: 10.1007/978-3-319-48311-5.

A. Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2017. ISBN 9781611974980. URL https://books.google.com.sg/books?id=xLk4DwAAQBAJ.

Amir Beck. *Introduction to Nonlinear Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014. doi: 10.1137/1.9781611973655. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611973655.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=oYIjw37pTP.

Eric Tillmann Bill, Enis Simsar, and Thomas Hofmann. Optimal control meets flow matching: A principled route to multi-subject fidelity. *arXiv preprint arXiv:2510.02315*, 2025.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. ISBN 9780199535255. doi: 10.1093/acprof:oso/9780199535255.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199535255.001.0001.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

João Carvalho, An Thai Le, Piotr Kicki, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and adapting robot motion planning with diffusion models. *IEEE Transactions on Robotics*, 41:4881–4901, 2025. doi: 10.1109/TRO.2025.3593109.

Yaofo Chen. Pytorch CIFAR models. https://github.com/chenyaofo/pytorch-cifar-models, 2025. Accessed: 2025-5-17.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11472–11481, 2022.

Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1887–1898. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/choi20a.html.

Yujin Choi, Jinseong Park, Hoki Kim, Jaewook Lee, and Saerom Park. Fair sampling in diffusion models through switching mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 21995–22003, 2024.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi S. Jaakkola. Particle guidance: non-i.i.d. diverse sampling with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KqbCvIFBY7.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

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

Carles Domingo-Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky T. Q. Chen. Stochastic optimal control matching. In *Advances in Neural Information Processing Systems*, volume 37, pp. 112459–112504, 2024. doi: 10.52202/079017-3573.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Zeyu Feng, Hao Luan, Pranav Goyal, and Harold Soh. LTLDoG: Satisfying temporally-extended symbolic constraints for safe diffusion-based planning. *IEEE Robotics and Automation Letters*, 9 (10):8571–8578, 2024. doi: 10.1109/LRA.2024.3443501.

Zeyu Feng, Hao Luan, Kevin Yuchen Ma, and Harold Soh. Diffusion meets options: Hierarchical generative skill composition for temporally-extended tasks. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10854–10860, 2025. doi: 10.1109/ICRA55743.2025.11127641.

Nic Fishman, Leo Klarner, Emile Mathieu, Michael Hutchinson, and Valentin De Bortoli. Metropolis sampling for constrained diffusion models. *Advances in Neural Information Processing Systems*, 36:62296–62331, 2023.

Wendell H Fleming and Raymond W Rishel. *Deterministic and stochastic optimal control*, volume 1. Springer Science & Business Media, 2012.

Felix Friedrich, Manuel Brack, Lukas Struppek, Dominik Hintersdorf, Patrick Schramowski, Sasha Luccioni, and Kristian Kersting. Auditing and instructing text-to-image generation models on fairness. *AI and Ethics*, 5(3):2103–2123, Jun 2025. ISSN 2730-5961. doi: 10.1007/s43681-024-00531-5.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *Advances in Neural Information Processing Systems*, 37:90736–90770, 2024.

Yinbin Han, Meisam Razaviyayn, and Renyuan Xu. Stochastic control for fine-tuning diffusion models: Optimality, regularity, and convergence. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DnNV3Ea09e.

Ce Hao, Anxing Xiao, Zhiwei Xue, and Harold Soh. CHD: Coupled hierarchical diffusion for long-horizon tasks. In *9th Annual Conference on Robot Learning*, 2025. URL https://openreview.net/forum?id=tXY6VQlXfA.

Akio Hayakawa, Masato Ishii, Takashi Shibuya, and Yuki Mitsufuji. MMDisco: Multi-modal discriminator-guided cooperative diffusion for joint audio and video generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=agbiPPuSeQ.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ruifei He, Chuhui Xue, Haoru Tan, Wenqing Zhang, Yingchen Yu, Song Bai, and Xiaojuan Qi. Debiasing text-to-image diffusion models. In *Proceedings of the 1st ACM Multimedia Workshop on Multi-Modal Misinformation Governance in the Era of Foundation Models*, MIS '24, pp. 29–36, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400712012. doi: 10.1145/3689090.3689387. URL https://doi.org/10.1145/3689090.3689387.

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

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9307–9315, 2024. doi: 10.1109/CVPR52733.2024.00889.

Puming Jiang, Meiyi Wang, Kelvin Lin, Ce Hao, and Harold Soh. Guided streaming stochastic interpolant policy. *arXiv preprint arXiv:2605.10051*, 2026.

Yilei Jiang, Wei-Hong Li, Yiyuan Zhang, Minghong Cai, and Xiangyu Yue. Fairgen: Enhancing fairness in text-to-image diffusion models via self-discovering latent directions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18411–18420, October 2025.

Mintong Kang, Vinayshekhar Bannihatti Kumar, Shamik Roy, Abhishek Kumar, Sopan Khosla, Balakrishnan Murali Narayanaswamy, and Rashmi Gangadharaiah. FairGen: Controlling sensitive attributes for fair generations in diffusion models via adaptive latent guidance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25336–25350. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1287.

Kimmo Karkkainen and Jungseock Joo. FairFace: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1548–1558, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=k7FuTOWMOc7.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

Kwanyoung Kim and Jong Chul Ye. Noise2score: Tweedie's approach to self-supervised image denoising without clean images. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 864–874, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/077b83af57538aa183971a2fe0971ec1-Paper.pdf.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D.E. Kirk. *Optimal Control Theory: An Introduction*. Dover Publications, 2004. ISBN 9780486434841. URL https://books.google.com.sg/books?id=fCh2SAtWIdwC.

Erwin Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, 1991. ISBN 0-471-50731-8.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Henry Li and Marcus Aloysius Pereira. Solving inverse problems via diffusion optimal control. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=wqLC4G1GN3.

Jia Li, Lijie Hu, Jingfeng Zhang, Tianhang Zheng, Hua Zhang, and Di Wang. Fair text-to-image diffusion via fair mapping. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39 (25):26256–26264, 2025. doi: 10.1609/aaai.v39i25.34823. URL https://ojs.aaai.org/index.php/AAAI/article/view/34823.

Qianxiao Li, Long Chen, Cheng Tai, and Weinan E. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18(165):1–29, 2018. URL http://jmlr.org/papers/v18/17-653.html.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Yusuke Kato, and Kazuki Kozuka. Aligning diffusion models by optimizing human utility. *Advances in Neural Information Processing Systems*, 37:24897–24925, 2024.

Jinhao Liang, Jacob K Christopher, Sven Koenig, and Ferdinando Fioretto. Simultaneous multi-robot motion planning with projected diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Sp7jclUwkV.

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

Buhua Liu, Shitong Shao, Bao Li, Lichen Bai, Zhiqiang Xu, Haoyi Xiong, James Kwok, Sumi Helal, and Zeke Xie. Alignment of diffusion models: Fundamentals, challenges, and future. *arXiv preprint arXiv:2409.07253*, 2024.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Zhen Liu, Tim Z. Xiao, Carles Domingo-Enrich, Weiyang Liu, and Dinghuai Zhang. Value gradient guidance for flow matching alignment. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6MmOy2Ji8V.

Hao Luan, See-Kiong Ng, and Chun Kai Ling. DDPS: Discrete diffusion posterior sampling for paths in layered graphs. In *ICLR 2025 Frontiers in Probabilistic Inference: Learning Meets Sampling Workshop*, 2025. URL https://openreview.net/forum?id=DBdkU0Ikzy.

Hao Luan, Yi Xian Goh, See-Kiong Ng, and Chun Kai Ling. Projected coupled diffusion for test-time constrained joint generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=1FEm5JLpvg.

Manuel Madeira, Clément Vignac, Dorina Thanou, and Pascal Frossard. Generative modelling of structurally constrained graphs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 137218–137262, 2024.

Marianthi Markatou, Yang Chen, Georgios Afendras, and Bruce G Lindsay. Statistical distances and their role in robustness. In *New advances in statistics and data science*, pp. 3–26. Springer, 2018.

Colin McDiarmid. *On the method of bounded differences*, pp. 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.

Zichen Miao, Jiang Wang, Ze Wang, Zhengyuan Yang, Lijuan Wang, Qiang Qiu, and Zicheng Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10844–10853, 2024.

Mashrur M. Morshed and Vishnu Boddeti. Diverseflow: Sample-efficient diverse mode coverage in flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23303–23312, June 2025.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4474–4484. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/niu20a.html.

Kushagra Pandey, Farrin Marouf Sofian, Felix Draxler, Theofanis Karaletsos, and Stephan Mandt. Variational control for guidance in diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47755–47780. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/pandey25a.html.

Rishubh Parihar, Abhijnya Bhat, Abhipsa Basu, Saswat Mallick, Jogendra Nath Kundu, and R. Venkatesh Babu. Balancing act: Distribution-guided debiasing in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6668–6678, June 2024.

Byoungwoo Park, Jungwon Choi, Sungbin Lim, and Juho Lee. Stochastic optimal control for diffusion bridges in function spaces. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=WyQW4G57Zd.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

pkr7098. cifar100-vit-base-patch16-224-in21k. Hugging Face model repository, 2024. URL https://huggingface.co/pkr7098/cifar100-vit-base-patch16-224-in21k. Retrieved July 2026.

A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in Applied Mathematics. Springer Berlin Heidelberg, 2006. ISBN 9783540346586. URL https://books.google.com.sg/books?id=31m4ahn_KfkC.

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

Litu Rout, Yujia Chen, Nataniel Ruiz, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. RB-modulation: Training-free stylization using reference-based modulation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=bnINPG5A32.

Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10219–10228, 2023.

Walter Rudin. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, New York, 3rd edition, 1976. ISBN 9780070856134. URL https://books.google.com.sg/books?id=kwqzPAAACAAJ.

Xudong Shen, Chao Du, Tianyu Pang, Min Lin, Yongkang Wong, and Mohan Kankanhalli. Finetuning text-to-image diffusion models for fairness. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hnrB5YHoYu.

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.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9_gsMA8MRKQ.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11895–11907, 2019.

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

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vaRCHVj0uGI.

syntheticbot. clip-face-attribute-classifier. Hugging Face model repository, 2025. URL https://huggingface.co/syntheticbot/clip-face-attribute-classifier. Retrieved July 2026.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Luran Wang, Chaoran Cheng, Yizhen Liao, Yanru Qu, and Ge Liu. Training free guided flow-matching with optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=61ss5RA1MM.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22370–22417, 2024.

Stefano Zampini, Jacob K Christopher, Luca Oneto, Davide Anguita, and Ferdinando Fioretto. Training-free constrained generation with stable diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=TrNB08KuHK.

Yu Zeng, Vishal M Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. Jedi: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6786–6795, 2024.

Kaizhen Zhu, Mokai Pan, Yuexin Ma, Yanwei Fu, Jingyi Yu, Jingya Wang, and Ye Shi. UniDB: A unified diffusion bridge framework via stochastic optimal control. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=uqCfoVXb67.

TABLE OF CONTENTS FOR APPENDIX

A Related Work **18**

B Method Details **18**

B.1 PF-ODE Instances 18

B.2 Algorithm 19

B.3 Computational Complexity 19

C Theoretical Derivations **21**

C.1 Close-Form Control Update 21

C.2 Algorithmic Stationarity of the E-MSA Iteration 21

C.2.1 Setup, Discretization, and Projected-Gradient Mapping 21

C.2.2 Assumptions 22

C.2.3 Statement 23

C.2.4 Proof of **Proposition C.7** 23

C.2.5 Remarks and Discussions 30

C.3 Statistical Bounds for Distribution Alignment 31

C.3.1 Setup and Population Objects 31

C.3.2 Assumptions 32

C.3.3 Population Alignment Proof Ingredients 33

C.3.4 Statement 44

C.3.5 Proof of **Proposition C.25** 44

C.3.6 Deterministic Batch Attribute Alignment Bound 46

D Experiment Details **48**

D.1 CIFAR Experiment 48

D.2 Human Face Generation Experiment 49

E Additional Results **51**

E.1 Additional Results for CIFAR 51

E.1.1 Ablations 51

E.1.2 Additional Samples 51

E.2 Additional Results for Face Generation Experiment 60

E.2.1 Ablations 60

E.2.2 Additional Metric 60

E.2.3 Additional Samples 63

F Discussions and Limitations **66**

A Related Work

Diffusion Models with Conditional Generation. Diffusion models define the generative process as iterative denoising (Ho et al., 2020) or, equivalently, as score-based Langevin dynamics (Song & Ermon, 2019; Song et al., 2021b). While DDIM (Song et al., 2021a) and EDM (Karras et al., 2022) improved sampling efficiency and design clarity, controllable generation often relies on guidance mechanisms. Dhariwal & Nichol (2021) introduced classifier guidance to steer pretrained models, while Ho & Salimans (2022) proposed classifier-free guidance for joint training with conditioning signals. Chung et al. (2023) proposed posterior sampling for inverse problems, upon which many more general training-free guidance techniques were built (Ye et al., 2024; Feng et al., 2024; Guo et al., 2024; Corso et al., 2024).

Fairness and Diversity in Diffusion. Fairness generation with diffusion is an instance of the DADA problem. Shen et al. (2024) propose a supervised fine-tuning method using distributional alignment loss and adjusted direct fine-tuning to mitigate demographic biases in T2I diffusion models. Miao et al. (2024) take a reinforcement learning approach for fine-tuning with policy gradient methods and a diversity reward. As for test-time methods, Friedrich et al. (2025) conduct a preliminary model audit to generate a lookup table of biased prompts and attributes, and then employs Semantic Guidance to promote fair attribute classes and suppress biased terms. Jiang et al. (2025) train extra attribute-specific adapters and guide diffusion generation in a plug-and-play fashion; Fair Mapping (Li et al., 2025) adopts a linear network to map input embeddings into a debiased representation space. FairGen (Kang et al., 2025) uses an additionally-trained latent module and an extra memory module to steer generations toward user-specified fair distributions. Distribution-focused techniques such as IDA (He et al., 2024) optimize the weights of multi-directional text descriptions, while Parihar et al. (2024) trains a predictor to map h-space features to attribute distributions during the denoising diffusion process. Notably, Corso et al. (2024) propose a training-free guidance-based method for promoting sample diversity in diffusion models. However, most of the above methods are specifically designed for T2I conditional models and leverage text conditioning mechanisms that are baked inside the T2I models, thus not generalizable to unconditional models nor to other domains.

Optimal Control in Diffusion- and Flow-based Models. Early theoretical connections between Stochastic Optimal Control (SOC) and diffusion were established in (Berner et al., 2024). Domingo-Enrich et al. (2024; 2025) propose Adjoint Matching and Stochastic Optimal Control Matching for fine-tuning diffusion- and flow-based models by learning optimal control fields via regression objectives, as other approaches follow (Han et al., 2025; Liu et al., 2025). Park et al. (2024); Zhu et al. (2025) study diffusion bridges from the SOC perspective. In contrast, training-free methods such as OC-Flow (Wang et al., 2025) and DTM (Pandey et al., 2025) steer pre-trained diffusion or flow models by differentiating through the generative ODE or SDE to minimize control costs without updating model parameters. The OC formulation also extends to solving inverse problems (Li & Pereira, 2024), adaptive guidance strength scheduling (Azangulov et al., 2026), stylization (Rout et al., 2025) and multi-subject generation (Bill et al., 2025) by modulating trajectories or maximizing likelihoods directly during the sampling process. While our method resonates with these works in terms of methodology, none of them are addressing the attribute distribution alignment problem.

B Method Details

B.1 PF-ODE Instances

In this work, we focus on the *PF-ODE of the reverse diffusion process*, which appears in various forms across literature. We identify two instances here. With the formulation of EDM (Karras et al., 2022), the PF-ODE of a reverse diffusion process reads:

$$\dot{\mathbf{x}}_t = -t\,\mathbf{s}^\theta(\mathbf{x}_t, t). \tag{15}$$

The associated PF-ODE of DDIM, which corresponds to that of the "Variance-Exploding" SDE in Song et al. (2021b), is as follows Song et al. (2021a):

$$\mathrm{d}\tilde{\mathbf{x}}_t = \boldsymbol{\epsilon}^\theta\left(\frac{\tilde{\mathbf{x}}_t}{\sqrt{1+\sigma(t)^2}}\right)\mathrm{d}\sigma(t) \quad \text{with} \quad \tilde{\mathbf{x}}_t := \frac{\mathbf{x}_t}{\sqrt{\alpha_t}} \quad \text{and} \quad \sigma(t) := \sqrt{(1-\alpha_t)/\alpha_t} \tag{16}$$

with α_t being a decreasing sequence related to the diffusion noising schedule[9], and ϵ^θ a learned noise prediction model parameterized by θ. Without loss of generality, we denote both of them by

$$\dot{\mathbf{x}}_t = \mathbf{f}^\theta(\mathbf{x}_t, t), \quad \mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n), \quad t \in [0, T]. \tag{17}$$

Flow ODE The ODE under the flow matching paradigm Lipman et al. (2023); Liu et al. (2023) is straightforward:

$$\dot{\mathbf{x}}_t = v^\theta(\mathbf{x}_t, t) \tag{18}$$

where $t \in [0, 1]$ and $v^\theta : \mathbb{R}^n \times [0, 1] \to \mathbb{R}^n$ is a learned vector field model parameterized by θ. The inference generation process is simply integrating the Eq. (18) given a prior distribution $p_1(\mathbf{x}_t)$. Under the latent Flow Matching (LFM) Dao et al. (2023) setting, the state $\mathbf{x}$ lies in the latent space of some pretrained VAE.

Perturbed PF-ODE Instances In this work, we instantiate the perturbed PF-ODE Eq. (4) under the EDM Karras et al. (2022) and DDIM Song et al. (2021a) formulations. For EDM, we take the following *perturbed* ODE:

$$\mathbf{f}^\theta_{\text{EDM}}(\mathbf{x}_t, \mathbf{u}_t, t) := \dot{\mathbf{x}}_t = (T - t)\mathbf{s}^\theta(\mathbf{x}_t, T - t) + \mathbf{u}_t, \tag{19}$$

setting $\mathbf{g}(\mathbf{x}_t, t) = \boldsymbol{I}_n$. For DDIM, we adopt

$$\mathbf{f}^\theta_{\text{DDIM}}(\tilde{\mathbf{x}}_\sigma, \mathbf{u}_\sigma, \sigma) := \frac{\mathrm{d}\tilde{\mathbf{x}}_\sigma}{\mathrm{d}\sigma} = \epsilon^\theta\left(\frac{\tilde{\mathbf{x}}_\sigma}{\sqrt{1 + \sigma^2}}\right) + \mathbf{u}_\sigma, \tag{20}$$

and perform control in the spaces of $\tilde{\mathbf{x}}_\sigma$ and $\sigma(T - t)$.

The perturbed ODE under the FM setting adopted in this work is plainly:

$$\mathbf{f}^\theta_{\text{FM}}(\mathbf{x}_t, \mathbf{u}_t, t) := \dot{\mathbf{x}}_t = v^\theta(\mathbf{x}_t, t) + \mathbf{u}_t. \tag{21}$$

B.2 Algorithm

The algorithmic description of our method is in Algorithm 1.

B.3 Computational Complexity

Time Complexity. Let C_{NFE} be the cost of one *batched* forward evaluation of the pretrained diffusion or flow network $\mathbf{f}^\theta$ on the M-sample batch, *i.e.*, one neural function evaluation (NFE), and C_{VJP} the cost of one batched reverse-mode vector-Jacobian product (VJP) through $\mathbf{f}^\theta$ on the same batch. Let C_Φ further denote the cost of one batched forward-plus-backward through the differentiable attribute model Ψ (composed with the density estimator) on the M-sample batch. Standard autodiff frameworks such as PyTorch gives $C_{\text{VJP}} \approx 2\text{–}3\, C_{\text{NFE}}$. Since C_{NFE}, C_{VJP}, and C_Φ all scale linearly with M in compute, we absorb the batch size M into them. Each outer iteration of Algorithm 1 consists of (i) a forward pass of K batched NFEs at cost $O(K\, C_{\text{NFE}})$; (ii) one forward and backward pass through Ψ to compute the terminal adjoint $\mathbf{N}_K$ at cost C_Φ, independent of K and is invoked only once per outer iteration; (iii) a backward (adjoint) pass of K batched VJPs at cost $O(K\, C_{\text{VJP}})$; and (iv) a closed-form control update Eq. (13) that is elementwise with an *optional* projection onto $\mathcal{U}^M$ (most commonly a clipping operation and negligible to the NFEs, so considered as $O(1)$ cost). A final forward pass of K NFEs generates the resulting samples. Overall, the algorithm runs in

$$O\big(I \cdot [K\,(C_{\text{NFE}} + C_{\text{VJP}}) + C_\Phi]\,\big). \tag{22}$$

As such, our method incurs an $I \cdot \big(1 + C_{\text{VJP}}/C_{\text{NFE}} + C_\Phi/(K\, C_{\text{NFE}})\big)$ multiplicative overhead over the $O(K\, C_{\text{NFE}})$ cost of vanilla unconditional sampling, where the last term is typically the smallest.

[9]There is a notional mismatch in α_t across different work. The α_t in (Song et al., 2021a) corresponds to the $\bar{\alpha}_t$ in (Ho et al., 2020). We refer readers to Ho et al. (2020); Song et al. (2021a) for the details. In the above, we adopt the notations by Song et al. (2021a).

Algorithm 1 DADA via E-MSA with Euler discretization

Require: Dynamics $\mathbf{F}^\theta$; terminal cost Φ; batch init $\boldsymbol{X}_{\text{init}}$; time grid $\{t_k\}_{k=0}^K$ with $t_0 = 0, t_K = T$; $\rho > 0, \xi \geq 0$; max iteration I; optional $\mathcal{U}$.

$\eta \leftarrow (1-\xi)/\rho$
$h_k \leftarrow t_{k+1} - t_k$ for $k = 0, \ldots, K-1$
$\mathbf{U}_k \leftarrow \mathbf{0}$ for $k = 0, \ldots, K-1$
for $j = 1$ to I **do**
  ▷ *(1) Forward pass with control* $\mathbf{U}$ ◁
  $\mathbf{X}_0 \leftarrow \mathbf{X}_{\text{init}}$
  for $k = 0, \ldots, K-1$ **do**
    $\mathbf{X}_{k+1} \leftarrow \mathbf{X}_k + h_k\, \mathbf{F}^\theta(\mathbf{X}_k, \mathbf{U}_k, t_k)$
  ▷ *(2) Cost evaluation and backward pass* ◁
  $\mathbf{N}_K \leftarrow \nabla_{\mathbf{X}} \Phi\left(\hat{p}^u_{\mathbf{y}}(\mathbf{X}_K)\right)$
  for $k = K-1, \ldots, 0$ **do**
    ▷ *Computed via VJP* ◁
    $\mathbf{V}_k \leftarrow \left(\nabla_{\mathbf{X}} \mathbf{F}^\theta(\mathbf{X}_k, \mathbf{U}_k, t_k)\right)^\top \mathbf{N}_{k+1}$
    $\mathbf{N}_k \leftarrow \mathbf{N}_{k+1} + h_k \mathbf{V}_k$
  ▷ *(3) Closed-form control update* ◁
  for $k = 0, \ldots, K-1$ **do**
    $\mathbf{U}_k^{\text{new}} \leftarrow \xi \mathbf{U}_k - \eta\, \mathbf{G}(X_k, t_k)^\top \mathbf{N}_{k+1}$
    $\mathbf{U}_k \leftarrow \Pi_{\mathcal{U}^M}(\mathbf{U}_k^{\text{new}})$ if $\mathcal{U}$ specified else $\mathbf{U}_k^{\text{new}}$
  if Converged **then break**
▷ *Forward simulation with final control* ◁
for $k = 0, \ldots, K-1$ **do**
  $\mathbf{X}_{k+1} \leftarrow \mathbf{X}_k + h_k\, \mathbf{F}^\theta(\mathbf{X}_k, \mathbf{U}_k, t_k)$
return $\mathbf{X}_K$

Space Complexity. Let $|\theta|$ and A_{net} denote the parameter size and the peak forward-plus-VJP activation memory of $\mathbf{f}^\theta$ on the M-sample batch, and let $|\Psi|$ and A_Φ denote the corresponding quantities for the differentiable attribute model Ψ (composed with the density estimator). Note that all four are independent of K. Across one outer iteration of Algorithm 1, the persistent buffers are (i) the state trajectory $\{\mathbf{X}_k\}_{k=0}^K$ at $O(KMn)$; (ii) the control trajectory $\{\mathbf{U}_k\}_{k=0}^{K-1}$ at $O(KMm)$; and (iii) the adjoint/costate trajectory $\{\mathbf{N}_k\}_{k=0}^K$ at $O(KMn)$, but this is reduced to $O(Mn)$ since the update Eq. (13) is fused into the backward loop in our implementation (note that the control update only depends on the state and costate of the current time step). Each per-step VJP is computed locally on one $\mathbf{f}^\theta$ call and discarded before the next, so transient activations are bounded by $O(A_{\text{net}})$ at any step within the backward loop. The terminal-cost gradient $\nabla_{\mathbf{X}}\Phi(\hat{p}^u_{\mathbf{y}}(\mathbf{X}_K))$ contributes one transient pass through Ψ of activation cost $O(A_\Phi)$, released before the adjoint loop begins. All of the aforementioned buffers are reused across outer iterations, so peak memory *does not* grow with I. In sum, the space complexity is

$$O\big(|\theta| + |\Psi| + KM(n+m) + A_{\text{net}} + A_\Phi\big), \tag{23}$$

and independent of number of iterations I.

Remarks on Avoided Costs. Two hypothetical memory costs are avoided by the structure of Algorithm 1. First, the Jacobian Eq. (10) is never explicitly formed; an explicit materialization would have incurred $O(M^2n^2)$ memory. Second, end-to-end backpropagation through the entire unrolled K-step sampler under standard reverse-mode auto-differentiation would have incurred activations of size $O(K\, A_{\text{net}})$, but our adjoint-based formulation instead retains only the state of size $O(KMn)$ and reconstructs the per-step activations on demand, which is a memory–compute trade-off equivalent to applying per-step gradient checkpointing to the unrolled sampler.

C THEORETICAL DERIVATIONS

C.1 CLOSE-FORM CONTROL UPDATE

Proposition C.1 (Proposition 3.1)**.** *For any fixed $t \in [0, T]$ and fixed $(\boldsymbol{x}_t, \boldsymbol{\nu}_t)$, with $\rho + \gamma > 0$, the optimal control $\mathbf{u}_t^*$ specified in Eq. (12) is given by*

$$\mathbf{u}_t^* \in \Pi_{\mathcal{U}}(\bar{\boldsymbol{u}}_t) := \arg\min_{\mathbf{u} \in \mathcal{U}} \|\mathbf{u} - \bar{\boldsymbol{u}}_t\|^2,$$

where

$$\bar{\boldsymbol{u}}_t := \frac{1}{\rho + \gamma} \left(\gamma \, \boldsymbol{u}_t^{\mathrm{ref}} - \mathbf{g}(\boldsymbol{x}_t, t)^\top \boldsymbol{\nu}_t \right),$$

and $\Pi_{\mathcal{U}}(\bar{\boldsymbol{u}}_t)$ is nonempty. Further, $\mathbf{u}_t^*$ is unique if $\mathcal{U}$ is convex.

Proof. For any fixed $t, (\boldsymbol{x}_t, \boldsymbol{\nu}_t)$, extract the part of H that depends on $\mathbf{u}$ as

$$\Lambda_t(\mathbf{u}) := \frac{\rho}{2}\|\mathbf{u}\|^2 + \frac{\gamma}{2}\|\mathbf{u} - \boldsymbol{u}_t^{\mathrm{ref}}\|^2 + \boldsymbol{\nu}_t^\top \mathbf{g}(\boldsymbol{x}_t, t)\mathbf{u}.$$

Since $\rho + \gamma > 0$, completing squares yields

$$\Lambda_t(\mathbf{u}) = \frac{\rho + \gamma}{2}\|\mathbf{u} - \bar{\boldsymbol{u}}_t\|^2 + c_t,$$

with

$$\bar{\boldsymbol{u}}_t = \frac{1}{\rho + \gamma} \left(\gamma \, \boldsymbol{u}_t^{\mathrm{ref}} - \mathbf{g}(\boldsymbol{x}_t, t)^\top \boldsymbol{\nu}_t \right)$$

and

$$c_t = \frac{\gamma}{2}\|\boldsymbol{u}_t^{\mathrm{ref}}\|^2 - \frac{\rho + \gamma}{2}\|\bar{\boldsymbol{u}}_t\|^2.$$

Therefore,

$$H(\boldsymbol{x}_t, \boldsymbol{\nu}_t, \mathbf{u}, t) = \frac{\rho + \gamma}{2}\|\mathbf{u} - \bar{\boldsymbol{u}}_t\|^2 + \boldsymbol{\nu}_t^\top \mathbf{f}^\theta(\boldsymbol{x}_t, t) + c_t.$$

The last two terms do not depend on $\mathbf{u}$, hence

$$\arg\min_{\mathbf{u} \in \mathcal{U}} \widetilde{H}(\boldsymbol{x}_t, \boldsymbol{\nu}_t, \mathbf{u}, t) = \arg\min_{\mathbf{u} \in \mathcal{U}} \|\mathbf{u} - \bar{\boldsymbol{u}}_t\|^2 = \Pi_{\mathcal{U}}(\bar{\boldsymbol{u}}_t).$$

Since $\mathcal{U}$ is a nonempty and closed set and $\mathbf{u} \mapsto \|\mathbf{u} - \bar{\boldsymbol{u}}_t\|^2$ is continuous and coercive, the minimum is attained per (Beck, 2014, Theorem 2.32). Hence, $\Pi_{\mathcal{U}}(\bar{\boldsymbol{u}}_t)$ is nonempty. Further, if set $\mathcal{U}$ is convex, since $\mathbf{u} \mapsto \|\mathbf{u} - \bar{\boldsymbol{u}}_t\|^2$ is *strictly convex* in $\mathbf{u}$, the set of minimizer, *i.e.*, $\Pi_{\mathcal{U}}(\bar{\boldsymbol{u}}_t)$, contains *at most one point* (Boyd & Vandenberghe, 2004, Sec. 4.2.1); its uniqueness is therefore obtained when combined with non-emptiness. □

C.2 ALGORITHMIC STATIONARITY OF THE E-MSA ITERATION

This subsection gives the full statement, the assumptions, and the proof of **Proposition 3.2** in the main text. The proposition identifies the closed-form update Eq. (13) as projected gradient descent (PGD) on the Euler-discretized objective of Eq. (7) in a time-weighted inner product space, and indicates the convergence of the norm of the gradient mapping and that all cluster points of the control sequence generated by Algorithm 1 are stationary points to the discretized problem.

C.2.1 SETUP, DISCRETIZATION, AND PROJECTED-GRADIENT MAPPING

Time grid and time-weighted inner product. Fix $T > 0$ and a time grid $0 = t_0 < t_1 < \cdots < t_K = T$ with step sizes $h_k := t_{k+1} - t_k > 0$. For control sequences $\mathbf{U} = (\mathbf{U}_0, \ldots, \mathbf{U}_{K-1}) \in (\mathbb{R}^{Mm})^K$ define the h-weighted inner product with norm

$$\langle \mathbf{U}, \mathbf{V} \rangle_h := \sum_{k=0}^{K-1} h_k \, \langle \mathbf{U}_k, \mathbf{V}_k \rangle, \qquad \|\mathbf{U}\|_h := \sqrt{\langle \mathbf{U}, \mathbf{U} \rangle_h}. \tag{24}$$

Euler-discretized objective. Let $\mathcal{C} := (\mathcal{U}^M)^K \subseteq \mathbb{R}^{Mm\times K}$ and write $\Phi^\sharp(\mathbf{X}) := \Phi\left(\hat{p}^u_{\mathbf{y}}(\mathbf{X})\right)$ for the composite state-to-cost map. The Euler-discretized version of the objective in Eq. (7) is

$$J_h(\mathbf{U}) \;:=\; \Phi^\sharp(\mathbf{X}_K) + \frac{\rho}{2}\sum_{k=0}^{K-1} h_k\,\|\mathbf{U}_k\|^2, \tag{25}$$

where $\mathbf{X}_K = \mathbf{X}_K(\mathbf{U})$ is the terminal state of the forward Euler recursion

$$\mathbf{X}_{k+1} = \mathbf{X}_k + h_k\,\mathbf{F}^\theta(\mathbf{X}_k, \mathbf{U}_k, t_k), \quad k = 0,\dots,K-1, \quad \mathbf{X}_0 = \mathbf{X}_{\text{init}}. \tag{26}$$

The discrete costate is generated by the backward recursion

$$\mathbf{N}_K := \nabla_{\mathbf{X}}\Phi^\sharp(\mathbf{X}_K), \qquad \mathbf{N}_k := \mathbf{N}_{k+1} + h_k\,\left(\nabla_{\mathbf{X}}\mathbf{F}^\theta(\mathbf{X}_k, \mathbf{U}_k, t_k)\right)^\top \mathbf{N}_{k+1}. \tag{27}$$

Riesz representation of the gradient. A standard chain-rule computation on Eq. (26) together with Eq. (27) gives the Euclidean partial gradient $\nabla_{\mathbf{U}_k} J_h(\mathbf{U}) = h_k\left(\rho\,\mathbf{U}_k + \mathbf{G}(\mathbf{X}_k, t_k)^\top \mathbf{N}_{k+1}\right)$. By (Kreyszig, 1991, Theorem 3.8-4), the unique element $\nabla^h J_h(\mathbf{U})$ satisfying $\langle \nabla^h J_h(\mathbf{U}), \mathbf{V}\rangle_h = \mathrm{d}J_h(\mathbf{U})[\mathbf{V}]$ for all $\mathbf{V}$ is therefore

$$\left[\nabla^h J_h(\mathbf{U})\right]_k \;=\; \rho\,\mathbf{U}_k + \mathbf{G}(\mathbf{X}_k, t_k)^\top \mathbf{N}_{k+1}. \tag{28}$$

Since $\mathcal{C}$ is a Cartesian product and the h-inner product separates across blocks, the h-projection onto $\mathcal{C}$ coincides with per-block Euclidean projection onto $\mathcal{U}^M$: $[\Pi^h_{\mathcal{C}}(\mathbf{V})]_k = \Pi_{\mathcal{U}^M}(\mathbf{V}_k)$. The projected-gradient mapping with step $\eta > 0$ is

$$\mathbf{G}_\eta(\mathbf{U}) := \eta^{-1}\left(\mathbf{U} - \Pi^h_{\mathcal{C}}\left(\mathbf{U} - \eta\,\nabla^h J_h(\mathbf{U})\right)\right). \tag{29}$$

C.2.2 Assumptions

We use lowercase $b_\bullet, \ell_\bullet$ for per-sample constants and uppercase $B_\bullet, L_\bullet$ for batched constants. Block-diagonality of $\mathbf{F}^\theta$ and $\mathbf{G}$ preserves operator-norm Jacobian bounds under stacking.

Assumption C.2 (Smooth dynamics)**.** $\mathbf{f}^\theta(\cdot, t)$ and $\mathbf{g}(\cdot, t)$ are C^1 in $\mathbf{x}$ for every $t \in [0, T]$, with $\nabla_{\mathbf{x}}\mathbf{f}^\theta$ and $D_{\mathbf{x}}\mathbf{g}$ bounded and Lipschitz (in operator norm, uniformly in t) on every bounded subset of $\mathbb{R}^n$.

Assumption C.3 (Smooth terminal cost on the reachable set)**.** On the reachable set $\mathcal{X}_R$ specified in **Assumption C.5**, $\Phi^\sharp(\mathbf{X}) = \Phi(\hat{p}^u_{\mathbf{y}}(\mathbf{X}))$ is C^1, with bounded gradient $\|\nabla_{\mathbf{X}}\Phi^\sharp(\mathbf{X})\| \le B_{\nabla\Phi^\sharp}$ and $L_{\Phi^\sharp}$-Lipschitz gradient.

Assumption C.4 (Compact admissible control set)**.** $\mathcal{U} \subseteq \mathbb{R}^m$ is nonempty, closed, convex, with per-sample bound $\|\mathbf{u}\| \le b_u$ for all $\mathbf{u} \in \mathcal{U}$, and $\mathbf{0} \in \mathcal{U}$. Hence $\mathcal{C} = (\mathcal{U}^M)^K$ is closed, convex, and compact.

Assumption C.5 (Bounded reachable set)**.** There exists a compact set $\mathcal{X}_R \subseteq \mathbb{R}^{Mn}$ such that every Euler trajectory $\{\mathbf{X}_k(\mathbf{U})\}_{k=0}^K$ generated by Eq. (26) with $\mathbf{U} \in \mathcal{C}$ satisfies $\mathbf{X}_k(\mathbf{U}) \in \mathcal{X}_R$ for all k. For analyses throughout this subsection, we assume the initial batch $\mathbf{X}_{\text{init}}$ is fixed; the result is therefore deterministic conditional on a fixed realized batch.

Remark 4. **Assumption C.5** is an a priori condition that prevents circularity in the per-sample bounds defined below. For typical neural-network drifts $\mathbf{f}^\theta$ used in practice, the controlled trajectories are empirically bounded throughout.

Under **Assumption C.2**, **Assumption C.4**, and **Assumption C.5**, the following finite per-sample constants exist for the single-sample drift $\mathbf{f}^\theta(\mathbf{x}, t) + \mathbf{g}(\mathbf{x}, t)\mathbf{u}$ on the projection of $\mathcal{X}_R$ to a single sample and $\|\mathbf{u}\| \le b_u$: $b_g := \sup\|\mathbf{g}(\mathbf{x}, t)\|_{\text{op}}$, $b_{\nabla f} := \sup\|\nabla_{\mathbf{x}}\mathbf{f}^\theta(\mathbf{x}, t)\|_{\text{op}}$, $b_{\nabla g} := \sup\|D_{\mathbf{x}}\mathbf{g}(\mathbf{x}, t)\|$, ℓ_f, ℓ_g for the Lipschitz constants of $\mathbf{f}^\theta, \mathbf{g}$ in $\mathbf{x}$, and $\ell_{\nabla f}, \ell_{\nabla g}$ for the Lipschitz constants of the respective Jacobians. From these, the controlled-drift Jacobian satisfies

$$b_j := b_{\nabla f} + b_{\nabla g}\,b_u, \qquad \ell_{j,x} := \ell_{\nabla f} + \ell_{\nabla g}\,b_u, \qquad \ell_{j,u} := b_{\nabla g}. \tag{30}$$

Block-diagonality preserves these operator-norm and Jacobian-Lipschitz bounds unchanged at the batched level; only vector-norm bounds scale with $\sqrt{M}$.

Assumption C.6 (Step-size hypothesis)**.** Algorithm parameters $\xi \ge 0$ and $\rho > 0$ determine $\eta := (1-\xi)/\rho$ and the proximal weight $\gamma := \rho\,\xi/(1-\xi)$ (equivalently $\xi = \gamma/(\rho+\gamma)$, $\eta = 1/(\rho+\gamma)$). Let L be the gradient-Lipschitz constant of $\nabla^h J_h$ on $\mathcal{C}$, whose existence is established in Lemma C.10 below. We require

$$\xi \in [0, 1) \quad \text{with} \quad \eta = \frac{1-\xi}{\rho} \;\le\; \frac{1}{L} \quad\Longleftrightarrow\quad \xi \;\ge\; \max\{0,\, 1 - \rho/L\}. \tag{31}$$

C.2.3 STATEMENT

Proposition C.7 (Algorithmic stationarity; restatement of **Proposition 3.2**). *Under* **Assumption C.2**–**Assumption C.6**, *the iterates* $\{\mathbf{U}^{(r)}\}_{r\geq 0}$ *produced by Algorithm 1 (with projection onto the admissible control set) satisfy*

$$\min_{0\leq r<I} \left\|\mathbf{G}_\eta(\mathbf{U}^{(r)})\right\|_h^2 \;\leq\; \frac{2\left(J_h(\mathbf{U}^{(0)}) - J_h^{\inf}\right)}{\eta\, I}, \qquad \forall\, I \geq 1, \tag{32}$$

where $J_h^{\inf} := \inf_{\mathbf{U}\in\mathcal{C}} J_h(\mathbf{U})$ *is finite and attained. Moreover,* $\sum_{r=0}^{\infty} \|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h^2 < \infty$, $\|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h \to 0$ *along the full sequence, and every cluster point* $\mathbf{U}^*$ *of* $\{\mathbf{U}^{(r)}\}$ *in* $\mathcal{C}$ *satisfies* $\mathbf{G}_\eta(\mathbf{U}^*) = 0$.

C.2.4 PROOF OF **Proposition C.7**

The proof of **Proposition C.7** roughly follows the structure of (Beck, 2017, Chapter 10.3). It proceeds in two steps, supported by three auxiliary lemmas: forward Lipschitz sensitivity of the Euler trajectories (Lemma C.8), backward Lipschitz sensitivity of the discrete costate (Lemma C.9), and a gradient-Lipschitz constant L for $\nabla^h J_h$ (Lemma C.10). The two steps then read: (i) the closed-form update is PGD on J_h in the h-inner product; (ii) the standard descent lemma plus telescoping gives the $O(1/I)$ rate. Throughout, $\mathbf{U}, \mathbf{U}' \in \mathcal{C}$ with corresponding states $\mathbf{X}_k, \mathbf{X}'_k$ and costates $\mathbf{N}_k, \mathbf{N}'_k$ sharing the same initial state $\mathbf{X}_0 = \mathbf{X}'_0 = \mathbf{X}_{\text{init}}$. Table 4 collects the Lipschitz constants and bounds appearing throughout the proof.

Table 4: Lipschitz and bounding constants used in the proof of Proposition C.7.

Symbol	Definition	Role	First appearance
Primitives (by the assumptions)			
b_g	$\sup_{(\mathbf{x},t)} \|\mathbf{g}(\mathbf{x},t)\|_{\text{op}}$	diffusion operator norm	Assumption C.2
$b_{\nabla f}$	$\sup \|\nabla_{\mathbf{x}}\mathbf{f}^\theta(\mathbf{x},t)\|_{\text{op}}$	drift Jacobian norm	Assumption C.2
$b_{\nabla g}$	$\sup \|D_{\mathbf{x}}\mathbf{g}(\mathbf{x},t)\|$	diffusion Jacobian norm	Assumption C.2
ℓ_f	Lipschitz of $\mathbf{f}^\theta(\cdot,t)$ in $\mathbf{x}$	drift Lipschitz	Assumption C.2
ℓ_g	Lipschitz of $\mathbf{g}(\cdot,t)$ in $\mathbf{x}$	diffusion Lipschitz	Assumption C.2
$\ell_{\nabla f}$	Lipschitz of $\nabla_{\mathbf{x}}\mathbf{f}^\theta$ in $\mathbf{x}$	drift-Jacobian Lipschitz	Assumption C.2
$\ell_{\nabla g}$	Lipschitz of $D_{\mathbf{x}}\mathbf{g}$ in $\mathbf{x}$	diffusion-Jacobian Lipschitz	Assumption C.2
$B_{\nabla\Phi^\sharp}$	$\sup_{\mathbf{X}\in\mathcal{X}_R} \|\nabla_{\mathbf{X}}\Phi^\sharp(\mathbf{X})\|$	terminal-gradient bound	Assumption C.3
$L_{\Phi^\sharp}$	Lipschitz of $\nabla_{\mathbf{X}}\Phi^\sharp$ on $\mathcal{X}_R$	terminal-gradient Lipschitz	Assumption C.3
b_u	$\sup_{\mathbf{u}\in\mathcal{U}} \|\mathbf{u}\|$	control-set diameter	Assumption C.4
Derived constants			
b_j	$b_{\nabla f} + b_{\nabla g}\, b_u$	controlled-drift Jacobian bound	Eq. (30)
$\ell_{j,x}$	$\ell_{\nabla f} + \ell_{\nabla g}\, b_u$	Jacobian Lipschitz in $\mathbf{x}$	Eq. (30)
$\ell_{j,u}$	$b_{\nabla g}$	Jacobian Lipschitz in $\mathbf{u}$	Eq. (30)
L_{dyn}	$\ell_f + \ell_g\, b_u$	total controlled-dynamics Lipschitz	Lemma C.8
C_X	$b_g\sqrt{T}\, e^{L_{\text{dyn}}T}$	forward-state Lipschitz w.r.t. $\mathbf{U}$	Lemma C.8
B_N	$e^{b_j T}\, B_{\nabla\Phi^\sharp}$	costate uniform bound	Lemma C.9
C_N	$e^{b_j T}\left[L_{\Phi^\sharp}\, C_X + B_N(\ell_{j,x}\, T\, C_X + \ell_{j,u}\sqrt{T})\right]$	costate Lipschitz w.r.t. $\mathbf{U}$	Lemma C.9
C_g	$b_g\, C_N + \ell_g\, B_N\, C_X$	$\nabla^h J_h$ pointwise Lipschitz factor	Eq. (51)
L	$\sqrt{2(\rho^2 + T\, C_g^2)}$	gradient Lipschitz constant of $\nabla^h J_h$	Eq. (50)

Lemma C.8 (Forward sensitivity)**.** *Under* **Assumption C.2**, **Assumption C.4**, **Assumption C.5** *there exists*

$$C_X = b_g \sqrt{T}\, e^{L_{\mathrm{dyn}} T} \tag{33}$$

where $L_{\mathrm{dyn}} := \ell_f + \ell_g\, b_u$*, such that*

$$\|\mathbf{X}_k(\mathbf{U}) - \mathbf{X}_k(\mathbf{U}')\| \;\le\; C_X\, \|\mathbf{U} - \mathbf{U}'\|_h, \qquad \forall\, k \in \{0, \dots, K\},\ \forall\, \mathbf{U}, \mathbf{U}' \in \mathcal{C}. \tag{34}$$

Proof. Let $\boldsymbol{\delta}_k := \mathbf{X}_k - \mathbf{X}'_k$. Subtracting Eq. (26) with $\mathbf{U}$ versus $\mathbf{U}'$, we may decompose it by control-affiness of the dynamical system as follows:

$$\begin{aligned}\mathbf{F}^\theta(\mathbf{X}, \mathbf{U}, t) - \mathbf{F}^\theta(\mathbf{X}', \mathbf{U}', t) = \big[\mathbf{F}^\theta(\mathbf{X}, \mathbf{0}, t) - \mathbf{F}^\theta(\mathbf{X}', \mathbf{0}, t)\big] \\ + [\mathbf{G}(\mathbf{X}, t) - \mathbf{G}(\mathbf{X}', t)]\, \mathbf{U} \\ + \mathbf{G}(\mathbf{X}', t)(\mathbf{U} - \mathbf{U}').\end{aligned} \tag{35}$$

Using block-diagonality and $\max_i \|\mathbf{u}^{[i]}\| \le b_u$ across samples, the three terms of Eq. (35) satisfy

$$\|\mathbf{F}^\theta(\mathbf{X}, \mathbf{0}, t) - \mathbf{F}^\theta(\mathbf{X}', \mathbf{0}, t)\| \le \ell_f \|\mathbf{X} - \mathbf{X}'\|, \tag{36}$$

$$\|\mathbf{G}(\mathbf{X}', t)(\mathbf{U} - \mathbf{U}')\| \le b_g \|\mathbf{U} - \mathbf{U}'\|, \tag{37}$$

$$\begin{aligned}\big\|[\mathbf{G}(\mathbf{X}, t) - \mathbf{G}(\mathbf{X}', t)]\mathbf{U}\big\|^2 &= \sum_{i=1}^{M} \|[\mathbf{g}(\mathbf{x}^{[i]}, t) - \mathbf{g}(\mathbf{x}'^{[i]}, t)]\mathbf{u}^{[i]}\|^2 && \text{(Block-diagonality of } \mathbf{G}) \\ &\le \sum_i b_u^2 \|\mathbf{g}(\mathbf{x}^{[i]}, t) - \mathbf{g}(\mathbf{x}'^{[i]}, t)\|^2 && \textbf{(Assumption C.4)} \\ &\le \ell_g^2 b_u^2 \|\mathbf{X} - \mathbf{X}'\|^2, && (38)\end{aligned}$$

where Eq. (36), Eq. (37) and Eq. (37) are all by **Assumption C.2**. Taking Eq. (36)–Eq. (37) into Eq. (35) yields the one-step bound

$$\|\boldsymbol{\delta}_{k+1}\| \le (1 + h_k L_{\mathrm{dyn}}) \|\boldsymbol{\delta}_k\| + h_k b_g \|\mathbf{U}_k - \mathbf{U}'_k\|. \tag{39}$$

Iterating Eq. (39) from $\boldsymbol{\delta}_0 = \mathbf{0}$ gives

$$\begin{aligned}\|\boldsymbol{\delta}_k\| &\le \sum_{j=0}^{k-1} \left(\prod_{m=j+1}^{k-1} (1 + h_m L_{\mathrm{dyn}}) \right) b_g h_j \|\mathbf{U}_j - \mathbf{U}'_j\| && \text{(Recursion)} \\ &\le \sum_{j=0}^{k-1} \left(\prod_{m=j+1}^{k-1} e^{h_m L_{\mathrm{dyn}}} \right) b_g h_j \|\mathbf{U}_j - \mathbf{U}'_j\| && (1 + x \le e^x) \\ &= b_g \sum_{j=0}^{k-1} \left(e^{L_{\mathrm{dyn}} \sum_m h_m} \right) h_j \|\mathbf{U}_j - \mathbf{U}'_j\| \\ &\le b_g\, e^{L_{\mathrm{dyn}} T} \sum_{j=0}^{k-1} h_j \|\mathbf{U}_j - \mathbf{U}'_j\| && (\textstyle\sum_k h_k = T) \\ &\le b_g\, e^{L_{\mathrm{dyn}} T} \sqrt{\sum_{j=0}^{k-1} \left(\sqrt{h_j} \right)^2 \cdot \sum_{j=0}^{k-1} \left(\sqrt{h_j}\, \|\mathbf{U}_j - \mathbf{U}'_j\| \right)^2} && \text{(Cauchy–Schwarz)} \\ &\le b_g\, e^{L_{\mathrm{dyn}} T} \sqrt{T} \sqrt{\sum_{j=0}^{k-1} h_j \|\mathbf{U}_j - \mathbf{U}'_j\|^2} && (\textstyle\sum_k h_k = T) \\ &= b_g \sqrt{T}\, e^{L_{\mathrm{dyn}} T}\ \|\mathbf{U} - \mathbf{U}'\|_h\end{aligned}$$

where the last step comes from the definition of the h-norm in Eq. (24). □

Lemma C.9 (Backward sensitivity)**.** *Under* **Assumption C.2**–**Assumption C.5***, there exist constants*

$$B_N = e^{b_j T}\, B_{\nabla\Phi^\sharp}$$

and

$$C_N = e^{b_j T}\left[L_{\Phi^\sharp}\, C_X + B_N\left(\ell_{j,x}\, T\, C_X + \ell_{j,u}\,\sqrt{T}\right)\right],$$

such that

$$\|\mathbf{N}_k(\mathbf{U})\| \le B_N, \tag{40}$$

$$\|\mathbf{N}_k(\mathbf{U}) - \mathbf{N}_k(\mathbf{U}')\| \le C_N\, \|\mathbf{U} - \mathbf{U}'\|_h. \tag{41}$$

Proof. By **Assumption C.5**, $\mathbf{X}_k(\mathbf{U}) \in \mathcal{X}_R$ for all $k, \mathbf{U}$. By **Assumption C.3**, we have

$$\|\mathbf{N}_K\| \le B_{\nabla\Phi^\sharp}, \qquad \|\mathbf{N}_K - \mathbf{N}'_K\| \le L_{\Phi^\sharp}\|\mathbf{X}_K - \mathbf{X}'_K\| \tag{42}$$

Combining Eq. (42) and Lemma C.8 yields

$$\|\mathbf{N}_K - \mathbf{N}'_K\| \le L_{\Phi^\sharp}\, C_X \|\mathbf{U} - \mathbf{U}'\|_h. \tag{43}$$

Block-diagonality of the dynamics Jacobian gives the following uniform operator-norm bound

$$\|\nabla_{\mathbf{X}}\mathbf{F}^\theta(\mathbf{X}, \mathbf{U}, t)\|_{\mathrm{op}} = \max_i \left\|\nabla_{\mathbf{x}}[\mathbf{f}^\theta + \mathbf{g}\,\mathbf{u}^{[i]}]\right\|_{\mathrm{op}} \le b_j. \tag{44}$$

Hence, the Euler discretized costate/adjoint dynamics Eq. (27) combined with Eq. (44) gives the following one-step bound

$$\|\mathbf{N}_k\| \le (1 + h_k b_j)\|\mathbf{N}_{k+1}\|. \tag{45}$$

Akin to the proof of Lemma C.8, Eq. (45) iterates from K through any $0 \le k < K$ to

$$\|\mathbf{N}_k\| \le e^{b_j T}\, B_{\nabla\Phi^\sharp} \;=\; B_N, \tag{46}$$

by applying $1 + x \le e^x$, and thus establishing Eq. (40).

For sensitivity, let $\mathbf{\Delta}_k := \mathbf{N}_k - \mathbf{N}'_k$ and $\mathbf{J}_k := \nabla_{\mathbf{X}}\mathbf{F}^\theta(\mathbf{X}_k, \mathbf{U}_k, t_k)$. From Eq. (27), we have

$$\mathbf{\Delta}_k = (\mathbf{I} + h_k \mathbf{J}_k^\top)\mathbf{\Delta}_{k+1} + h_k(\mathbf{J}_k - \mathbf{J}'_k)^\top \mathbf{N}'_{k+1}. \tag{47}$$

Applying Eq. (30) blockwise across samples,

$$\|\mathbf{J}_k - \mathbf{J}'_k\|_{\mathrm{op}} \le \ell_{j,x}\|\mathbf{X}_k - \mathbf{X}'_k\| + \ell_{j,u}\|\mathbf{U}_k - \mathbf{U}'_k\|. \tag{48}$$

Combining Eq. (47)–Eq. (48) with

$$\|\mathbf{I} + h_k \mathbf{J}_k^\top\|_{\mathrm{op}} \le 1 + h_k b_j$$

and Eq. (46) yields

$$\begin{aligned}\|\mathbf{\Delta}_k\| &\le (1 + h_k b_j)\|\mathbf{\Delta}_{k+1}\| \\ &\quad + h_k B_N\left(\ell_{j,x}\|\mathbf{X}_k - \mathbf{X}'_k\| + \ell_{j,u}\|\mathbf{U}_k - \mathbf{U}'_k\|\right).\end{aligned} \tag{49}$$

Iterating Eq. (49) backward from K to k and using the same technique as in the proof of Lemma C.8 ($\prod_j (1 + h_j b_j) \le e^{b_j T}$ and Cauchy–Schwarz) together with C_X established in Lemma C.8, we can obtain

$$\begin{aligned}\|\mathbf{N}_k(\mathbf{U}) - \mathbf{N}_k(\mathbf{U}')\| &\le e^{b_j T}\left[L_{\Phi^\sharp}\, C_X + B_N\left(\ell_{j,x}\, T\, C_X + \ell_{j,u}\,\sqrt{T}\right)\right] \cdot \|\mathbf{U} - \mathbf{U}'\|_h \\ &= C_N\, \|\mathbf{U} - \mathbf{U}'\|_h.\end{aligned}$$

□

Lemma C.10 (Gradient Lipschitz constant)**.** *Under* **Assumption C.2–Assumption C.5**, $\nabla^h J_h$ *is Lipschitz on* $\mathcal{C}$ *in the* h*-norm with constant*

$$L \;=\; \sqrt{2\,(\rho^2 + T\,C_g^2)} \tag{50}$$

where

$$C_g \;:=\; b_g\,C_N + \ell_g\,B_N\,C_X \tag{51}$$

Proof. By Eq. (28) and the triangle inequality,

$$\begin{aligned}\left\|[\nabla^h J_h(\mathbf{U}) - \nabla^h J_h(\mathbf{U}')]_k\right\| \le \rho\|\mathbf{U}_k - \mathbf{U}'_k\| \\ + \left\|\mathbf{G}(\mathbf{X}_k, t_k)^\top \mathbf{N}_{k+1} - \mathbf{G}(\mathbf{X}'_k, t_k)^\top \mathbf{N}'_{k+1}\right\|.\end{aligned} \tag{52}$$

The second term of Eq. (52) deomposes as

$$\begin{aligned}\mathbf{G}(\mathbf{X}_k, t_k)^\top \mathbf{N}_{k+1} - \mathbf{G}(\mathbf{X}'_k, t_k)^\top \mathbf{N}'_{k+1} =& \mathbf{G}(\mathbf{X}_k, t_k)^\top \boldsymbol{\Delta}_{k+1} && (53)\\ &+ (\mathbf{G}(\mathbf{X}_k, t_k) - \mathbf{G}(\mathbf{X}'_k, t_k))^\top \mathbf{N}'_{k+1} && (54)\end{aligned}$$

and for Eq. (53) we have

$$\|\mathbf{G}^\top \boldsymbol{\Delta}_{k+1}\| \le b_g \|\boldsymbol{\Delta}_{k+1}\| \tag{55}$$

per Assumption C.2. For Eq. (54), we have

$$\begin{aligned}\left\|(\mathbf{G}(\mathbf{X}_k, t_k) - \mathbf{G}(\mathbf{X}'_k, t_k))^\top \mathbf{N}'_{k+1}\right\|^2 &= \sum_{i=1}^{M} \left\| \left[\mathbf{g}(\mathbf{x}_k^{[i]}, t_k) - \mathbf{g}(\mathbf{x}_k'^{[i]}, t_k)\right] \boldsymbol{\nu}_{k+1}'^{[i]} \right\|^2 && \text{(Block-diagonal)}\\ &\le \sum_{i=1}^{M} \left\|\mathbf{g}(\mathbf{x}^{[i]}, t) - \mathbf{g}(\mathbf{x}'^{[i]}, t)\right\|_{\mathrm{op}}^2 \left\|\boldsymbol{\nu}'^{[i]}\right\|^2 && \text{(Operator norm)}\\ &\le \sum_{i=1}^{M} \ell_g^2 \left\|\mathbf{x}^{[i]} - \mathbf{x}'^{[i]}\right\|^2 \left\|\boldsymbol{\nu}'^{[i]}\right\|^2 && \text{(Lipschitz } \mathbf{g}\text{)}\\ &\le \ell_g^2 \left(\max_j \left\|\boldsymbol{\nu}'^{[j]}\right\|^2\right) \sum_{i=1}^{M} \left\|\mathbf{x}^{[i]} - \mathbf{x}'^{[i]}\right\|^2 \\ &= \ell_g^2 \left(\max_j \left\|\boldsymbol{\nu}'^{[j]}\right\|^2\right) \|\mathbf{X}_k - \mathbf{X}'_k\|^2 && (L_2 \text{ norm})\\ &\le \ell_g^2 \left\|\mathbf{N}'_{k+1}\right\|^2 \|\mathbf{X}_k - \mathbf{X}'_k\|^2 && (\max_j v_j \le \|\mathbf{v}\|)\\ &\le \ell_g^2 B_N^2 \|\mathbf{X}_k - \mathbf{X}'_k\|^2\end{aligned}$$

Thus,

$$\left\|(\mathbf{G}(\mathbf{X}, t) - \mathbf{G}(\mathbf{X}', t))^\top \mathbf{N}'\right\| \le \ell_g\,B_N\,\|\mathbf{X} - \mathbf{X}'\|. \tag{56}$$

Plugging Eq. (55), and Eq. (56) into Eq. (52) yields

$$\left\|[\nabla^h J_h(\mathbf{U}) - \nabla^h J_h(\mathbf{U}')]_k\right\| \le \rho\|\mathbf{U}_k - \mathbf{U}'_k\| + b_g\|\boldsymbol{\Delta}_{k+1}\| + \ell_g\,B_N\,\|\mathbf{X} - \mathbf{X}'\|.$$

Further invoking Lemma C.8 and Lemma C.9 gives

$$\left\|[\nabla^h J_h(\mathbf{U}) - \nabla^h J_h(\mathbf{U}')]_k\right\| \;\le\; \rho\|\mathbf{U}_k - \mathbf{U}'_k\| + C_g\|\mathbf{U} - \mathbf{U}'\|_h, \tag{57}$$

where $C_g = b_g C_N + \ell_g\,B_N\,C_X$.

Hence,

$$
\begin{aligned}
\|\nabla^h J_h(\mathbf{U}) - \nabla^h J_h(\mathbf{U}')\|_h &= \sqrt{\sum_k h_k \left\|[\nabla^h J_h(\mathbf{U}) - \nabla^h J_h(\mathbf{U}')]_k\right\|^2} \\
&\le \sqrt{\sum_k h_k \left(\rho\|\mathbf{U}_k - \mathbf{U}'_k\| + C_g\|\mathbf{U} - \mathbf{U}'\|_h\right)^2} && \text{(By Eq. (57))} \\
&\le \sqrt{\sum_k 2h_k \left(\rho^2\|\mathbf{U}_k - \mathbf{U}'_k\|^2 + C_g^2\|\mathbf{U} - \mathbf{U}'\|_h^2\right)} && (58) \\
&= \sqrt{2\rho^2 \sum_k h_k\|\mathbf{U}_k - \mathbf{U}'_k\|^2 + 2C_g^2\|\mathbf{U} - \mathbf{U}'\|_h^2 \sum_k h_k} \\
&= \sqrt{2\rho^2\|\mathbf{U} - \mathbf{U}'\|_h^2 + 2C_g^2\|\mathbf{U} - \mathbf{U}'\|_h^2 \sum_k h_k} && (h\text{-norm definition}) \\
&= \sqrt{2\rho^2\|\mathbf{U} - \mathbf{U}'\|_h^2 + 2TC_g^2\|\mathbf{U} - \mathbf{U}'\|_h^2} && (\textstyle\sum_k h_k = T) \\
&= \sqrt{2(\rho^2 + TC_g^2)}\ \|\mathbf{U} - \mathbf{U}'\|_h \\
&= L\|\mathbf{U} - \mathbf{U}'\|_h
\end{aligned}
$$

where Eq. (58) is obtained by $(a+b)^2 \le 2(a^2+b^2)$. □

Remark 5. Note that C_X, B_N, C_N depends only on T and the per-step constants from the assumptions, not on K or the smallest h_k, so the gradient Lipschitz constant L is *independent* of the Euler discretization schedule.

With the lemmas above, we now prove Proposition C.7 (Proposition 3.2) in full.

Proof (Proposition C.7, i.e., Proposition 3.2).
Part 1: The closed-form update is PGD on discreized cost J_h
The update line in Eq. (13), restated at iteration r, reads

$$\mathbf{U}_k^{(r+1)} = \Pi_{\mathcal{U}^M}\left(\xi\,\mathbf{U}_k^{(r)} - \eta\,\mathbf{G}(\mathbf{X}_k^{(r)}, t_k)^\top \mathbf{N}_{k+1}^{(r)}\right).$$

Using $1-\eta\rho = \gamma/(\rho+\gamma) = \xi$, the argument of the projection at time step k is then

$$\begin{aligned}\xi\,\mathbf{U}_k^{(r)} - \eta\,\mathbf{G}(\mathbf{X}_k^{(r)}, t_k)^\top \mathbf{N}_{k+1}^{(r)} &= \mathbf{U}_k^{(r)} - \eta\left[\rho\,\mathbf{U}_k^{(r)} + \mathbf{G}(\mathbf{X}_k^{(r)}, t_k)^\top \mathbf{N}_{k+1}^{(r)}\right] \\ &= \mathbf{U}_k^{(r)} - \eta\left[\nabla^h J_h(\mathbf{U}^{(r)})\right]_k. \qquad \text{(By Eq. (28))}\end{aligned}$$

Note that per-block projection on the Cartesian product $\mathcal{C} = (\mathcal{U}^M)^K$ equals $\Pi_{\mathcal{C}}^h$, so aggregating all K steps together gives

$$\mathbf{U}^{(r+1)} = \Pi_{\mathcal{C}}^h\left(\mathbf{U}^{(r)} - \eta\,\nabla^h J_h(\mathbf{U}^{(r)})\right), \tag{59}$$

which is exactly PGD on J_h in the h-inner-product space with step size η and feasible set $\mathcal{C}$.

Part 2: Cost descent and Cluster Point Stationarity.
Existence of $J_h^{\inf}$. Under **Assumption C.4**, $\mathcal{C}$ is compact. J_h is continuous on $\mathcal{C}$ by the chain rule on Eq. (26) together with **Assumption C.3**. Hence,

$$J_h^{\inf} := \inf_{\mathbf{U}\in\mathcal{C}} J_h(\mathbf{U})$$

is finite and can be attained.

Per-iteration cost descent. By the descent lemma (Bauschke & Combettes, 2017, Lemma 1.2.3) (applied here in $\|\cdot\|_h$ with the Lipschitz constant L from Lemma C.10), we have the following for J_h:

$$J_h(\mathbf{U}^+) \le J_h(\mathbf{U}) + \langle \nabla^h J_h(\mathbf{U}), \mathbf{U}^+ - \mathbf{U}\rangle_h + \tfrac{L}{2}\|\mathbf{U}^+ - \mathbf{U}\|_h^2, \qquad \forall\,\mathbf{U}^+ \in \mathcal{C}.$$

Applying $\mathbf{U} = \mathbf{U}^{(r)}$ and $\mathbf{U}^+ = \mathbf{U}^{(r+1)}$ from Eq. (59) yields

$$J_h(\mathbf{U}^{(r+1)}) \le J_h(\mathbf{U}^{(r)}) + \langle \nabla^h J_h(\mathbf{U}^{(r)}), \mathbf{U}^{(r+1)} - \mathbf{U}^{(r)}\rangle_h + \tfrac{L}{2}\|\mathbf{U}^{(r+1)} - \mathbf{U}^{(r)}\|_h^2. \tag{Descent inequality}$$

We next obtain an upper bound for the inner product term in Eq. (Descent inequality). For notational ease, we first set

$$\mathbf{G}_r := \mathbf{G}_\eta(\mathbf{U}^{(r)}) = \eta^{-1}(\mathbf{U}^{(r)} - \mathbf{U}^{(r+1)}) \tag{60}$$

which gives

$$\mathbf{U}^{(r+1)} - \mathbf{U}^{(r)} = -\eta\,\mathbf{G}_r. \tag{61}$$

By the Projection Theorem (variational inequality) (Bauschke & Combettes, 2017, Theorem 3.16):

$$\langle \mathbf{U}^{(r+1)} - \boldsymbol{Z}, \mathbf{U}^{(r+1)} - \mathbf{V}\rangle_h \le 0, \quad \boldsymbol{Z} \in \mathbb{R}^{Mn\times K},\ \forall \mathbf{V} \in \mathcal{C}.$$

Let $\boldsymbol{Z} = \mathbf{U}^{(r)} - \eta\,\nabla^h J_h(\mathbf{U}^{(r)})$, *i.e.*, the gradient update step *before* projection, and $\mathbf{V} = \mathbf{U}^{(r)} \in \mathcal{C}$, and we will have the following:

$$\begin{aligned}\langle \mathbf{U}^{(r+1)} - \mathbf{U}^{(r)} + \eta\,\nabla^h J_h(\mathbf{U}^{(r)}), \mathbf{U}^{(r+1)} - \mathbf{U}^{(r)}\rangle_h &\le 0 \\ \langle \eta\nabla^h J_h(\mathbf{U}^{(r)}) - \eta\mathbf{G}_r, -\eta\mathbf{G}_r\rangle_h &\le 0 \qquad \text{(Per Eq. (60), Eq. (61))} \\ \eta\langle \nabla^h J_h(\mathbf{U}^{(r)}), -\eta\mathbf{G}_r\rangle_h + \eta^2\|\mathbf{G}_r\|_h^2 &\le 0 \qquad \text{(Linearity of inner product)} \\ \langle \nabla^h J_h(\mathbf{U}^{(r)}), -\eta\mathbf{G}_r\rangle_h &\le -\eta\|\mathbf{G}_r\|_h^2. \qquad (62)\end{aligned}$$

Substituting Eq. (61) and Eq. (62) into Eq. (Descent inequality) yields

$$\begin{aligned} J_h(\mathbf{U}^{(r+1)}) &\le J_h(\mathbf{U}^{(r)}) - \eta\|\mathbf{G}_r\|_h^2 + \tfrac{L\eta^2}{2}\|\mathbf{G}_r\|_h^2 \\ &= J_h(\mathbf{U}^{(r)}) - \eta(1 - L\eta/2)\|\mathbf{G}_r\|_h^2. \end{aligned} \tag{63}$$

For $\eta \le 1/L$ per Eq. (31) of Assumption C.6, $\eta(1 - L\eta/2) \ge \eta/2$. Hence,

$$J_h(\mathbf{U}^{(r+1)}) \;\le\; J_h(\mathbf{U}^{(r)}) - \frac{\eta}{2}\left\|\mathbf{G}_\eta(\mathbf{U}^{(r)})\right\|_h^2, \tag{64}$$

where per-iteration descent is established.

Full-sequence convergence. Telescoping Eq. (64) from iteration $r = 0$ to $I - 1$,

$$\begin{aligned} \sum_{r=0}^{I-1} \frac{\eta}{2}\,\|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h^2 \;&\le\; J_h(\mathbf{U}^{(0)}) - J_h(\mathbf{U}^{(I)}) \\ &\le\; J_h(\mathbf{U}^{(0)}) - J_h^{\inf}. \end{aligned} \tag{65}$$

Bounding the average on Eq. (65) by the minimum gives Eq. (32):

$$\min_{0\le r<I} \frac{\eta}{2}\,\|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h^2 \le \frac{1}{I}\sum_{r=0}^{I-1} \frac{\eta}{2}\,\|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h^2 \le \frac{1}{I}\left(J_h(\mathbf{U}^{(0)}) - J_h^{\inf}\right)$$

$$\min_{0\le r<I} \|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h^2 \;\le\; \frac{2\left(J_h(\mathbf{U}^{(0)}) - J_h^{\inf}\right)}{\eta\, I}.$$

Letting $I \to \infty$ in Eq. (65), the partial sums are bounded

$$\sum_{r=0}^{\infty} \|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h^2 < \infty.$$

Per (Rudin, 1976, Theorem 3.24) and the continuity of a quadratic function, we have

$$\lim_{r\to\infty} \|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h = 0$$

for the full sequence.

Continuity of h-norm of projected-gradient mapping. By Lemma C.10 and the nonexpansiveness of projector $\Pi_{\mathcal{C}}^h$ ((Bauschke & Combettes, 2017, Proposition 4.16)), the mapping $\mathbf{U} \mapsto \mathbf{G}_\eta(\mathbf{U})$ is continuous on $\mathcal{C}$. Since $\|\cdot\|_h$ is continuous, $\mathbf{U} \mapsto \|\mathbf{G}_\eta(\mathbf{U})\|_h$ is also continuous on $\mathcal{C}$.

Stationarity on cluster point(s). By (Rudin, 1976, Theorem 2.37), at least one cluster point exists for the sequence $\{\mathbf{U}^{(r)}\}$ due to compactness of $\mathcal{C}$. Then, for every cluster point $\mathbf{U}^*$ of $\{\mathbf{U}^{(r)}\}$, by definition there exists a subsequence $\{\mathbf{U}^{(r_l)}\}$ of $\{\mathbf{U}^{(r)}\}$ where $\lim_{l\to\infty} \mathbf{U}^{(r_l)} = \mathbf{U}^*$. Therefore, the continuity of $\|\mathbf{G}_\eta(\cdot)\|_h$ together with (Bartle & Sherbert, 2011, Theorem 3.4.2) gives

$$\lim_{l\to\infty} \|\mathbf{G}_\eta(\mathbf{U}^{(r_l)})\| = \lim_{r\to\infty} \|\mathbf{G}_\eta(\mathbf{U}^{(r)})\|_h = 0,$$

i.e., every cluster point $\mathbf{U}^*$ satisfies $\mathbf{G}_\eta(\mathbf{U}^*) = 0$. □

C.2.5 Remarks and Discussions

Stationarity, not optimality. Eq. (32) bounds the projected-gradient mapping but not the gap $\left[J_h(\mathbf{U}^{(I)}) - J_h^*\right]$ to global minima. Note that the cost function is generically **nonconvex** through the drift $\mathbf{f}^\theta$ and the terminal cost that possibly involve neural networks. As such, further objective-gap guarantee is hard to establish without further hypotheses.

Smoothness of $\Phi^\sharp$. Assumption C.3 requires *gradient* Lipschitzness of $\Phi^\sharp = \Phi \circ \hat{p}^u_{\mathbf{y}}$ on the reachable set. For reverse KL on the *floored simplex* $\Delta_\varepsilon := \{\mathbf{p} \in \Delta_{o-1} : p^{(j)} \geq \varepsilon/o\}$, which is also the case in practical implementations of our method, the Hessian $\nabla^2\Phi(\mathbf{p}) = \mathrm{diag}\left(1/p^{(j)}\right)$ has operator norm at most o/ε. This gives $L_{\Phi^\sharp} = O(o/\varepsilon)$, is polynomial in $1/\varepsilon$.

Hyperparameter discussion. The step-size hypothesis Eq. (31) reads $\xi \geq 1 - \rho/L$. We identify two cases: (i) If $L \leq \rho$, any $\xi \in [0, 1)$ satisfies it. (ii) If $L > \rho$ (typical scenario when involving neural networks), one needs ξ close to 1 (as reflected by the actual hyperparameters adopted in our experiments, disclosed in later sections), which is equivalently $\gamma \geq L - \rho$ on the proximal-weight.

C.3 Statistical Bounds for Distribution Alignment

This subsection complements the stationarity statement in **Proposition 3.2**. The stationarity result controls the first-order residual of the Euler-discretized objective. Results in this subsection explain what a small realized terminal distributional cost implies for the population attribute distribution. All statements concern the same Euler system used by Algorithm 1.

Remark C.11 (Batch-size scaling and scope). We consider *fixed* batch size M in this subsection. Since the empirical terminal distribution is formed by averaging over M samples, whereas the control-effort term sums over samples, the regularization coefficient ρ may be viewed as batch-size dependent when comparing different values of M: letting $\rho_M = \frac{\bar{\rho}}{M}$ yields an equivalent mean per-sample control regularization. The results below characterize the batch-size-M algorithm and should not be interpreted as an $M \to \infty$ consistency guarantee toward the target distribution.

C.3.1 Setup and Population Objects

Spaces and grid. Fix positive integers n, m, o, M, K, where n is the state dimension, m is the control dimension, o is the number of attribute classes, M is the batch size, and K is the number of Euler steps. Whenever hard-label or margin statements are used, we assume $o \geq 2$. Fix a horizon $T > 0$ and a grid

$$0 = t_0 < t_1 < \cdots < t_K = T, \qquad h_k := t_{k+1} - t_k > 0, \qquad \sum_{k=0}^{K-1} h_k = T.$$

For any sequence $\mathbf{z} = (\mathbf{z}_0, \ldots, \mathbf{z}_{K-1})$, set

$$\|\mathbf{z}\|_h^2 := \sum_{k=0}^{K-1} h_k \|\mathbf{z}_k\|^2.$$

Discrete sample-wise flow. For an initial state $\mathbf{x}_0 \in \mathbb{R}^n$ and a control sequence $\mathbf{u} = (\mathbf{u}_0, \ldots, \mathbf{u}_{K-1}) \in (\mathbb{R}^m)^K$, write $\mathbf{x}_K(\mathbf{x}_0, \mathbf{u})$ for the terminal state of the Euler recursion

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h_k \left[\mathbf{f}^\theta(\mathbf{x}_k, t_k) + \mathbf{g}(\mathbf{x}_k, t_k)\mathbf{u}_k\right], \qquad k = 0, \ldots, K-1. \tag{66}$$

For a realized batch, the sample-wise controls $\mathbf{U} = (\mathbf{u}^{[1]}, \ldots, \mathbf{u}^{[M]})$ produce $\mathbf{x}_K^{[i]} := \mathbf{x}_K(\mathbf{x}_0^{[i]}, \mathbf{u}^{[i]})$.

Soft and hard attribute distributions. Let the differentiable attribute oracle have logits $\Psi : \mathbb{R}^n \to \mathbb{R}^o$ and let, for a temperature $\tau > 0$,

$$V(\mathbf{x}) := \mathrm{softmax}_\tau(\Psi(\mathbf{x})) \in \Delta_{o-1}, \qquad V^{(j)}(\mathbf{x}) = \frac{\exp(\Psi^{(j)}(\mathbf{x})/\tau)}{\sum_{\ell=1}^{o} \exp(\Psi^{(\ell)}(\mathbf{x})/\tau)}. \tag{67}$$

The empirical soft distribution, empirical hard histogram, and realized terminal cost are

$$\hat{p} := \frac{1}{M} \sum_{i=1}^{M} V(\mathbf{x}_K^{[i]}), \qquad \hat{p}_{\mathrm{hard}} := \frac{1}{M} \sum_{i=1}^{M} \mathbf{e}_{\hat{y}_i}, \qquad \hat{\varepsilon} := \mathbb{D}_{\mathrm{KL}}\left[\hat{p} \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right], \tag{68}$$

where

$$\hat{y}_i := \min \arg\max_j \Psi^{(j)}(\mathbf{x}_K^{[i]}) \tag{69}$$

uses a deterministic tie-break. Let γ_i be the margin between the largest logit minus the second largest logit at $\mathbf{x}_K^{[i]}$:

$$\gamma_i := \Psi^{(\hat{y}_i)}(\mathbf{x}_K^{[i]}) - \max_{j \neq \hat{y}_i} \Psi^{(j)}(\mathbf{x}_K^{[i]}) \tag{70}$$

and also define the proportion of samples residing in the α-sublevel set of all γ_i in the batch as

$$\hat{\beta}_\alpha := \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\{\gamma_i < \alpha\}, \qquad \alpha > 0. \tag{71}$$

We use total variation $\mathrm{TV}(p, q) := \frac{1}{2}\|p - q\|_1$ throughout.

Population object for sample-wise DADA. For the population statement, the initial states are i.i.d. $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ and Algorithm 1 is viewed as a deterministic fixed-iteration map from the initial-state batch to its returned controls and terminal states. Since each sample's control depends on the whole batch, there is no fixed control to plug into a standard population objective. The relevant object is the per-sample marginal of the coupled batch-size-M algorithm,

$$\bar{p}_M := \mathbb{E}[\hat{p}] = \mathbb{E}[V(\mathbf{x}_K^{[1]})] \in \Delta_{o-1}, \tag{72}$$

where the second equality follows from permutation equivariance of the zero-initialized sample-wise algorithm.

C.3.2 Assumptions

The following population-bound assumptions are global assumptions for the sample-wise, optimize-then-generate statement. They are separate from the local fixed-batch assumptions used for stationarity in Section C.2.2.

Assumption C.12 (Dynamics regularity)**.** For all grid times t_k, the drift and actuator satisfy, for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$,

$$\|\mathbf{f}^\theta(\mathbf{x}, t_k) - \mathbf{f}^\theta(\mathbf{x}', t_k)\| \le \ell_f \|\mathbf{x} - \mathbf{x}'\|, \qquad b_f := \max_k \|\mathbf{f}^\theta(\mathbf{0}, t_k)\| < \infty,$$
$$\|\mathbf{g}(\mathbf{x}, t_k)\|_{\mathrm{op}} \le b_g, \qquad \|\mathbf{g}(\mathbf{x}, t_k) - \mathbf{g}(\mathbf{x}', t_k)\|_{\mathrm{op}} \le \ell_g \|\mathbf{x} - \mathbf{x}'\|. \tag{73}$$

Assumption C.13 (Attribute oracle and target floor)**.** The logit map $\Psi : \mathbb{R}^n \to \mathbb{R}^o$ is globally ℓ_Ψ-Lipschitz:

$$\|\Psi(\mathbf{x}) - \Psi(\mathbf{x}')\| \le \ell_\Psi \|\mathbf{x} - \mathbf{x}'\|, \qquad \forall\, \mathbf{x}, \mathbf{x}' \in \mathbb{R}^n.$$

For KL-form statements and replacement-sensitivity constants, the logits are also uniformly bounded: $\|\Psi(\mathbf{x})\|_\infty \le b_\Psi$ for all $\mathbf{x}$. The target satisfies $p_{\mathbf{y}}^{\mathrm{tar,(j)}} \ge q_{\min} > 0$ for every class j.

Assumption C.14 (Admissible controls and projected algorithm convention)**.** The admissible control set $\mathcal{U} \subseteq \mathbb{R}^m$ is nonempty, closed, convex, and satisfies $\|\mathbf{u}\| \le b_u$ for all $\mathbf{u} \in \mathcal{U}$ and $\mathbf{0} \in \mathcal{U}$. The projected variant of **Algorithm 1** is used with fixed outer iteration budget I, zero initialization $\mathbf{u}_k^{[i],(0)} = \mathbf{0}$, deterministic Euler/VJP/projection operations, and componentwise projection onto $\mathcal{U}$ after every update. Thus the mapping

$$(\mathbf{x}_0^{[1]}, \ldots, \mathbf{x}_0^{[M]}) \longmapsto \mathbf{U}^{(I)} \longmapsto \hat{p}$$

has no internal randomness once the initial states are fixed.

Assumption C.15 (Global Jacobian regularity for one-initial-state replacement)**.** In addition to **Assumption C.12** and **Assumption C.13**, the maps $\mathbf{f}^\theta(\cdot, t_k)$, $\mathbf{g}(\cdot, t_k)$, and Ψ are C^1 with globally bounded and Lipschitz Jacobians, uniformly over grid times:

$$\|\nabla_{\mathbf{x}} \mathbf{f}^\theta(\mathbf{x}, t_k)\|_{\mathrm{op}} \le b_{\nabla f}, \qquad \|\nabla_{\mathbf{x}} \mathbf{f}^\theta(\mathbf{x}, t_k) - \nabla_{\mathbf{x}} \mathbf{f}^\theta(\mathbf{x}', t_k)\|_{\mathrm{op}} \le \ell_{\nabla f} \|\mathbf{x} - \mathbf{x}'\|,$$
$$\|D_{\mathbf{x}} \mathbf{g}(\mathbf{x}, t_k)\|_{\mathrm{op}} \le b_{\nabla g}, \qquad \|D_{\mathbf{x}} \mathbf{g}(\mathbf{x}, t_k) - D_{\mathbf{x}} \mathbf{g}(\mathbf{x}', t_k)\|_{\mathrm{op}} \le \ell_{\nabla g} \|\mathbf{x} - \mathbf{x}'\|,$$
$$\|\nabla \Psi(\mathbf{x})\|_{\mathrm{op}} \le \ell_\Psi, \qquad \|\nabla \Psi(\mathbf{x}) - \nabla \Psi(\mathbf{x}')\|_{\mathrm{op}} \le \ell_{\nabla \Psi} \|\mathbf{x} - \mathbf{x}'\|. \tag{74}$$

The bound on $D_{\mathbf{x}} \mathbf{g}$ is in the induced bilinear sense used for the controlled drift.

Remark 6. The assumptions above are stronger than the local fixed-batch assumptions used for **Proposition C.7**. The stationarity result works on a fixed realized batch and a reachable set, whereas the population bound replaces that batch by Gaussian initial-state draws and controls arbitrary one-initial-state replacements.

C.3.3 Population Alignment Proof Ingredients

Replacement-sensitivity constants. The source concentration package denotes the sample-wise stability radius by $\varepsilon_{\mathrm{ustab}}$; we keep the paper notation $\varepsilon_{\mathrm{rep}}$. The derived constants are introduced at their first logical use in the definitions, lemma statements, and proposition statements below. Table 5 is only a lookup table: it aggregates the same definitions and points to their introduction sites. The primitive bounds $b_g, \ell_f, \ell_g, \ell_\Psi, b_\Psi, q_{\min}, b_u, b_{\nabla f}, b_{\nabla g}, \ell_{\nabla f}, \ell_{\nabla g}, \ell_{\nabla\Psi}$ are supplied by **Assumption C.12–Assumption C.15**.

Table 5: Summary of derived constants and bounds used in the proof of Proposition C.25.

Symbol	Definition	Role	First appearance
Flow and attribute constants			
E_u	$\exp((\ell_f + \ell_g b_u)T)$	Gronwall factor for controlled flow	Eq. (83)
L_{xu}	$b_g\sqrt{T}\,E_u$	control-to-terminal-state Lipschitz bound	Eq. (83)
ℓ_V	ℓ_Ψ/τ	soft attribute Lipschitz bound	Eq. (91)
L_w	$\sqrt{o}\,\ell_V L_{xu}$	control-to-soft Lipschitz bound in ℓ_1	Eq. (94)
$v_{\min}$	$\exp(-2b_\Psi/\tau)/o$	softmax floor for all coordinates	Eq. (95)
$L_\Phi(v)$	$1 + \log(1/\min\{v, q_{\min}\})$	KL Lipschitz bound on a floored simplex	Eq. (96)
Jacobian, costate, and update constants			
b_j	$b_{\nabla f} + b_{\nabla g} b_u$	controlled-drift Jacobian bound	Eq. (102)
$\ell_{j,x}$	$\ell_{\nabla f} + \ell_{\nabla g} b_u$	Jacobian Lipschitz bound in state	Eq. (103)
$\ell_{j,u}$	$b_{\nabla g}$	Jacobian Lipschitz bound in control	Eq. (103)
$\ell_{\nabla V}$	$\ell_{\nabla\Psi}/\tau + 3\ell_\Psi^2/\tau^2$	Lipschitz bound for $x \mapsto \nabla V(x)$	Eq. (101)
$B_{\nabla\Phi}$	$\sqrt{o}\,L_\Phi(v_{\min})$	KL-gradient norm bound	Eq. (101)
$L_{\nabla\Phi}$	$1/v_{\min}$	KL-gradient Lipschitz bound	Eq. (101)
B_ν	$\ell_V B_{\nabla\Phi} \exp(b_j T)$	per-sample costate bound, before the $1/M$ factor	Eq. (102)
$C_{\nu,d}$	$\exp(b_j T)[\ell_{\nabla V} L_{xu} B_{\nabla\Phi}$ $+$ $B_\nu(\ell_{j,x} L_{xu} T + \ell_{j,u}\sqrt{T})]$	costate deviation from a control perturbation	Eq. (104)
$C_{\nu,\pi}$	$\exp(b_j T)\ell_V/v_{\min}$	costate deviation from a soft-mean perturbation	Eq. (104)
A	$\sqrt{T}(\ell_g L_{xu} B_\nu + b_g C_{\nu,d})$	update-deviation coefficient for the state channel	Eq. (105)
$B_\star$	$\sqrt{T}\,b_g C_{\nu,\pi}$	update-deviation coefficient for the mean channel	Eq. (105)
Stability recursion and replacement radius			
κ	$\xi + \eta(A + B_\star L_w)/M$	M-dependent recursion factor	Eq. (106)
$\bar\kappa$	$\xi + \eta(A + B_\star L_w)$	M-free recursion factor for fixed-I bounds	Eq. (107)
M_0	$2\eta(A + B_\star L_w)/(1-\xi)$	mean-field threshold giving $\kappa < 1$	Eq. (107)
Θ_I	$2/(1-\xi)$ or $I\max(1,\bar\kappa)^{I-1}$	geometric-sum stability factor	Eq. (107)
c_M	$\eta B_\star L_w \Theta_I / M$	finite-M feedback correction	Eq. (108)
c_{rep}	$2(1 + c_M)$	ℓ_1 one-initial-state bounded-difference constant	Eq. (108)
$c_{\mathrm{rep}}^{(2)}$	$\sqrt{2} + 2c_M$	Euclidean one-initial-state bounded-difference constant	Eq. (108)
$\varepsilon_{\mathrm{rep}}(M,\delta)$	$c_{\mathrm{rep}}^{(2)}\sqrt{o/(2M)}$ $+$ $c_{\mathrm{rep}}\sqrt{\log(1/\delta)/(2M)}$	concentration radius around $\bar p_M$	Eq. (124)

Lemma C.16 (Discrete Gronwall inequality; from Lemma 11.2 of (Quarteroni et al., 2006))**.** *Let $a_0, \dots, a_K$ and $c_0, \dots, c_{K-1}, d_0, \dots, d_{K-1}$ be nonnegative and suppose*

$$a_{k+1} \le (1 + c_k)a_k + d_k, \qquad k = 0, \dots, K-1.$$

Then, for every $0 \le r \le k \le K$,

$$a_k \le a_r \prod_{\ell=r}^{k-1}(1 + c_\ell) + \sum_{j=r}^{k-1} d_j \prod_{\ell=j+1}^{k-1}(1 + c_\ell), \tag{75}$$

with empty products equal to one and empty sums equal to zero. In particular,

$$a_k \le \left(a_r + \sum_{j=r}^{k-1} d_j \right) \exp\left(\sum_{\ell=r}^{k-1} c_\ell \right). \tag{76}$$

Lemma C.17 (Reachable-set bound)**.** *Under* **Assumption C.12** *and* **Assumption C.14***, for any initial state $\mathbf{x}_0 \in \mathbb{R}^n$ and any admissible control sequence $\mathbf{u} = (\mathbf{u}_0, \dots, \mathbf{u}_{K-1}) \in \mathcal{U}^K$, the Euler path in Eq. (66) satisfies*

$$\|\mathbf{x}_k(\mathbf{x}_0, \mathbf{u})\| \le (\|\mathbf{x}_0\| + (b_f + b_g b_u)T) \exp(\ell_f T), \qquad 0 \le k \le K. \tag{77}$$

Proof. Write $\mathbf{x}_k := \mathbf{x}_k(\mathbf{x}_0, \mathbf{u})$ and set $a_k := \|\mathbf{x}_k\|$. One Euler step gives

$$a_{k+1} \le a_k + h_k \underbrace{\|\mathbf{f}^\theta(\mathbf{x}_k, t_k)\|}_{(\text{drift})_k} + h_k \underbrace{\|\mathbf{g}(\mathbf{x}_k, t_k)\mathbf{u}_k\|}_{(\text{act})_k}. \tag{78}$$

By **Assumption C.12**,

$$(\text{drift})_k \le \|\mathbf{f}^\theta(\mathbf{x}_k, t_k) - \mathbf{f}^\theta(\mathbf{0}, t_k)\| + \|\mathbf{f}^\theta(\mathbf{0}, t_k)\| \le \ell_f a_k + b_f, \tag{79}$$

$$(\text{act})_k \le \|\mathbf{g}(\mathbf{x}_k, t_k)\|_{\text{op}} \|\mathbf{u}_k\| \le b_g b_u, \tag{80}$$

The bound on $(\text{act})_k$ uses the admissible-control bound $\|\mathbf{u}_k\| \le b_u$ from **Assumption C.14**. Combining Eq. (78)–Eq. (80),

$$a_{k+1} \le (1 + \ell_f h_k) a_k + (b_f + b_g b_u) h_k.$$

Applying Lemma C.16 with $c_k = \ell_f h_k$ and $d_k = (b_f + b_g b_u) h_k$ yields

$$\begin{aligned} a_k &\le \left(\|\mathbf{x}_0\| + (b_f + b_g b_u) \sum_{j=0}^{k-1} h_j \right) \exp\left(\ell_f \sum_{j=0}^{k-1} h_j \right) \\ &\le (\|\mathbf{x}_0\| + (b_f + b_g b_u)T) \exp(\ell_f T) \qquad (\textstyle\sum_{j=0}^{k-1} h_j \le T) \end{aligned}$$

and Eq. (77) is proved. □

Lemma C.18 (Flow Lipschitzness)**.** *Under* **Assumption C.12** *and* **Assumption C.14***, for any admissible controls $\mathbf{u}, \mathbf{u}' \in \mathcal{U}^K$ and any initial states $\mathbf{x}_0, \mathbf{x}_0' \in \mathbb{R}^n$, the Euler flow in Eq. (66) satisfies, for every $0 \le k \le K$,*

$$\|\mathbf{x}_k(\mathbf{x}_0, \mathbf{u}) - \mathbf{x}_k(\mathbf{x}_0', \mathbf{u})\| \le E_u \|\mathbf{x}_0 - \mathbf{x}_0'\|, \tag{81}$$

$$\|\mathbf{x}_k(\mathbf{x}_0, \mathbf{u}) - \mathbf{x}_k(\mathbf{x}_0, \mathbf{u}')\| \le L_{xu} \|\mathbf{u} - \mathbf{u}'\|_h. \tag{82}$$

where

$$E_u := \exp\left((\ell_f + \ell_g b_u)T\right), \qquad L_{xu} := b_g \sqrt{T}\, E_u. \tag{83}$$

In particular, at $k = K$,

$$\|\mathbf{x}_K(\mathbf{x}_0, \mathbf{u}) - \mathbf{x}_K(\mathbf{x}_0', \mathbf{u})\| \le E_u \|\mathbf{x}_0 - \mathbf{x}_0'\|, \qquad \|\mathbf{x}_K(\mathbf{x}_0, \mathbf{u}) - \mathbf{x}_K(\mathbf{x}_0, \mathbf{u}')\| \le L_{xu} \|\mathbf{u} - \mathbf{u}'\|_h. \tag{84}$$

Proof. **Part 1: Initial state sensitivity.** Write

$$\begin{aligned} \mathbf{x}_k &:= \mathbf{x}_k(\mathbf{x}_0, \mathbf{u}) \\ \mathbf{x}_k' &:= \mathbf{x}_k(\mathbf{x}_0', \mathbf{u}) \\ \delta_k &:= \|\mathbf{x}_k - \mathbf{x}_k'\|. \end{aligned}$$

Subtracting the Euler recursions with common control $\mathbf{u}$ gives

$$\mathbf{x}_{k+1} - \mathbf{x}_{k+1}' = \mathbf{x}_k - \mathbf{x}_k' + h_k \underbrace{\left[\mathbf{f}^\theta(\mathbf{x}_k, t_k) - \mathbf{f}^\theta(\mathbf{x}_k', t_k)\right]}_{(\text{drift})_k} + h_k \underbrace{\left[\mathbf{g}(\mathbf{x}_k, t_k) - \mathbf{g}(\mathbf{x}_k', t_k)\right] \mathbf{u}_k}_{(\text{act-state})_k}. \tag{85}$$

By **Assumption C.12** and **Assumption C.14**, the two labeled terms satisfy

$$\|(\text{drift})_k\| \le \ell_f \delta_k,$$
$$\|(\text{act-state})_k\| \le \ell_g \delta_k \|\mathbf{u}_k\| \le \ell_g b_u \delta_k.$$

Hence,

$$\delta_{k+1} \le (1 + h_k(\ell_f + \ell_g b_u))\, \delta_k.$$

Applying Lemma C.16 with $c_k = h_k(\ell_f + \ell_g b_u)$ and $d_k = 0$ yields,

$$\delta_k \le \exp\left((\ell_f + \ell_g b_u) \sum_{j=0}^{k-1} h_j \right) \|\mathbf{x}_0 - \mathbf{x}'_0\| \le E_u \|\mathbf{x}_0 - \mathbf{x}'_0\|, \quad \forall 0 \le k \le K,$$

which proves Eq. (81).

Part 2: Control sensitivity. Rewrite

$$\mathbf{x}_k := \mathbf{x}_k(\mathbf{x}_0, \mathbf{u})$$
$$\mathbf{x}'_k := \mathbf{x}_k(\mathbf{x}_0, \mathbf{u}')$$
$$\delta_k := \|\mathbf{x}_k - \mathbf{x}'_k\|$$

with $\delta_0 = 0$. Subtracting the recursions and decomposing the increment gives

$$\begin{aligned}\mathbf{x}_{k+1} - \mathbf{x}'_{k+1} = \mathbf{x}_k - \mathbf{x}'_k + h_k \underbrace{\left[\mathbf{f}^\theta(\mathbf{x}_k, t_k) - \mathbf{f}^\theta(\mathbf{x}'_k, t_k)\right]}_{(\text{drift})_k} \\ + h_k \underbrace{\left[\mathbf{g}(\mathbf{x}_k, t_k) - \mathbf{g}(\mathbf{x}'_k, t_k)\right] \mathbf{u}_k}_{(\text{state-act})_k} + h_k \underbrace{\mathbf{g}(\mathbf{x}'_k, t_k)(\mathbf{u}_k - \mathbf{u}'_k)}_{(\text{ctrl-act})_k}. \end{aligned} \tag{86}$$

Again, by **Assumption C.12** and **Assumption C.14** we have

$$\|(\text{drift})_k\| \le \ell_f \delta_k, \qquad \|(\text{state-act})_k\| \le \ell_g b_u \delta_k, \qquad \|(\text{ctrl-act})_k\| \le b_g \|\mathbf{u}_k - \mathbf{u}'_k\|. \tag{87}$$

Thus,

$$\delta_{k+1} \le (1 + h_k(\ell_f + \ell_g b_u))\, \delta_k + h_k b_g \|\mathbf{u}_k - \mathbf{u}'_k\|.$$

Applying Lemma C.16 with $c_k = h_k(\ell_f + \ell_g b_u)$ and $d_k = h_k b_g \|\mathbf{u}_k - \mathbf{u}'_k\|$ gives

$$\delta_k \le b_g E_u \sum_{j=0}^{k-1} h_j \|\mathbf{u}_j - \mathbf{u}'_j\| \le b_g E_u \sum_{j=0}^{K-1} h_j \|\mathbf{u}_j - \mathbf{u}'_j\|, \quad \forall 0 \le k \le K. \tag{88}$$

Converting the RHS sum of Eq. (88) into h-norm yields

$$\begin{aligned}\sum_{j=0}^{K-1} h_j \|\mathbf{u}_j - \mathbf{u}'_j\| &= \sum_{j=0}^{K-1} \sqrt{h_j}\, \sqrt{h_j} \|\mathbf{u}_j - \mathbf{u}'_j\| \\ &\le \sqrt{\sum_{j=0}^{K-1} h_j} \sqrt{\sum_{j=0}^{K-1} h_j \|\mathbf{u}_j - \mathbf{u}'_j\|^2} \qquad \text{(Cauchy–Schwarz)} \\ &= \sqrt{T} \|\mathbf{u} - \mathbf{u}'\|_h. \end{aligned} \tag{89}$$

Combining Eq. (88), Eq. (89), and $L_{xu} = b_g \sqrt{T}\, E_u$ proves Eq. (82). □

Lemma C.19 (Soft attribute Lipschitzness and floor)**.** *Under* **Assumption C.13***, the map* $\text{softmax}_\tau : \mathbb{R}^o \to \Delta_{o-1}$ *is* $(1/\tau)$*-Lipschitz in Euclidean norm:*

$$\|\,\text{softmax}_\tau(z) - \text{softmax}_\tau(z')\| \le \frac{1}{\tau} \|z - z'\|, \qquad \forall z, z' \in \mathbb{R}^o \tag{90}$$

where

$$\ell_V := \frac{\ell_\Psi}{\tau} \tag{91}$$

Consequently,

$$\|V(\mathbf{x}) - V(\mathbf{x}')\| \le \ell_V \|\mathbf{x} - \mathbf{x}'\|, \qquad \|V(\mathbf{x}) - V(\mathbf{x}')\|_1 \le \sqrt{o}\, \ell_V \|\mathbf{x} - \mathbf{x}'\|. \tag{92}$$

If the assumptions of Lemma C.18 also hold, then for any initial state $\mathbf{x}_0$ and admissible controls $\mathbf{u}, \mathbf{u}' \in \mathcal{U}^K$,

$$\|V(\mathbf{x}_K(\mathbf{x}_0, \mathbf{u})) - V(\mathbf{x}_K(\mathbf{x}_0, \mathbf{u}'))\|_1 \le L_w \|\mathbf{u} - \mathbf{u}'\|_h \tag{93}$$

where

$$L_w := \sqrt{o}\, \ell_V L_{xu} \tag{94}$$

with L_{xu} defined in Eq. (83). Additionally, $V(\mathbf{x}) \in \Delta_{\ge v_{\min}}$ for every $\mathbf{x}$, where

$$v_{\min} := \frac{\exp(-2b_\Psi/\tau)}{o}, \tag{95}$$

and $\Delta_{\ge v} := \{p \in \Delta_{o-1} : p^{(j)} \ge v,\ \forall j\}$. Convex averages and coordinatewise expectations preserve this floor; in particular, $\hat{p}, \bar{p}_M \in \Delta_{\ge v_{\min}}$.

Proof. Write $s(z) := \mathrm{softmax}_\tau(z)$. Its Jacobian

$$\mathrm{D}s(z) = \tau^{-1} \left(\mathrm{diag}(s(z)) - s(z)s(z)^\top\right)$$

gives $\|\mathrm{D}s(z)\|_{\mathrm{op}} \le 1/\tau$ for every z, because the matrix in parentheses is the covariance matrix of a one-hot random vector with law $s(z)$. The mean-value theorem proves Eq. (90). Composing this bound with the Lipschitz bound on Ψ and then using Cauchy–Schwarz gives

$$\|V(\mathbf{x}) - V(\mathbf{x}')\| \le \frac{\ell_\Psi}{\tau} \|\mathbf{x} - \mathbf{x}'\| = \ell_V \|\mathbf{x} - \mathbf{x}'\|, \tag{attribute-Lipschitz}$$

$$\|V(\mathbf{x}) - V(\mathbf{x}')\|_1 \le \sqrt{o} \|V(\mathbf{x}) - V(\mathbf{x}')\| \le \sqrt{o}\, \ell_V \|\mathbf{x} - \mathbf{x}'\|. \tag{Cauchy–Schwarz}$$

By **Assumption C.13**, for every coordinate j,

$$\begin{aligned} V^{(j)}(\mathbf{x}) &= \frac{\exp(\Psi^{(j)}(\mathbf{x})/\tau)}{\sum_{\ell=1}^{o} \exp(\Psi^{(\ell)}(\mathbf{x})/\tau)} \\ &\ge \frac{\exp(-b_\Psi/\tau)}{o \exp(b_\Psi/\tau)} = v_{\min}. \end{aligned} \tag{floor}$$

The preservation claim regarding the floor on $\hat{p}$ and $\bar{p}_M$ follows coordinatewise from convexity. Combining the second bound in Eq. (84) with Eq. (92) proves Eq. (93). □

Lemma C.20 (KL Lipschitzness on a floored simplex)**.** *Under* **Assumption C.13***, for any $v \in (0, 1/o]$ and any $p, p' \in \Delta_{\ge v}$, the following holds:*

$$\left|\mathbb{D}_{KL}\left[p \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right] - \mathbb{D}_{KL}\left[p' \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right]\right| \le L_\Phi(v) \|p - p'\|_1 \tag{96}$$

where

$$L_\Phi(v) = 1 + \log \frac{1}{\min\{v, q_{\min}\}}. \tag{97}$$

Proof. Let $\Phi(p) := \mathbb{D}_{\mathrm{KL}}\left[p \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right]$ and write $q := p_{\mathbf{y}}^{\mathrm{tar}}$. For $p_\sigma := p + \sigma(p' - p)$, convexity of $\Delta_{\ge v}$ gives $p_\sigma \in \Delta_{\ge v}$ for every $\sigma \in [0, 1]$. On this segment,

$$\nabla\Phi(p_\sigma)^{(j)} = 1 + \log \frac{p_\sigma^{(j)}}{q^{(j)}}.$$

Since $p_\sigma^{(j)} \in [v, 1]$ and $q^{(j)} \in [q_{\min}, 1]$,

$$\|\nabla\Phi(p_\sigma)\|_\infty \le 1 + \log \frac{1}{\min\{v, q_{\min}\}} = L_\Phi(v). \tag{gradient-bound}$$

The fundamental theorem of calculus along the segment gives

$$\begin{aligned} |\Phi(p) - \Phi(p')| &= \left| \int_0^1 \langle \nabla\Phi(p_\sigma), p - p' \rangle \, \mathrm{d}\sigma \right| \\ &\le \int_0^1 \|\nabla\Phi(p_\sigma)\|_\infty \|p - p'\|_1 \, \mathrm{d}\sigma \le L_\Phi(v) \|p - p'\|_1. \end{aligned} \tag{mean-value}$$

□

Lemma C.21 (Softmax-Jacobian and gradient regularity)**.** *Let* $s(z) := \mathrm{softmax}_\tau(z)$*,* $J_{\mathrm{sm}}(z) := \mathrm{D}s(z)$ *and* $\Phi(p) := \mathbb{D}_{KL}\left[p \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right]$*. Under* **Assumption C.13** *and* **Assumption C.15***, the following statements hold:*

(i) For all $z, z' \in \mathbb{R}^o$*, the softmax Jacobian satisfies*

$$\|J_{\mathrm{sm}}(z)\|_{\mathrm{op}} \le \frac{1}{\tau}, \qquad \|J_{\mathrm{sm}}(z) - J_{\mathrm{sm}}(z')\|_{\mathrm{op}} \le \frac{3}{\tau^2}\|z - z'\|. \tag{98}$$

(ii) The map $V = s \circ \Psi$ *is* C^1 *with*

$$\nabla V(\mathbf{x}) = J_{\mathrm{sm}}(\Psi(\mathbf{x}))\nabla\Psi(\mathbf{x}), \qquad \|\nabla V(\mathbf{x})\|_{\mathrm{op}} \le \ell_V,$$

where ℓ_V *is defined in Eq. (91). Moreover, for all* $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$*,*

$$\|\nabla V(\mathbf{x}) - \nabla V(\mathbf{x}')\|_{\mathrm{op}} \le \ell_{\nabla V}\|\mathbf{x} - \mathbf{x}'\|. \tag{99}$$

(iii) For all $p, p' \in \Delta_{\ge v_{\min}}$*,*

$$\|\nabla\Phi(p)\| \le B_{\nabla\Phi}, \qquad \|\nabla\Phi(p) - \nabla\Phi(p')\| \le L_{\nabla\Phi}\|p - p'\|_1. \tag{100}$$

The constants in parts (ii) and (iii) are

$$\ell_{\nabla V} := \frac{\ell_{\nabla\Psi}}{\tau} + \frac{3\ell_\Psi^2}{\tau^2}, \qquad B_{\nabla\Phi} := \sqrt{o}\, L_\Phi(v_{\min}), \qquad L_{\nabla\Phi} := \frac{1}{v_{\min}}. \tag{101}$$

Proof. **Part (i).** The softmax Jacobian calculation in Lemma C.19 gives

$$J_{\mathrm{sm}}(z) = \frac{1}{\tau}\left(\mathrm{diag}(s(z)) - s(z)s(z)^\top\right).$$

The matrix in parentheses is the covariance matrix of a one-hot random vector with law $s(z)$, hence has operator norm at most one. Therefore $\|J_{\mathrm{sm}}(z)\|_{\mathrm{op}} \le 1/\tau$.

Write $s = s(z)$, $\tilde{s} = s(z')$, and $\Delta s = s - \tilde{s}$. Since

$$\tau\left[J_{\mathrm{sm}}(z) - J_{\mathrm{sm}}(z')\right] = \mathrm{diag}(\Delta s) - \left(ss^\top - \tilde{s}\tilde{s}^\top\right),$$

we have

$$\|\,\mathrm{diag}(\Delta s)\|_{\mathrm{op}} \le \|\Delta s\|, \qquad ss^\top - \tilde{s}\tilde{s}^\top = s(\Delta s)^\top + \Delta s\,\tilde{s}^\top.$$

Thus,

$$\begin{aligned}
\|J_{\mathrm{sm}}(z) - J_{\mathrm{sm}}(z')\|_{\mathrm{op}} &\le \frac{1}{\tau}\left(\|\,\mathrm{diag}(\Delta s)\|_{\mathrm{op}} + \|ss^\top - \tilde{s}\tilde{s}^\top\|_{\mathrm{op}}\right) \\
&\le \frac{1}{\tau}\left[\|\Delta s\| + (\|s\|\,\|\Delta s\| + \|\Delta s\|\,\|\tilde{s}\|)\right] && (\|ab^\top\|_{\mathrm{op}} = \|a\|\|b\|) \\
&\le \frac{3}{\tau}\|\Delta s\| && (s, \tilde{s} \in \Delta_{o-1}) \\
&\le \frac{3}{\tau}\cdot\frac{1}{\tau}\|z - z'\| && (\text{Lemma C.19}) \\
&= \frac{3}{\tau^2}\|z - z'\|,
\end{aligned}$$

proving Eq. (98).

Part (ii). The chain rule gives $\nabla V(\mathbf{x}) = J_{\mathrm{sm}}(\Psi(\mathbf{x}))\nabla\Psi(\mathbf{x})$ and $\|\nabla V(\mathbf{x})\|_{\mathrm{op}} \le \ell_\Psi/\tau = \ell_V$. Adding and subtracting the mixed product yields

$$\begin{aligned}
\nabla V(\mathbf{x}) - \nabla V(\mathbf{x}') = J_{\mathrm{sm}}(\Psi(\mathbf{x}))\left[\nabla\Psi(\mathbf{x}) - \nabla\Psi(\mathbf{x}')\right] \\
+ \left[J_{\mathrm{sm}}(\Psi(\mathbf{x})) - J_{\mathrm{sm}}(\Psi(\mathbf{x}'))\right]\nabla\Psi(\mathbf{x}').
\end{aligned}$$

Combining Eq. (98), $\|\nabla\Psi\|_{\mathrm{op}} \le \ell_\Psi$ and $\ell_{\nabla\Psi}$-Lipschitzness of $\nabla\Psi$ from **Assumption C.15**, and ℓ_Ψ-Lipschitzness of Ψ from **Assumption C.13**, proves Eq. (99).

Part (iii). Finally, for $p \in \Delta_{\geq v_{\min}}$,

$$\nabla\Phi(p)^{(j)} = 1 + \log \frac{p^{(j)}}{p_{\mathbf{y}}^{\mathrm{tar,(j)}}}.$$

Since $p^{(j)} \in [v_{\min}, 1]$ and $p_{\mathbf{y}}^{\mathrm{tar,(j)}} \in [q_{\min}, 1]$,

$$\log v_{\min} \leq \log p^{(j)} \leq 0, \qquad 0 \leq -\log p_{\mathbf{y}}^{\mathrm{tar,(j)}} \leq \log \frac{1}{q_{\min}}.$$

Therefore

$$\log v_{\min} \leq \log \frac{p^{(j)}}{p_{\mathbf{y}}^{\mathrm{tar,(j)}}} \leq \log \frac{1}{q_{\min}},$$

and hence

$$\left| 1 + \log \frac{p^{(j)}}{p_{\mathbf{y}}^{\mathrm{tar,(j)}}} \right| \leq 1 + \log \frac{1}{\min\{v_{\min}, q_{\min}\}} = L_\Phi(v_{\min}),$$

where the last equality is Eq. (97). Taking the maximum over j gives $\|\nabla\Phi(p)\|_\infty \leq L_\Phi(v_{\min})$, and therefore

$$\|\nabla\Phi(p)\| \leq \sqrt{o}\, L_\Phi(v_{\min}) = B_{\nabla\Phi}.$$

For the Lipschitz bound, the constant 1 and the target term cancel coordinatewise:

$$\nabla\Phi(p)^{(j)} - \nabla\Phi(p')^{(j)} = \log p^{(j)} - \log p'^{(j)}.$$

By the scalar mean-value theorem, for some θ_j between $p^{(j)}$ and $p'^{(j)}$,

$$\log p^{(j)} - \log p'^{(j)} = \frac{p^{(j)} - p'^{(j)}}{\theta_j}.$$

Since $p, p' \in \Delta_{\geq v_{\min}}$, the interval between the two coordinates lies in $[v_{\min}, 1]$, so $\theta_j \geq v_{\min}$ and

$$\left| \nabla\Phi(p)^{(j)} - \nabla\Phi(p')^{(j)} \right| \leq \frac{|p^{(j)} - p'^{(j)}|}{v_{\min}}.$$

Passing to Euclidean norm gives

$$\|\nabla\Phi(p) - \nabla\Phi(p')\| \leq \frac{1}{v_{\min}} \|p - p'\| \leq \frac{1}{v_{\min}} \|p - p'\|_1 = L_{\nabla\Phi} \|p - p'\|_1,$$

which proves Eq. (100). □

One-initial-state replacement constants. Fix the algorithmic parameters $I \geq 1$, $\xi \in [0, 1)$, $\rho > 0$, and $\eta = (1 - \xi)/\rho$. For the replacement-stability lemma, use the previously defined constants ℓ_V, L_w, $v_{\min}$, $\ell_{\nabla V}$, $B_{\nabla\Phi}$, and L_{xu}. Define

$$b_j := b_{\nabla f} + b_{\nabla g} b_u, \qquad B_\nu := \ell_V B_{\nabla\Phi} \exp(b_j T), \tag{102}$$

and

$$\ell_{j,x} := \ell_{\nabla f} + \ell_{\nabla g} b_u, \qquad \ell_{j,u} := b_{\nabla g}. \tag{103}$$

The costate-deviation constants are

$$\begin{aligned} C_{\nu,d} &:= \exp(b_j T) \left[\ell_{\nabla V} L_{xu} B_{\nabla\Phi} + B_\nu \left(\ell_{j,x} L_{xu} T + \ell_{j,u} \sqrt{T} \right) \right], \\ C_{\nu,\pi} &:= \exp(b_j T) \frac{\ell_V}{v_{\min}}, \end{aligned} \tag{104}$$

and the update-deviation constants are

$$A := \sqrt{T} \left(\ell_g L_{xu} B_\nu + b_g C_{\nu,d} \right), \qquad B_\star := \sqrt{T}\, b_g C_{\nu,\pi}. \tag{105}$$

Set

$$\kappa := \xi + \frac{\eta(A + B_\star L_w)}{M}, \tag{106}$$

and

$$\bar{\kappa} := \xi + \eta(A + B_\star L_w), \qquad M_0 := \frac{2\eta(A + B_\star L_w)}{1-\xi},$$
$$\Theta_I := \begin{cases} \dfrac{2}{1-\xi}, & M \geq M_0, \\ I \max(1, \bar{\kappa})^{I-1}, & \text{arbitrary } M \text{ at fixed } I. \end{cases} \tag{107}$$

Finally, define

$$c_M := \frac{\eta B_\star L_w \Theta_I}{M}, \qquad c_{\mathrm{rep}} := 2(1 + c_M), \qquad c_{\mathrm{rep}}^{(2)} := \sqrt{2} + 2c_M. \tag{108}$$

Lemma C.22 (One-initial-state replacement algorithmic stability)**.** *Assume* **Assumption C.12**–**Assumption C.15***, and the projected algorithm convention. Run the deterministic, zero-initialized, projected variant of* ***Algorithm 1*** *for I iterations. Let $F : (\mathbb{R}^n)^M \to \Delta_{o-1}$ be the optimize-then-generate map $F(\mathbf{X}_{\mathrm{init}}) = \hat{p}$ returned by this algorithm after I iterations. If two initial-state batches $\mathbf{X}_{\mathrm{init}} = (\mathbf{x}_0^{[1]}, \ldots, \mathbf{x}_0^{[M]})$ and $\widetilde{\mathbf{X}}_{\mathrm{init}} = (\widetilde{\mathbf{x}}_0^{[1]}, \ldots, \widetilde{\mathbf{x}}_0^{[M]})$ differ in exactly one sample i by arbitrary replacement, mark the second run with tildes and set, for $j \neq i$,*

$$d_j^{(r)} := \|\mathbf{u}^{[j],(r)} - \widetilde{\mathbf{u}}^{[j],(r)}\|_h, \qquad S_r := \sum_{j \neq i} d_j^{(r)}, \qquad \pi_r := \|\hat{p}^{(r)} - \widetilde{\hat{p}}^{(r)}\|_1.$$

Then the following hold:

(i) For every $0 \leq r \leq I$,

$$\pi_r \leq \frac{2}{M} + \frac{L_w}{M} S_r. \tag{109}$$

Moreover,

$$S_r \leq \frac{2\eta B_\star \Theta_I}{M}. \tag{110}$$

(ii) With the constants from Eq. (108), the returned soft mean has the one-initial-state bounded differences

$$\|F(\mathbf{X}_{\mathrm{init}}) - F(\widetilde{\mathbf{X}}_{\mathrm{init}})\|_1 \leq \frac{c_{\mathrm{rep}}}{M}, \qquad \|F(\mathbf{X}_{\mathrm{init}}) - F(\widetilde{\mathbf{X}}_{\mathrm{init}})\| \leq \frac{c_{\mathrm{rep}}^{(2)}}{M}. \tag{111}$$

Proof. **Part (i): aggregate stability recursion.** Let the two runs be as in the statement. For update rounds $r = 0, \ldots, I$, the index $r = I$ denotes the final forward pass. Zero initialization gives $d_j^{(0)} = 0$ and hence $S_0 = 0$. All controls used below satisfy the pointwise bound $\|\mathbf{u}_k\| \leq b_u$: after projection this is **Assumption C.14**, while at $r = 0$ it follows from zero initialization.

Step 1: flow Lipschitz state deviation. For $j \neq i$, the two trajectories start from the same initial state and differ only through their controls. By the uniform-in-k form of Lemma C.18,

$$\max_{0 \leq k \leq K} \|\mathbf{x}_k^{[j],(r)} - \widetilde{\mathbf{x}}_k^{[j],(r)}\| \leq L_{xu} d_j^{(r)}. \tag{112}$$

Step 2: soft-mean split. Split the empirical mean into the replaced sample i and the rest. The replaced sample contributes at most the ℓ_1 diameter 2 of the simplex, and Lemma C.19 with Eq. (112) gives, for $j \neq i$,

$$\|V(\mathbf{x}_K^{[j],(r)}) - V(\widetilde{\mathbf{x}}_K^{[j],(r)})\|_1 \leq \sqrt{o}\, \ell_V L_{xu} d_j^{(r)} = L_w d_j^{(r)}.$$

Therefore

$$\pi_r \leq \frac{2}{M} + \frac{L_w}{M} S_r. \tag{113}$$

This proves Eq. (109).

Step 3: terminal costate condition and magnitude. For terminal states $X_K^{(r)} = (\mathbf{x}_K^{[1],(r)}, \dots, \mathbf{x}_K^{[M],(r)})$, write $\Phi^\sharp(X_K^{(r)}) := \Phi(\hat{p}^{(r)})$. Since

$$\frac{\partial \hat{p}^{(r)}}{\partial \mathbf{x}_K^{[j],(r)}} = \frac{1}{M} \nabla V(\mathbf{x}_K^{[j],(r)}),$$

the chain rule and the floor from Lemma C.19 give

$$\nu_K^{[j],(r)} = \frac{1}{M} \nabla V(\mathbf{x}_K^{[j],(r)})^\top \nabla \Phi(\hat{p}^{(r)}). \tag{114}$$

Thus Lemma C.21 implies

$$\|\nu_K^{[j],(r)}\| \le \frac{\ell_V B_{\nabla\Phi}}{M}.$$

Use b_j and B_ν from Eq. (102). That is,

$$b_j := b_{\nabla f} + b_{\nabla g} b_u, \qquad B_\nu := \ell_V B_{\nabla\Phi} \exp(b_j T). \tag{102}$$

Let

$$J_k^{[j],(r)} := \nabla_{\mathbf{x}} \left[\mathbf{f}^\theta(\mathbf{x}, t_k) + \mathbf{g}(\mathbf{x}, t_k) \mathbf{u}_k^{[j],(r)} \right]_{\mathbf{x}=\mathbf{x}_k^{[j],(r)}}.$$

The backward recursion $\nu_k^{[j],(r)} = \nu_{k+1}^{[j],(r)} + h_k J_k^{[j],(r)\top} \nu_{k+1}^{[j],(r)}$ has $\|J_k^{[j],(r)}\|_{\mathrm{op}} \le b_j$ by **Assumption C.15**. The backward form of Lemma C.16 yields, for every sample, time, iteration, and run,

$$\max_{0 \le k \le K} \|\nu_k^{[j],(r)}\| \le \frac{\ell_V B_{\nabla\Phi} \exp(b_j T)}{M} = \frac{B_\nu}{M}. \tag{115}$$

Step 4: costate deviation. Fix $j \neq i$ and $r \le I-1$. Subtracting Eq. (114) for the two runs and adding the mixed term gives

$$\|\nu_K^{[j],(r)} - \widetilde{\nu}_K^{[j],(r)}\| \le \frac{1}{M} \left[\ell_{\nabla V} L_{xu} B_{\nabla\Phi} d_j^{(r)} + \frac{\ell_V}{v_{\min}} \pi_r \right],$$

where Eq. (112) and Lemma C.21 are used. For propagation, write $\Delta\nu_k := \nu_k^{[j],(r)} - \widetilde{\nu}_k^{[j],(r)}$, $\Delta\mathbf{x}_k := \mathbf{x}_k^{[j],(r)} - \widetilde{\mathbf{x}}_k^{[j],(r)}$, $\Delta\mathbf{u}_k := \mathbf{u}_k^{[j],(r)} - \widetilde{\mathbf{u}}_k^{[j],(r)}$, and $\Delta J_k := J_k^{[j],(r)} - \widetilde{J}_k^{[j],(r)}$. Use $\ell_{j,x}$ and $\ell_{j,u}$ from Eq. (103). That is,

$$\ell_{j,x} := \ell_{\nabla f} + \ell_{\nabla g} b_u, \qquad \ell_{j,u} := b_{\nabla g}. \tag{103}$$

The Jacobian regularity gives

$$\|\Delta J_k\|_{\mathrm{op}} \le \ell_{j,x} \|\Delta\mathbf{x}_k\| + \ell_{j,u} \|\Delta\mathbf{u}_k\|.$$

Using Eq. (115), the backward recursion satisfies

$$\|\Delta\nu_k\| \le (1 + h_k b_j) \|\Delta\nu_{k+1}\| + h_k \|\Delta J_k\|_{\mathrm{op}} \frac{B_\nu}{M}.$$

The time-weighted sum is bounded by

$$\sum_{k=0}^{K-1} h_k \|\Delta J_k\|_{\mathrm{op}} \le \left(\ell_{j,x} L_{xu} T + \ell_{j,u} \sqrt{T} \right) d_j^{(r)},$$

where the first term uses Eq. (112) and the second uses Cauchy–Schwarz in the h-norm. Another backward Gronwall step and the definitions in Eq. (104) give

$$\begin{aligned} C_{\nu,d} &:= \exp(b_j T) \left[\ell_{\nabla V} L_{xu} B_{\nabla\Phi} + B_\nu \left(\ell_{j,x} L_{xu} T + \ell_{j,u} \sqrt{T} \right) \right], \\ C_{\nu,\pi} &:= \exp(b_j T) \frac{\ell_V}{v_{\min}}. \end{aligned} \tag{104}$$

Therefore

$$\max_{0 \le k \le K} \|\nu_k^{[j],(r)} - \widetilde{\nu}_k^{[j],(r)}\| \le \frac{C_{\nu,d} d_j^{(r)} + C_{\nu,\pi} \pi_r}{M}. \tag{116}$$

Step 5: projection nonexpansive update. The update is

$$\mathbf{u}_k^{[j],(r+1)} = \Pi_{\mathcal{U}}\left(\xi \mathbf{u}_k^{[j],(r)} - \eta \mathbf{w}_k^{[j],(r)}\right), \qquad \mathbf{w}_k^{[j],(r)} := \mathbf{g}(\mathbf{x}_k^{[j],(r)}, t_k)^\top \nu_{k+1}^{[j],(r)}.$$

The blockwise projection is nonexpansive in the h-norm, so

$$d_j^{(r+1)} \le \xi d_j^{(r)} + \eta \|\mathbf{w}^{[j],(r)} - \widetilde{\mathbf{w}}^{[j],(r)}\|_h.$$

Splitting $\mathbf{g}^\top \nu - \widetilde{\mathbf{g}}^\top \widetilde{\nu} = (\mathbf{g} - \widetilde{\mathbf{g}})^\top \nu + \widetilde{\mathbf{g}}^\top (\nu - \widetilde{\nu})$ and using Eq. (112), Eq. (115), and Eq. (116),

$$\|\mathbf{w}_k^{[j],(r)} - \widetilde{\mathbf{w}}_k^{[j],(r)}\| \le \frac{1}{M}\left[(\ell_g L_{xu} B_\nu + b_g C_{\nu,d})\, d_j^{(r)} + b_g C_{\nu,\pi} \pi_r\right]$$

uniformly in k. Hence with A and $B_\star$ from Eq. (105),

$$A := \sqrt{T}\,(\ell_g L_{xu} B_\nu + b_g C_{\nu,d}), \qquad B_\star := \sqrt{T}\, b_g C_{\nu,\pi}. \tag{105}$$

we obtain

$$\|\mathbf{w}^{[j],(r)} - \widetilde{\mathbf{w}}^{[j],(r)}\|_h \le \sqrt{T} \max_k \|\mathbf{w}_k^{[j],(r)} - \widetilde{\mathbf{w}}_k^{[j],(r)}\| \le \frac{1}{M}\left(A d_j^{(r)} + B_\star \pi_r\right).$$

Consequently,

$$d_j^{(r+1)} \le \left(\xi + \frac{\eta A}{M}\right) d_j^{(r)} + \frac{\eta B_\star}{M} \pi_r. \tag{117}$$

Step 6: recursion and geometric sum. Use κ from Eq. (106).

$$\kappa := \xi + \frac{\eta(A + B_\star L_w)}{M}. \tag{106}$$

Summing Eq. (117) over $j \neq i$ and inserting Eq. (113),

$$S_{r+1} \le \kappa S_r + \frac{2\eta B_\star}{M}. \tag{118}$$

Since $S_0 = 0$,

$$S_r \le \frac{2\eta B_\star}{M} \sum_{s=0}^{r-1} \kappa^s.$$

The geometric constants from Eq. (107) are

$$\begin{aligned} \bar{\kappa} &:= \xi + \eta(A + B_\star L_w), \qquad M_0 := \frac{2\eta(A + B_\star L_w)}{1 - \xi}, \\ \Theta_I &:= \begin{cases} \dfrac{2}{1-\xi}, & M \ge M_0, \\ I \max(1, \bar{\kappa})^{I-1}, & \text{arbitrary } M \text{ at fixed } I. \end{cases} \end{aligned} \tag{107}$$

If $M \ge M_0$, then $\eta(A + B_\star L_w)/M \le (1 - \xi)/2$, hence $\kappa \le (1 + \xi)/2 < 1$, and

$$\begin{aligned} \sum_{s=0}^{I-1} \kappa^s &= \frac{1 - \kappa^I}{1 - \kappa} \\ &\le \frac{1}{1 - \kappa} \qquad (0 < \kappa < 1) \\ &\le \frac{1}{1 - (1+\xi)/2} = \frac{2}{1 - \xi}. \end{aligned}$$

Otherwise for arbitrary M at fixed I, since $\kappa \le \bar{\kappa}$, we have

$$\sum_{s=0}^{I-1} \kappa^s \le \sum_{s=0}^{I-1} \max(1, \bar{\kappa})^{I-1} = I \max(1, \bar{\kappa})^{I-1}.$$

Combining the two cases with Θ_I from Eq. (107),

$$S_r \le \frac{2\eta B_\star \Theta_I}{M}, \qquad 0 \le r \le I. \tag{119}$$

and thus Eq. (110).

Part (ii): bounded differences. Let

$$c_M = \frac{\eta B_\star L_w \Theta_I}{M}, \qquad c_{\mathrm{rep}} = 2(1 + c_M), \qquad c_{\mathrm{rep}}^{(2)} = \sqrt{2} + 2c_M. \qquad \text{(defined in (108))}$$

Apply Eq. (113) at the final forward pass and insert Eq. (119):

$$\left\|\hat{p} - \widetilde{\hat{p}}\right\|_1 = \pi_I \le \frac{2}{M} + \frac{L_w}{M}\frac{2\eta B_\star \Theta_I}{M} = \frac{2(1 + c_M)}{M} = \frac{c_{\mathrm{rep}}}{M}.$$

For the Euclidean bound, write $V_j := V(\mathbf{x}_K^{[j],(I)})$ and $\widetilde{V}_j := V(\widetilde{\mathbf{x}}_K^{[j],(I)})$. Since the two batches differ only at sample i,

$$\hat{p} - \widetilde{\hat{p}} = \frac{1}{M}(V_i - \widetilde{V}_i) + \frac{1}{M}\sum_{j \ne i}(V_j - \widetilde{V}_j).$$

Both V_i and $\widetilde{V}_i$ lie in the simplex, whose Euclidean diameter is $\sqrt{2}$, so $\|V_i - \widetilde{V}_i\| \le \sqrt{2}$. For each $j \ne i$, the same soft-attribute estimate used in Step 2 gives

$$\|V_j - \widetilde{V}_j\| \le \|V_j - \widetilde{V}_j\|_1 \le L_w d_j^{(I)}.$$

Therefore the triangle inequality and Eq. (119) give

$$\left\|\hat{p} - \widetilde{\hat{p}}\right\| \le \frac{\sqrt{2}}{M} + \frac{L_w}{M} S_I \le \frac{\sqrt{2} + 2c_M}{M} = \frac{c_{\mathrm{rep}}^{(2)}}{M}.$$

This proves Eq. (111). □

Lemma C.23 (Permutation equivariance and output mean identity)**.** *Assume the projected algorithm convention. Run the deterministic, zero-initialized, projected variant of **Algorithm 1** for I iterations. The algorithm is permutation equivariant: permuting the input batch permutes the returned controls and terminal states and leaves $\hat{p}$ unchanged. Consequently, for i.i.d. initial states the returned terminal states are exchangeable and*

$$\mathbb{E}[\hat{p}] = \mathbb{E}[V(\mathbf{x}_K^{[1]})] = \bar{p}_M. \tag{120}$$

Proof. Let σ be a permutation of $\{1, \dots, M\}$ and run the algorithm on $(\mathbf{x}_0^{[\sigma(1)]}, \dots, \mathbf{x}_0^{[\sigma(M)]})$. We prove by induction on r that

$$\mathbf{u}^{\sigma,[j],(r)} = \mathbf{u}^{[\sigma(j)],(r)}, \qquad \mathbf{x}_k^{\sigma,[j],(r)} = \mathbf{x}_k^{[\sigma(j)],(r)}, \qquad \hat{p}^{\sigma,(r)} = \hat{p}^{(r)}.$$

At $r = 0$ the controls are all zero. If the control identity holds at round r, then the forward Euler recursion is applied blockwise with the same initial state and control, so the state identity holds for all k. The soft mean is symmetric in the blocks, hence unchanged by the permutation. The terminal costate formula uses the same ∇V at the permuted block and the same $\nabla\Phi(\hat{p}^{(r)})$; the backward recursion is blockwise, so the costates permute in the same way. Finally, the projected update is also blockwise and single-valued, giving the control identity at round $r + 1$. Thus the returned terminal states are permuted and $\hat{p}$ is unchanged. For i.i.d. initial states, the input batch has the same law as any permutation of itself, so the returned terminal states are exchangeable. Since V is bounded,

$$\mathbb{E}[\hat{p}] = \frac{1}{M}\sum_{j=1}^{M} \mathbb{E}[V(\mathbf{x}_K^{[j]})] = \mathbb{E}[V(\mathbf{x}_K^{[1]})] = \bar{p}_M.$$

□

Lemma C.24 (Bounded-difference concentration for simplex-valued maps)**.** *Let $\mathbf{X} = (X_1, \dots, X_M)$ have independent samples in $\mathbb{R}^n$ and let $F : (\mathbb{R}^n)^M \to \Delta_{o-1}$ be measurable. Assume that there are constants c_1, c_2 such that every two deterministic batches $\mathbf{x}, \widetilde{\mathbf{x}} \in (\mathbb{R}^n)^M$ differing in one sample satisfy*

$$\|F(\mathbf{x}) - F(\widetilde{\mathbf{x}})\|_1 \le \frac{c_1}{M}, \qquad \|F(\mathbf{x}) - F(\widetilde{\mathbf{x}})\| \le \frac{c_2}{M}.$$

Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\|F(\mathbf{X}) - \mathbb{E}F(\mathbf{X})\|_1 \le c_2\sqrt{\frac{o}{2M}} + c_1\sqrt{\frac{\log(1/\delta)}{2M}}. \tag{121}$$

Proof. Write $Y := F(\mathbf{X})$ and $\mu := \mathbb{E}Y$. For each i, let $\mathbf{X}^{(i)}$ be the batch obtained by resampling only X_i independently from its own law. Then $\mathbf{X}$ and $\mathbf{X}^{(i)}$ differ only in sample i, so the bounded-difference assumption gives $\|F(\mathbf{X}) - F(\mathbf{X}^{(i)})\| \le c_2/M$ and $\|F(\mathbf{X}) - F(\mathbf{X}^{(i)})\|_1 \le c_1/M$.

Apply Efron–Stein inequality (Boucheron et al., 2013, Thm. 3.1) coordinatewise to $Y^{(l)}$, $l = 1, \dots, o$:

$$\mathrm{Var}(Y^{(l)}) \le \frac{1}{2}\sum_{i=1}^{M} \mathbb{E}\left[\left(Y^{(l)} - F(\mathbf{X}^{(i)})^{(l)}\right)^2\right].$$

Summing over coordinates and using the Euclidean replacement bound gives

$$\mathbb{E}\|Y - \mu\|^2 = \sum_{l=1}^{o} \mathrm{Var}(Y^{(l)}) \le \frac{1}{2}\sum_{i=1}^{M} \mathbb{E}\|F(\mathbf{X}) - F(\mathbf{X}^{(i)})\|^2 \le \frac{c_2^2}{2M}.$$

Therefore,

$$\begin{aligned}
\mathbb{E}\|Y - \mu\|_1 &= \mathbb{E}\sum_{l=1}^{o} |Y^{(l)} - \mu^{(l)}| \\
&\le \mathbb{E}\left[\sqrt{o} \cdot \sqrt{\sum_{l=1}^{o} |Y^{(l)} - \mu^{(l)}|^2}\right] && \text{(Cauchy–Schwarz)} \\
&= \sqrt{o}\,\mathbb{E}\|Y - \mu\| \\
&\le \sqrt{o}\,\sqrt{\mathbb{E}\|Y - \mu\|^2} && \text{(Jensen)} \\
&\le c_2\sqrt{\frac{o}{2M}}. && (122)
\end{aligned}$$

Now set $Z := \|F(\mathbf{X}) - \mu\|_1$. If two inputs differ only in sample i, the reverse triangle inequality and the ℓ_1 replacement bound give

$$|Z(\mathbf{x}) - Z(\widetilde{\mathbf{x}})| \le \|F(\mathbf{x}) - F(\widetilde{\mathbf{x}})\|_1 \le \frac{c_1}{M}.$$

Applying the McDiarmid's inequality (bounded difference inequality) (McDiarmid, 1989, Lem. 1.2)(Boucheron et al., 2013, Thm. 6.2) yields, for every $t > 0$,

$$\mathbb{P}\{Z \ge \mathbb{E}Z + t\} \le \exp\left(-\frac{2t^2}{\sum_{i=1}^{M}\left(\frac{c_1}{M}\right)^2}\right) = \exp\left(-\frac{2Mt^2}{c_1^2}\right). \tag{123}$$

Taking $t = c_1\sqrt{\frac{\log(1/\delta)}{2M}}$ into Eq. (123) yields

$$\begin{aligned}
\mathbb{P}\left\{Z \ge \mathbb{E}Z + c_1\sqrt{\frac{\log(1/\delta)}{2M}}\right\} &= \mathbb{P}\left\{\|F(\mathbf{X}) - \mu\|_1 \ge \mathbb{E}\|F(\mathbf{X}) - \mu\|_1 + c_1\sqrt{\frac{\log(1/\delta)}{2M}}\right\} \\
&\le \delta.
\end{aligned}$$

Inserting the expectation bound Eq. (122), which is a deterministic event, gives:

$$\mathbb{P}\left\{\|F(\mathbf{X}) - \mu\|_1 \ge c_2\sqrt{\frac{o}{2M}} + c_1\sqrt{\frac{\log(1/\delta)}{2M}}\right\} \le \delta$$

and this proves Eq. (121). □

C.3.4 STATEMENT

For the subsequent concentration step, define

$$\varepsilon_{\mathrm{rep}}(M,\delta) := c_{\mathrm{rep}}^{(2)}\sqrt{\frac{o}{2M}} + c_{\mathrm{rep}}\sqrt{\frac{\log(1/\delta)}{2M}}. \tag{124}$$

Proposition C.25 (restatement of **Proposition 3.3**). *Assume* **Assumption C.12–Assumption C.15**. *Let the initial states be i.i.d.* $\mathcal{N}(\mathbf{0},\, \boldsymbol{I}_n)$*, and run* ***Algorithm 1*** *with control projection and for a fixed* I*. Let* $\bar{p}_M$ *be defined by Eq. (72) and let* $\hat{\varepsilon} = \mathbb{D}_{KL}\left[\hat{p} \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right]$*. Then, for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\|\hat{p} - \bar{p}_M\|_1 \le \varepsilon_{\mathrm{rep}}(M,\delta), \tag{125}$$

$$\mathrm{TV}(\bar{p}_M, p_{\mathbf{y}}^{\mathrm{tar}}) \le \sqrt{\frac{\hat{\varepsilon}}{2}} + \frac{1}{2}\varepsilon_{\mathrm{rep}}(M,\delta), \tag{126}$$

$$\mathbb{D}_{KL}\left[\bar{p}_M \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right] \le \hat{\varepsilon} + L_\Phi(v_{\min})\varepsilon_{\mathrm{rep}}(M,\delta), \tag{127}$$

where

$$\varepsilon_{\mathrm{rep}}(M,\delta) = (1+\sqrt{2}\,c_M)\sqrt{\frac{o}{M}} + (1+c_M)\sqrt{\frac{2\log(1/\delta)}{M}}. \tag{128}$$

with c_M *as in Eq. (108). If* $M \ge M_0$ *(defined in Eq. (107)), or if the iteration budget* I *is fixed, the constant* c_M *is* $O(1/M)$*.*

C.3.5 PROOF OF **Proposition C.25**

Proof. Let $\mathbf{X}_{\mathrm{init}} := (\mathbf{x}_0^{[1]}, \dots, \mathbf{x}_0^{[M]})$ and define $F(\mathbf{X}_{\mathrm{init}}) = \hat{p}$ to be the optimize-then-generate map returned by **Algorithm 1** with a fixed iteration budget I and control projection. Lemma C.22 supplies the arbitrary one-sample replacement bounds with $c_1 = c_{\mathrm{rep}}$ and $c_2 = c_{\mathrm{rep}}^{(2)}$. Lemma C.23 identifies $\mathbb{E}F(\mathbf{X}_{\mathrm{init}}) = \bar{p}_M$. Invoking Lemma C.24 gives, with probability at least $1-\delta$,

$$\|F(\mathbf{X}_{\mathrm{init}}) - \mathbb{E}F(\mathbf{X}_{\mathrm{init}})\|_1 \le c_{\mathrm{rep}}^{(2)}\sqrt{\frac{o}{2M}} + c_{\mathrm{rep}}\sqrt{\frac{\log(1/\delta)}{2M}} = \varepsilon_{\mathrm{rep}}(M,\delta).$$

This proves Eq. (125); let $\mathcal{E}$ denote this event.

TV alignment. On event $\mathcal{E}$,

$$\begin{aligned}\mathrm{TV}(\bar{p}_M, p_{\mathbf{y}}^{\mathrm{tar}}) &\le \frac{1}{2}\|\bar{p}_M - \hat{p}\|_1 + \mathrm{TV}(\hat{p}, p_{\mathbf{y}}^{\mathrm{tar}}) && \text{(triangle ineq.)}\\ &\le \frac{1}{2}\varepsilon_{\mathrm{rep}}(M,\delta) + \sqrt{\frac{\hat{\varepsilon}}{2}}, && \text{(Pinsker's ineq.)}\end{aligned}$$

and this proves Eq. (126).

KL alignment. By Lemma C.19, both $\hat{p}$ and $\bar{p}_M$ lie in $\Delta_{\ge v_{\min}}$. Applying Lemma C.20, equivalently Eq. (96), with $v = v_{\min}$,

$$\begin{aligned}\mathbb{D}_{\mathrm{KL}}\left[\bar{p}_M \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right] &\le \mathbb{D}_{\mathrm{KL}}\left[\hat{p} \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right] + L_\Phi(v_{\min})\|\bar{p}_M - \hat{p}\|_1 && \text{(KL-Lipschitz)}\\ &\le \hat{\varepsilon} + L_\Phi(v_{\min})\varepsilon_{\mathrm{rep}}(M,\delta). && \text{(concentration Eq. (125))}\end{aligned}$$

Radius algebra. Using Eq. (108) in Eq. (124),

$$\begin{aligned}\varepsilon_{\mathrm{rep}}(M,\delta) &= (\sqrt{2}+2c_M)\sqrt{\frac{o}{2M}} + 2(1+c_M)\sqrt{\frac{\log(1/\delta)}{2M}}\\ &= (1+\sqrt{2}\,c_M)\sqrt{\frac{o}{M}} + (1+c_M)\sqrt{\frac{2\log(1/\delta)}{M}},\end{aligned} \tag{129}$$

and this proves Eq. (128).

The two cases in Eq. (107) make Θ_I independent of M, so $c_M = O(1/M)$ in both cases. □

Remark 7 (Status of the population result). In **Proposition C.25**, the left-hand sides involving $\bar{p}_M$ are deterministic functions of the algorithm and initial-state law. The terminal cost $\hat{\varepsilon}$ is observed from the returned batch. The radius $\varepsilon_{\mathrm{rep}}(M, \delta)$ is computable only given upper bounds on the model and algorithm constants summarized in Table 5; it is not a batch-only quantity. The auxiliary batch-only bound below is **Proposition C.26**.

C.3.6 Deterministic Batch Attribute Alignment Bound

Scope. Complementing the population bound above, this result gives a batch-only bound for the returned samples, converting the realized soft KL cost into a hard-label histogram guarantee under a margin term. The next proposition is a deterministic statement about a realized finite batch. It uses no random initial-state assumption, no dynamics regularity assumption, no optimality assumption, and no admissibility assumption on the controls. The full-support target condition only prevents the realized KL term from being vacuous.

Proposition C.26 (Deterministic batch-level alignment bound)**.** *Let $\mathbf{x}_K^{[1]}, \dots, \mathbf{x}_K^{[M]}$ be any realized terminal states generated with sample-wise controls. Assume $p_{\mathbf{y}}^{\mathrm{tar}}$ has full support, and define $\hat{p}$, $\hat{p}_{\mathrm{hard}}$, $\hat{\varepsilon}$, and $\hat{\beta}_\alpha$ by Eq. (68)–Eq. (71). Then, for every $\alpha > 0$,*

$$\mathrm{TV}\big(\hat{p}_{\mathrm{hard}}, p_{\mathbf{y}}^{\mathrm{tar}}\big) \le \sqrt{\frac{\hat{\varepsilon}}{2}} + (o-1)e^{-\alpha/\tau} + \hat{\beta}_\alpha. \tag{130}$$

All terms on the right-hand side are computable from the returned batch.

Proof. Fix $\alpha > 0$.

Per definition of TV and the triangle inequality:

$$\mathrm{TV}(\hat{p}_{\mathrm{hard}}, p_{\mathbf{y}}^{\mathrm{tar}}) \le \mathrm{TV}(\hat{p}_{\mathrm{hard}}, \hat{p}) + \mathrm{TV}(\hat{p}, p_{\mathbf{y}}^{\mathrm{tar}}). \tag{131}$$

By Pinsker's inequality applied with first argument $\hat{p}$,

$$\mathrm{TV}(\hat{p}, p_{\mathbf{y}}^{\mathrm{tar}}) \le \sqrt{\frac{1}{2}\,\mathbb{D}_{\mathrm{KL}}\left[\hat{p} \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right]} = \sqrt{\frac{\hat{\varepsilon}}{2}}. \tag{132}$$

By Eq. (68) and the triangle inequality,

$$\mathrm{TV}(\hat{p}_{\mathrm{hard}}, \hat{p}) = \frac{1}{2}\left\| \frac{1}{M}\sum_{i=1}^{M} (\mathbf{e}_{\hat{y}_i} - V(\mathbf{x}_K^{[i]})) \right\|_1 \le \frac{1}{2M}\sum_{i=1}^{M} \|\mathbf{e}_{\hat{y}_i} - V(\mathbf{x}_K^{[i]})\|_1. \tag{133}$$

Split the summands on the RHS of Eq. (133) by the logits margin: Let $G_\alpha = \{i : \gamma_i \ge \alpha\}$ and $B_\alpha = \{i : \gamma_i < \alpha\}$, so $|B_\alpha| = M\hat{\beta}_\alpha$.

- For $i \in B_\alpha$, the simplex diameter gives

$$\|\mathbf{e}_{\hat{y}_i} - V(\mathbf{x}_K^{[i]})\|_1 \le 2. \tag{134}$$

- For $i \in G_\alpha$, further split *each* summand by

$$\begin{aligned}
\|\mathbf{e}_{\hat{y}_i} - V(\mathbf{x}_K^{[i]})\|_1 &= \left|1 - V^{(\hat{y}_i)}\left(\mathbf{x}_K^{[i]}\right)\right| + \sum_{j \ne \hat{y}_i}^{o} \left|0 - V^{(j)}\left(\mathbf{x}_K^{[i]}\right)\right| && \text{(def. } L_1 \text{ norm)} \\
&= 1 - V^{(\hat{y}_i)} + \sum_{j \ne \hat{y}_i}^{o} V^{(j)} && (V^{(\cdot)} \in (0,1]) \\
&= 2\sum_{j \ne \hat{y}_i}^{o} V^{(j)}. && (\textstyle\sum_{j=1}^{o} V^{(j)} = 1)
\end{aligned}$$

  For each $j \ne \hat{y}_i$, the condition $i \in G_\alpha$ bounds the following ratio:

$$\frac{V^{(j)}}{V^{(\hat{y}_i)}} = \exp\left(\frac{\Psi^{(j)} - \Psi^{(\hat{y}_i)}}{\tau}\right) \le e^{-\frac{\alpha}{\tau}},$$

  where the inequality uses $\Psi^{(\hat{y}_i)} - \Psi^{(j)} \ge \alpha$. This further yields

$$V^{(j)} \le e^{-\frac{\alpha}{\tau}}.$$

  since $V^{(\hat{y}_i)} \in (0,1]$. Thus,

$$\|\mathbf{e}_{\hat{y}_i} - V(\mathbf{x}_K^{[i]})\|_1 \le 2\sum_{j \ne \hat{y}_i}^{o} e^{-\frac{\alpha}{\tau}} = 2(o-1)e^{-\frac{\alpha}{\tau}}. \tag{136}$$

Substituting the two bounds Eq. (134) and Eq. (136) into Eq. (133) yields

$$\mathrm{TV}(\hat{p}_{\mathrm{hard}}, \hat{p}) \leq (1 - \hat{\beta}_\alpha)(o-1)e^{-\alpha/\tau} + \hat{\beta}_\alpha \leq (o-1)e^{-\alpha/\tau} + \hat{\beta}_\alpha. \tag{137}$$

Combining Eq. (131), Eq. (132), and Eq. (137) proves Eq. (130). □

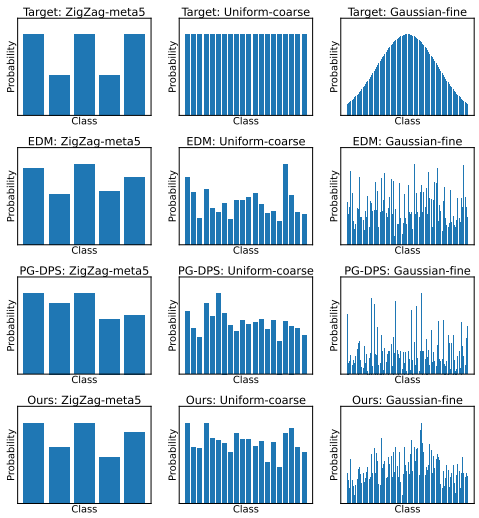
D EXPERIMENT DETAILS

Software and Codebase All experiments were run using PyTorch Paszke et al. (2019). For the CIFAR experiment, our implementation is built upon Karras et al. (2022) and the implementation of the image classifier used as the attribute model therein is from Chen (2025). For the human face generation experiment, diffusion-based implementations are adapted from Choi et al. (2022), and flow-based implementations are built upon Dao et al. (2023). The attribute model implementation is from Luan et al. (2026).

Computational Hardware The training of the unconditional EDM model in the CIFAR experiment and *all evaluation*, including computing all metrics, were run on a workstation with 1 AMD Ryzen Threadripper PRO 5995WX 64-Core CPU, 504 GB RAM, and 2 NVIDIA RTX A6000 GPUs each with 48GB VRAM. *Inference* of all methods was run on a high-performance-computing (HPC) cluster with NVIDIA H200 GPUs. For each experiment run, only 1 GPU was utilized.

D.1 CIFAR EXPERIMENT

Hierarchical attribute distributions We construct three different levels of class labels with a coarse-to-fine hierarchy: `meta5`, `coarse`, and `fine`, with numbers of classes of 5, 20, and 100, respectively. The `coarse` and `fine` levels of class labels are native in CIFAR-100. The `meta5` level is obtained by merging every 4 `coarse` classes. For each class level, we choose three different attribute distributions as test-time targets : Uniform, ZigZag, and Gaussian. ZigZag has an alternating "high-low" pattern where every even-numbered class is exactly twice as likely to be selected as any odd-numbered class. For Gaussian, the target resembles a smooth bell curve centered in the middle-numbered class, with the standard deviation as $1/4$ of the support size.



(a) ZigZag-`meta5` (b) Uniform-`coarse` (c) Gaussian-`fine`

Figure 3: **Top row**: Test-time *target* attribute distributions (CIFAR-100). **2nd/3rd rows:** Generated distributions of baselines. **Bottom row:** Generated distributions of our method.

Base diffusion model and attribute models We trained a base EDM model on CIFAR-100 dataset Krizhevsky et al. (2009) with the default training split and the network backbone is the UNet in Song & Ermon (2019) implemented by Karras et al. (2022). The EDM training followed default hyperparameters disclosed in Karras et al. (2022). All attribute models are ResNet56 trained on CIFAR-100 training set, and we adopt the implementations and training hyperparameters in Chen (2025).

Diffusion inference and evaluation For diffusion inference for all methods, we adopt the default settings provided by Karras et al. (2022) with $K = 18$ sampling steps, but first-order Euler discretization instead of second-order Heun. For each method, we sample 10240 images and evaluate all metrics based on the empirical distribution of attributes.

Particle Guidance implementation We implement the Particle Guidance (PG) (Corso et al., 2024) method by taking the same cost function used in our method as the potential field for PG. Concretely, PG operating on the PF-ODE is as follows:

$$\frac{\mathrm{d}\mathbf{x}_t^{[i]}}{\mathrm{d}t} = -f(\mathbf{x}_t^{[i]}, t) + \frac{1}{2} g(t)^2 \left(\mathbf{s}^\theta \left(\mathbf{x}_t^{[i]}, t \right) + \nabla_{\mathbf{x}_t^{[i]}} \log \mathcal{L} \left(\mathbf{x}_t^{[1]}, \ldots, \mathbf{x}_t^{[M]} \right) \right), \tag{138}$$

where $\log \mathcal{L}$ is the potential field defined over M samples. We set

$$\log \mathcal{L}\left(\mathbf{x}_t^{[1]}, \ldots, \mathbf{x}_t^{[M]}\right) := -\lambda \mathbb{D}_{\mathrm{KL}}\left[\hat{p}_{\mathbf{y}}^t \,\|\, p_{\mathbf{y}}^{\mathrm{tar}}\right], \qquad \hat{p}_{\mathbf{y}}^t := \frac{1}{M}\sum_{i=1}^{M} \mathrm{softmax}\left(\Psi\left(\hat{\mathbf{x}}_T^{[i]}(\mathbf{x}_t^{[i]})\right)\right) \tag{139}$$

where $\lambda > 0$ is a hyperparameter, and $\hat{\mathbf{x}}_T^{[i]}(\mathbf{x}_t^{[i]})$ is a "predicted clean sample" given a noisy sample at time t, obtained via Tweedie's formula Efron (2011). Under EDM (Karras et al., 2022) with a learned score model $\mathbf{s}^\theta$, it takes the form of

$$\hat{\mathbf{x}}_T^{[i]}(\mathbf{x}_t^{[i]}) = \mathbf{x}_t^{[i]} + t^2 \mathbf{s}^\theta(\mathbf{x}_t^{[i]}, t). \tag{140}$$

Under DDIM(Song et al., 2021a) with a learned noise prediction model $\boldsymbol{\epsilon}^\theta$, we take the PF-ODE in $\tilde{\mathbf{x}}_t$ (see Eq. (16)) and this term reads

$$\hat{\tilde{\mathbf{x}}}_T^{[i]}(\tilde{\mathbf{x}}_t^{[i]}) = \tilde{\mathbf{x}}_t^{[i]} - \sigma(t)\boldsymbol{\epsilon}^\theta\left(\tilde{\mathbf{x}}_t^{[i]}/\sqrt{1+\sigma(t)^2},\, t\right) \tag{141}$$

where $\tilde{\mathbf{x}}_t$ and $\sigma(t)$ are defined in Eq. (16).

Hyperparameters The hyperparameters used to obtain the results in Table 1 are reported in Table 6 and Table 7.

Table 6: Hyperparameters used for our method in CIFAR.

Target	I	ρ	M	ξ
Uniform-`meta5`	10	0.1	32	0.99
Uniform-`coarse`	10	0.1	64	0.99
Uniform-`fine`	10	0.1	256	0.99
ZigZag-`meta5`	10	0.05	32	0.99
ZigZag-`coarse`	10	0.05	64	0.99
ZigZag-`fine`	10	0.05	256	0.99
Gaussian-`meta5`	10	0.1	64	0.95
Gaussian-`coarse`	10	0.1	128	0.95
Gaussian-`fine`	10	0.1	256	0.95

Table 7: Hyperparameters used for PG-DPS in CIFAR.

Target	w	M
Uniform-`meta5`	4.0	32
Uniform-`coarse`	4.0	64
Uniform-`fine`	4.0	256
ZigZag-`meta5`	4.0	32
ZigZag-`coarse`	4.0	64
ZigZag-`fine`	4.0	256
Gaussian-`meta5`	4.0	64
Gaussian-`coarse`	4.0	128
Gaussian-`fine`	4.0	256

D.2 Human Face Generation Experiment

Implementation details We use the pretrained DDIM weights from Choi et al. (2022) for the base diffusion model, and pretrained LFM weights from (Dao et al., 2023) for the base latent flow model. For all diffusion-based methods, we take $K = 25$ sampling steps for during diffusion inference with FP16 precision, with other inference hyperparameters following the defaults set in Choi et al. (2022). For latent flow models, we take $K = 20$ sampling steps for inference, with other settings default as in LFM (Dao et al., 2023), including using a standard pretrained variational Autoencoder (Kingma & Welling, 2013) from Stable Diffusion (Rombach et al., 2022). We sample 960 images for each method in the single-attribute distribution alignment experiments (Table 3) for evaluation, and sample 1080 images for each method in the joint attribute distribution alignment cases (Table 2).

Attribute model The attribute model is a lightweight ResNet image classifier implemented by Luan et al. (2026). We train the classifier on FFHQ-Aging. Since the FFHQ-Aging dataset does not have a label for attribute `race`, we leverage a pretrained race classifier from Karkkainen & Joo (2021) to label all images in FFHQ-Aging and use them as ground truth.

Hyperparameters The hyperparameters specific to each method used to obtain the results in Tables 2 and 3 are reported in Table 8 Table 9, and Table 10. The selection of hyperparameters is based on each method's best performance in the TV metric.

Table 8: Hyperparameters used for our method (diffusion-based) in human face generation.

Target	I	ρ	M	ξ
Custom1-`age`	12	0.0015	24	0.95
Custom1-`gender`	10	0.00125	20	0.95
Custom1-`race`	10	0.0005	20	0.95
Custom2-`age`	10	0.0010	24	0.95
Custom2-`gender`	10	0.00075	20	0.95
Custom2-`race`	10	0.0010	20	0.95
CustomJoint	10	0.0002	72	0.95
Uniform-`age`	10	0.002	24	0.95
Uniform-`gender`	10	0.001	20	0.95
Uniform-`race`	10	0.0005	20	0.95
UniformJoint	10	0.0002	72	0.95

Table 9: Hyperparameters used for our method (flow-based) in human face generation.

Target	I	ρ	M	ξ
Custom1-`age`	10	0.00075	24	0.99
Custom1-`gender`	10	0.001	20	0.99
Custom1-`race`	10	0.0005	20	0.99
Custom2-`age`	10	0.00075	24	0.99
Custom2-`gender`	10	0.001	20	0.99
Custom2-`race`	10	0.0005	20	0.99
CustomJoint	10	0.0001	72	0.99
Uniform-`age`	10	0.001	24	0.99
Uniform-`gender`	10	0.00125	20	0.99
Uniform-`race`	10	0.0005	20	0.99
UniformJoint	10	0.00025	72	0.99

Table 10: Hyperparameters used for PG-DPS in human face generation.

Target	w	M
Custom1-`age`	10	24
Custom1-`gender`	50	20
Custom1-`race`	50	20
Custom2-`age`	10	24
Custom2-`gender`	50	20
Custom2-`race`	100	20
CustomJoint	4	72
Uniform-`age`	15	24
Uniform-`gender`	10	20
Uniform-`race`	10	20
UniformJoint	7	72

E Additional Results

E.1 Additional Results for CIFAR

E.1.1 Ablations

Batch size M We perform ablation study in the batch size M used for empirical distribution estimation. We set $M \in \{8, 16, 32, 64, 128, 256\}$ and test all methods with all of the three types of target distributions (Uniform, Guassian, ZigZag) with all three levels of attributes (`meta5`, `coarse`, `fine`). The resulting metrics are plotted in Figure 5, Figure 6, and Figure 7. The figures show that across all target distributions with different support sizes, the performance (in terms of distributional metrics) appears to first improve as the batch size M increases, and then start to slowly degrade if M keeps growing (note the log scale of the x-axis in all figures). A similar pattern is also reported by Parihar et al. (2024) when using sample batches to perform distribution guidance. The turning point appears to vary across different targets: ~ 5 times of the support size in Uniform, while up to 10 times of the support size in Guassian and ZigZag. For baseline PG-DPS, the performance change appears to be minor as the batch size M changes. With a reasonable batch size M for estimating the empirical distribution (considering both attribute distribution support size and the target), our method can achieve better performance than the baselines.

Diffusion inference steps K We ablate the number of diffusion steps taken during inference. For both the baseline EDM and our method, we take $K \in \{10, 14, 18, 22, 26\}$; all other hyperparameters are frozen. We test on two granularities on all three types of targets and report the results in Table 11. The vanilla EDM generation results serve as baseline anchors. Distribution alignment appears to improve as K increases, though all variations stay within the same order of magnitude, with fairly small *relative changes*. The sample quality appears to receive monotonic improvements as K increases, which is expected and aligns with known characteristics of diffusion sampling. These results show that the gains of our method are *not* restricted to particular K values.

Oracle Model Architecture Our method is conceptually applicable to different types of differentiable oracle models, *not limited* to the ResNet classifiers instantiated in the main paper. We further showcase our method's robustness against oracle model architectures by replacing the ResNet oracle with an off-the-shelf, pretrained Vision Transformer (ViT) classifier (pkr7098, 2024). We freeze hyperparameters for our method as the ones reported in the main table. The quantitative results are in Table 12. Within the tested settings, our method outperforms PG-DPS in all metrics across all tested targets. These results repeat the pattern similar to the ResNet oracles: our method achieves better distributional alignment while preserving substantially better sample quality. Even where the TV gap is relatively modest, PG-DPS pays a considerably larger quality price. This indicates that *the gains of our method are not specific to a ResNet oracle backbone* and persist with ViT oracles under the current setup.

Runtime and Memory Benchmark We perform ablations to empirically show how runtime and memory scale over batch size M, inference steps K, and max number of iterations I. The results are in Table 13, Table 14, Table 15. The total batch runtime profile of our method is also shown in Figure 4 as the M, I, and K varies. The observed patterns from the empirical results appear to match our analyses in Appendix **B.3**:

- Runtime per sample grows roughly linearly with max number of iterations I and inference steps K.
- Runtime per sample decreases with larger batch size M because of batching while peak memory increases linearly with M.
- Peak memory usage is mostly dominated by batch size M.

E.1.2 Additional Samples

We provide qualitative samples generated by baselines and our methods in Figure 8, Figure 9, Figure 10.

Table 11: `CIFAR-100` sensitivity to sampling steps K.

K	METHOD	meta5 TV↓	meta5 JS↓	meta5 χ^2↓	meta5 FID↓	coarse TV↓	coarse JS↓	coarse χ^2↓	coarse FID↓
				Gaussian					
10	EDM	0.269	0.219	0.0932	29.7	0.327	0.279	0.146	29.6
	OURS	**0.125**	**0.113**	**0.0250**	**23.3**	**0.170**	**0.147**	**0.0424**	**21.5**
14	EDM	0.261	0.204	0.0815	20.9	0.295	0.247	0.116	20.8
	OURS	**0.127**	**0.111**	**0.0243**	**18.8**	**0.175**	**0.147**	**0.0426**	**17.7**
18	EDM	0.252	0.196	0.0749	17.4	0.274	0.227	0.0988	17.4
	OURS	**0.136**	**0.117**	**0.0272**	**16.5**	**0.176**	**0.146**	**0.0421**	**15.7**
22	EDM	0.249	0.193	0.0726	**15.3**	0.265	0.219	0.0925	15.4
	OURS	**0.138**	**0.117**	**0.0272**	15.8	**0.168**	**0.138**	**0.0377**	**14.5**
26	EDM	0.244	0.190	0.0703	**14.3**	0.260	0.214	0.0886	14.3
	OURS	**0.140**	**0.118**	**0.0277**	14.9	**0.173**	**0.141**	**0.0391**	**14.0**
				Zigzag					
10	EDM	0.0584	**0.0493**	**0.00485**	27.8	0.263	0.205	0.0818	28.9
	OURS	**0.0567**	0.0568	0.00644	**21.4**	**0.113**	**0.103**	**0.0211**	**20.7**
14	EDM	**0.0544**	**0.0479**	**0.00458**	18.8	0.204	0.167	0.0547	19.8
	OURS	0.0589	0.0524	0.00548	**16.5**	**0.105**	**0.0968**	**0.0186**	**16.7**
18	EDM	0.0670	0.0574	0.00658	15.3	0.185	0.148	0.0433	16.1
	OURS	**0.0544**	**0.0495**	**0.00489**	**14.0**	**0.103**	**0.0927**	**0.0171**	**14.8**
22	EDM	0.0798	0.0649	0.00840	13.3	0.172	0.138	0.0375	14.1
	OURS	**0.0632**	**0.0549**	**0.00602**	**12.9**	**0.104**	**0.0932**	**0.0172**	**13.9**
26	EDM	0.0921	0.0729	0.0106	**12.2**	0.165	0.133	0.0349	**12.9**
	OURS	**0.0658**	**0.0571**	**0.00650**	12.3	**0.105**	**0.0905**	**0.0163**	13.1
				Uniform					
10	EDM	0.142	0.116	0.0267	28.7	0.240	0.197	0.0758	28.7
	OURS	**0.0477**	**0.0448**	**0.00400**	**20.6**	**0.102**	**0.0968**	**0.0185**	**19.3**
14	EDM	0.104	0.0823	0.0135	19.3	0.171	0.144	0.0409	19.3
	OURS	**0.0312**	**0.0306**	**0.00187**	**15.6**	**0.0848**	**0.0757**	**0.0114**	**14.7**
18	EDM	0.0830	0.0651	0.00845	15.5	0.132	0.114	0.0255	15.5
	OURS	**0.0281**	**0.0292**	**0.00171**	**13.1**	**0.0690**	**0.0655**	**0.00853**	**12.6**
22	EDM	0.0702	0.0544	0.00592	13.4	0.107	0.0951	0.0180	13.4
	OURS	**0.0255**	**0.0244**	**0.00119**	**11.8**	**0.0610**	**0.0594**	**0.00703**	**11.4**
26	EDM	0.0579	0.0446	0.00398	12.2	0.0935	0.0832	0.0138	12.2
	OURS	**0.0240**	**0.0214**	**9.14e-04**	**11.1**	**0.0578**	**0.0556**	**0.00616**	**10.9**

Table 12: Quantitative evaluation metrics (all lower the better) on `CIFAR-100` with a pretrained ViT oracle model.

METHOD	meta5 TV↓	meta5 JS↓	meta5 χ^2↓	meta5 FID↓	coarse TV↓	coarse JS↓	coarse χ^2↓	coarse FID↓	fine TV↓	fine JS↓	fine χ^2↓	fine FID↓
					Gaussian							
PG-DPS	0.149	0.122	0.0294	18.3	0.154	0.126	0.0312	32.5	0.161	0.152	0.0451	23.3
OURS	**0.103**	**0.0781**	**0.0122**	**17.7**	**0.132**	**0.108**	**0.0233**	**16.1**	**0.133**	**0.122**	**0.0292**	**14.8**
					Zigzag							
PG-DPS	0.0420	0.0334	0.00223	17.1	0.0786	0.0834	0.0138	20.9	0.141	0.130	0.0331	20.3
OURS	**0.0354**	**0.0290**	**0.00168**	**14.2**	**0.0634**	**0.0594**	**0.00703**	**14.7**	**0.0963**	**0.0868**	**0.0149**	**13.0**
					Uniform							
PG-DPS	0.0186	0.0160	5.13e-04	16.9	0.0875	0.0823	0.0134	19.0	0.134	0.120	0.0282	20.0
OURS	**0.00986**	**0.00970**	**1.88e-04**	**12.9**	**0.0326**	**0.0342**	**0.00233**	**12.8**	**0.0730**	**0.0672**	**0.00897**	**12.8**

Table 13: Runtime per sample and peak memory as batch size M varies.

M	K	I	Runtime per sample (ms)	Peak Memory (GB)
8	18	10	912 ± 1.46	1.21
16	18	10	473 ± 0.981	2.11
32	18	10	260 ± 0.239	3.93
64	18	10	221 ± 0.663	7.59
128	18	10	209 ± 0.238	14.9
256	18	10	203 ± 0.105	29.5

Table 14: Runtime per sample and peak memory as max iterations I varies.

I	M	K	Runtime per sample (ms)	Peak Memory (GB)
4	32	18	108 ± 0.15	3.93
6	32	18	159 ± 0.291	3.93
8	32	18	210 ± 0.309	3.93
10	32	18	260 ± 0.181	3.93
12	32	18	308 ± 0.274	3.93
14	32	18	358 ± 0.195	3.93

Table 15: Runtime per sample and peak memory as inference steps K varies.

K	M	I	Runtime per sample (ms)	Peak Memory (GB)
10	32	10	146 ± 0.242	3.92
14	32	10	202 ± 0.239	3.93
18	32	10	260 ± 0.202	3.93
22	32	10	317 ± 0.206	3.94
26	32	10	370 ± 0.306	3.94
30	32	10	428 ± 0.213	3.95

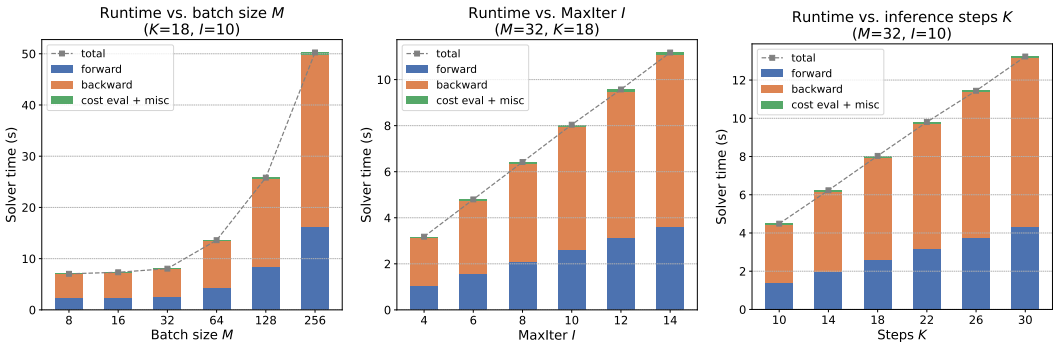


Figure 4: Total runtime profile of our method as batch size M, max iterations I and inference steps K scale. Note the *log scale of the x-axis* in the figure for batch M (left).

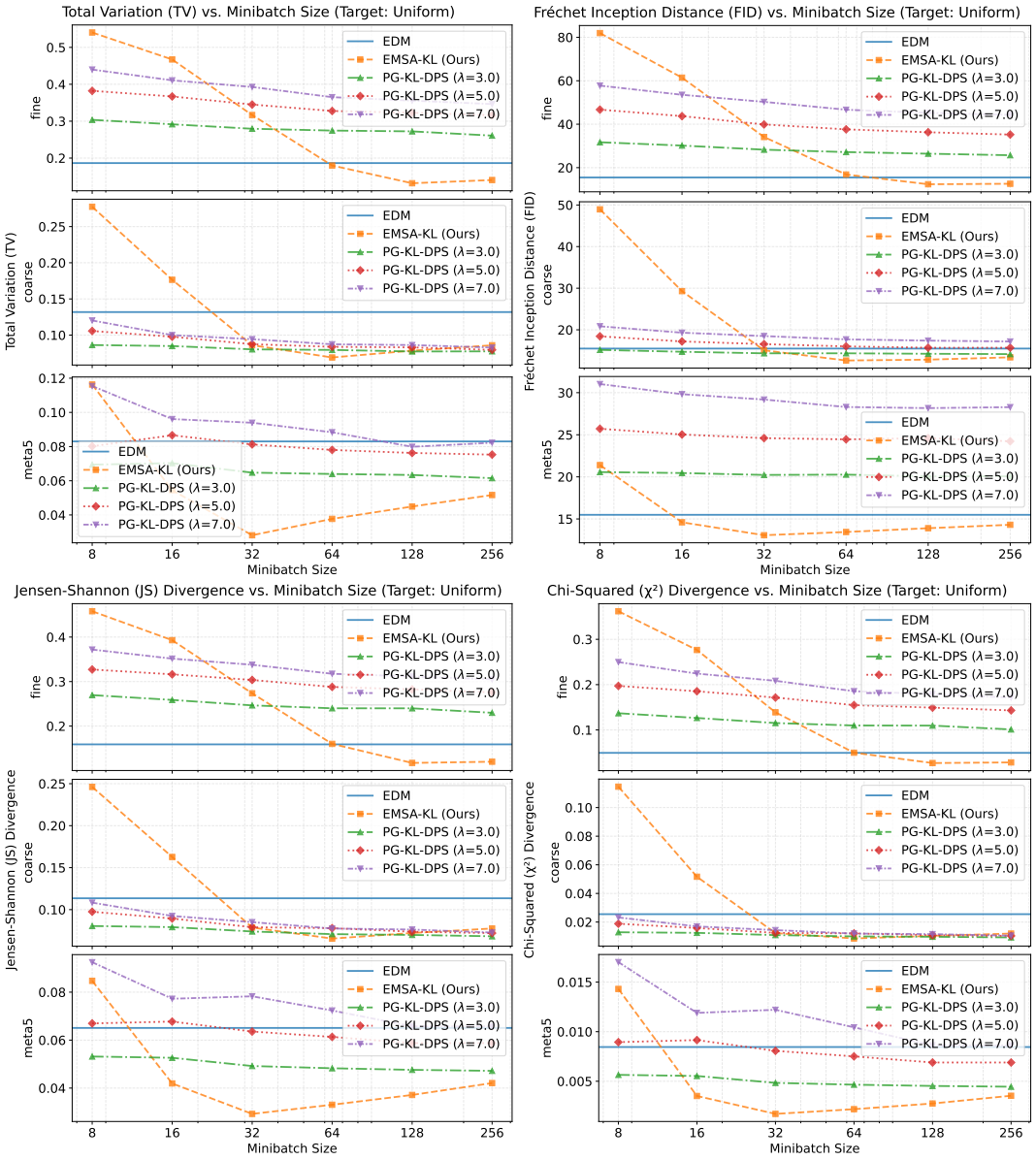


Figure 5: Total Variation (top left), FID (top right), Jensen-Shannon divergence, and χ^2 divergence metrics with different batch size M for target Uniform.

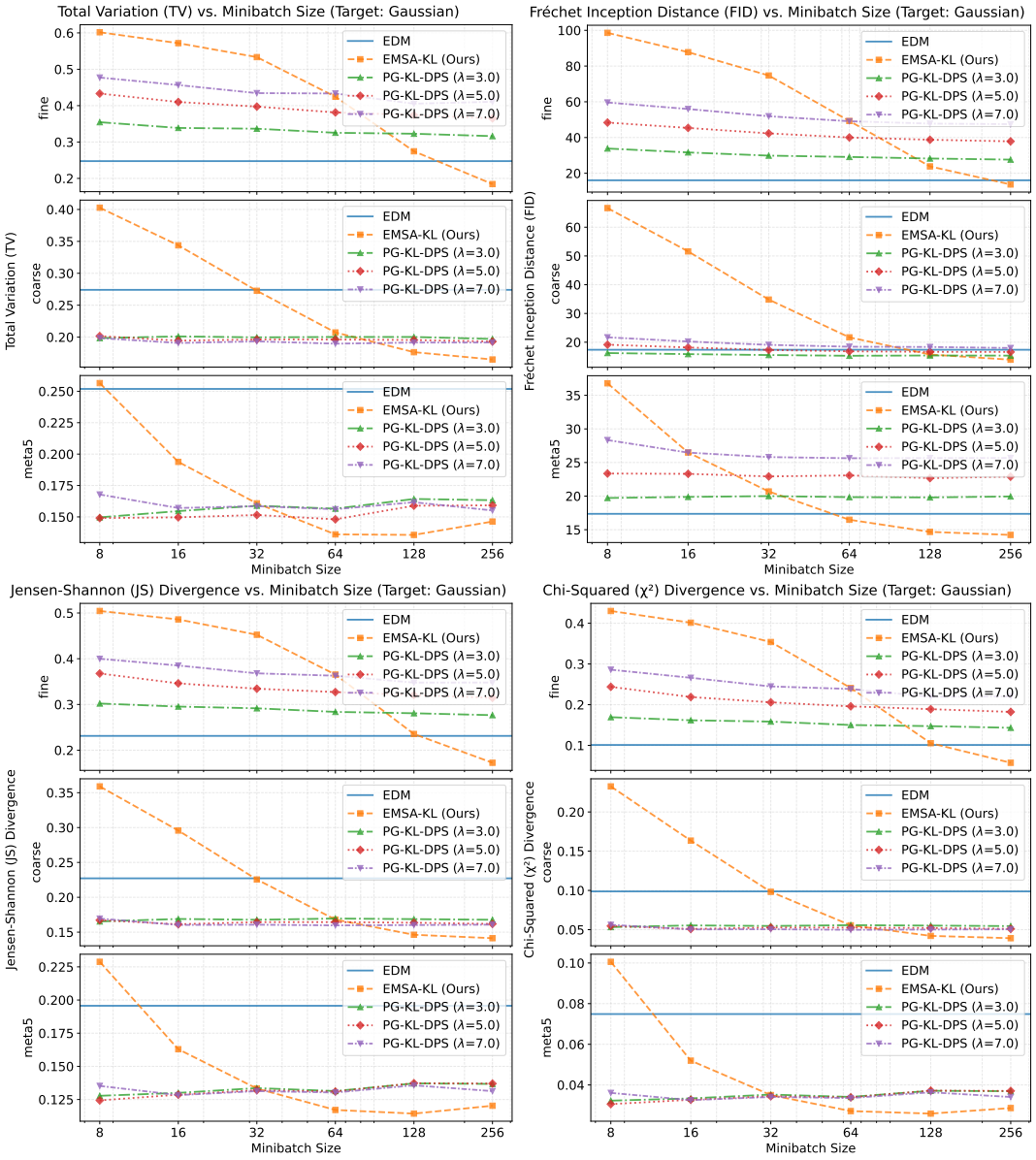


Figure 6: Total Variation (top left), FID (top right), Jensen-Shannon divergence, and χ^2 divergence metrics with different batch size M when target is Gaussian.

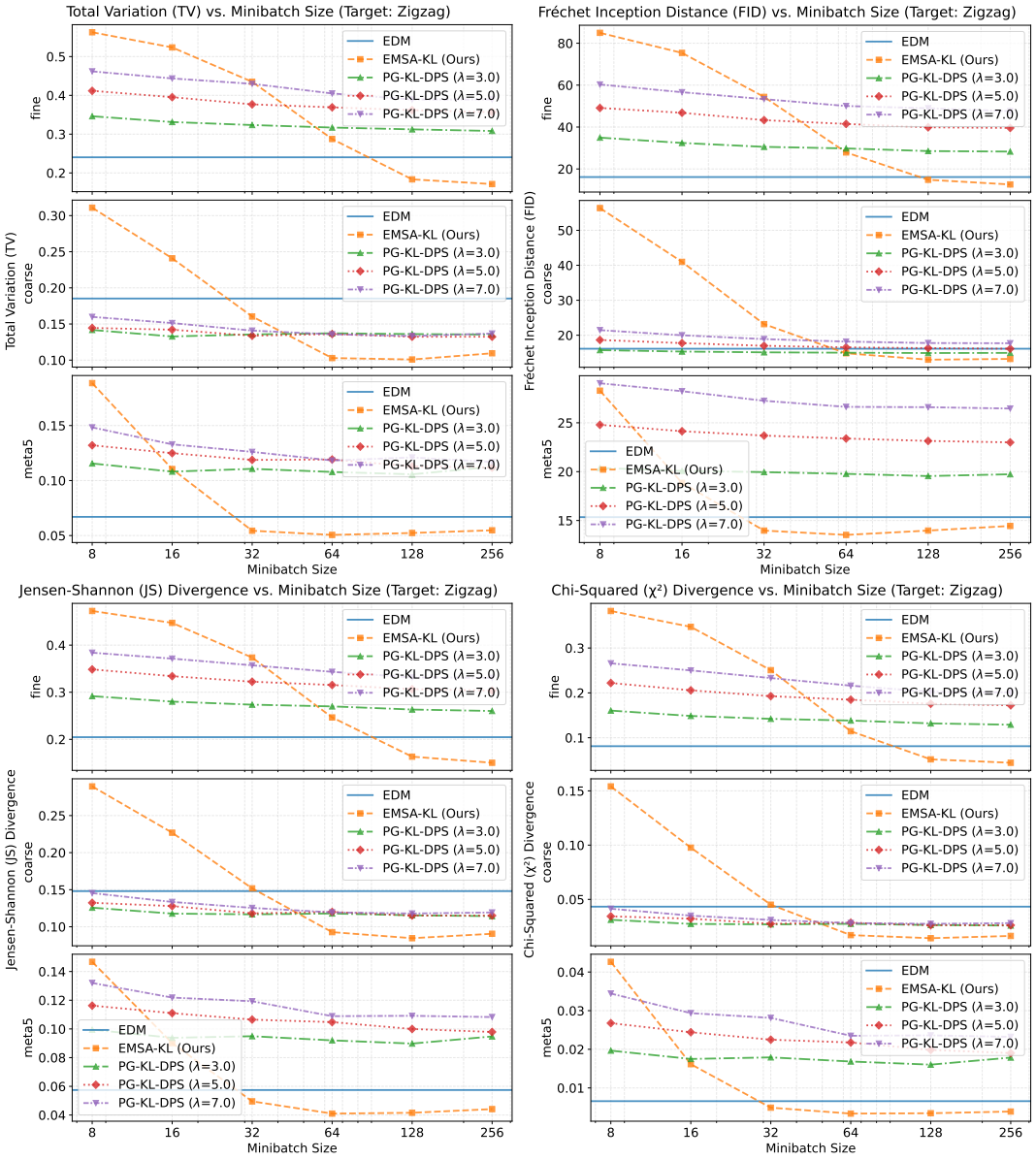


Figure 7: Total Variation (top left), FID (top right), Jensen-Shannon divergence, and χ^2 divergence metrics with different batch size M when target is ZigZag.

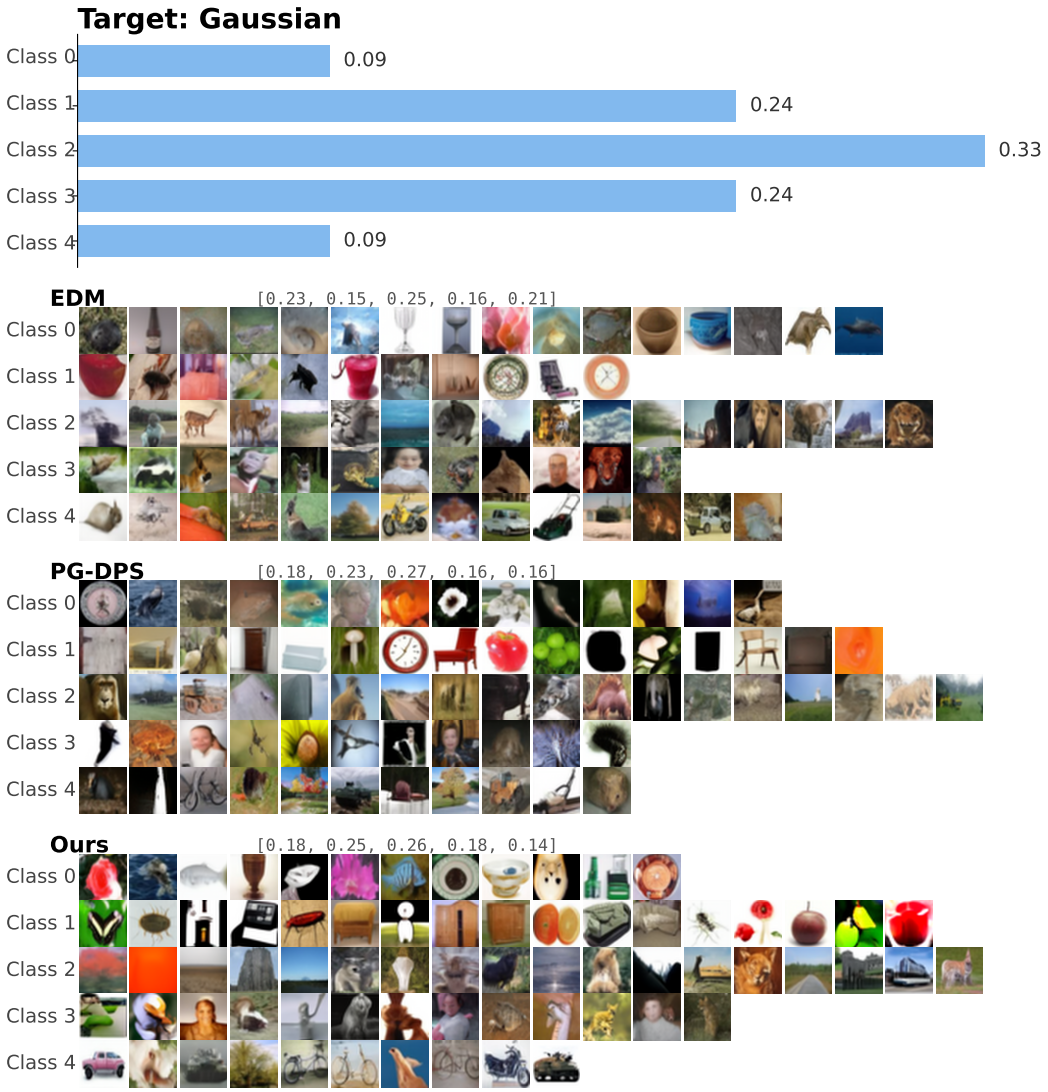


Figure 8: Given a target attribute distribution as `meta5`-Guassian, this figure provides a qualitative comparison of samples generated by different methods in the CIFAR experiment reported in Table 1. The top block shows the target attribute distribution. Each block along the vertical direction exhibits samples generated by one method, clustered by the samples' attribute values, and the empirical attribute distribution is also marked.

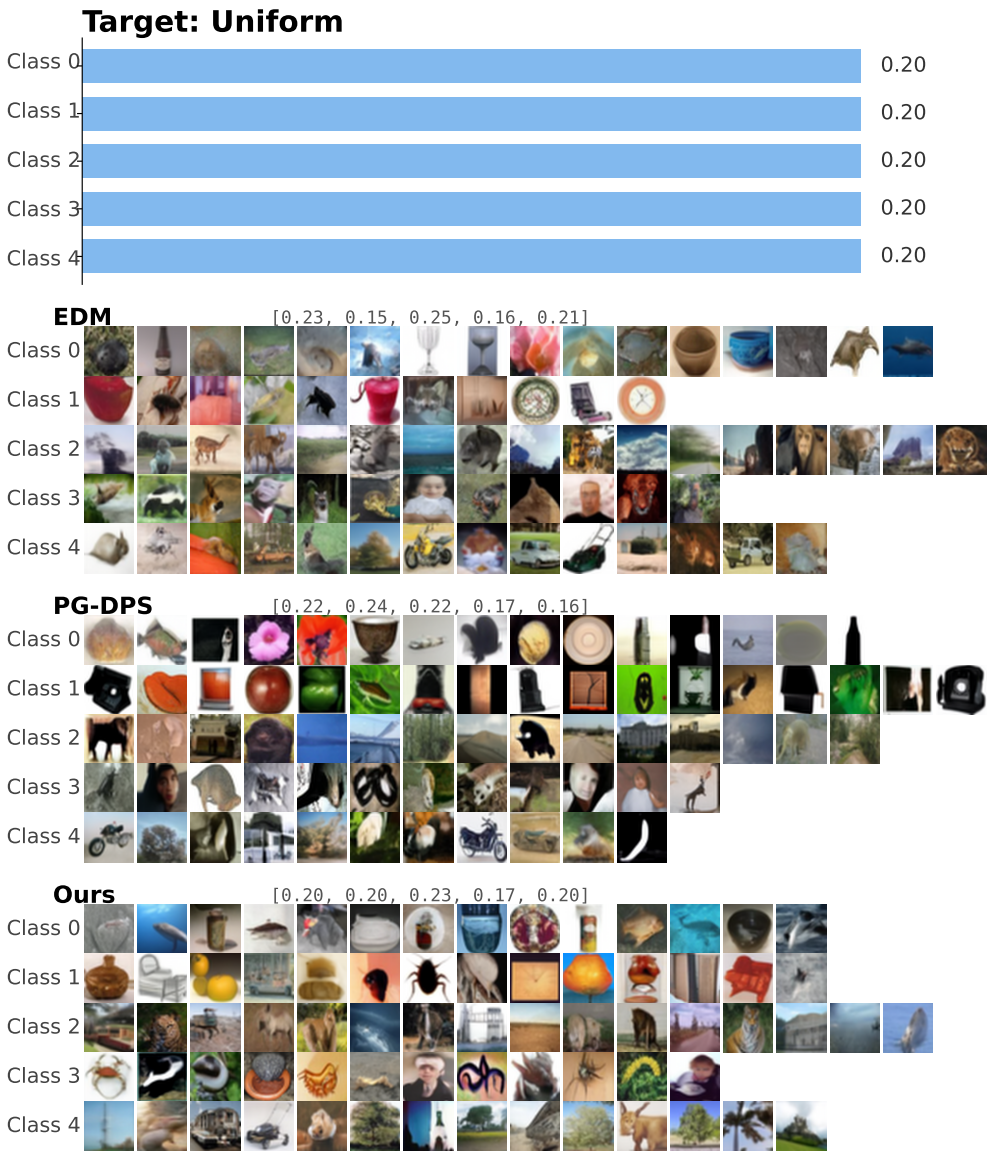


Figure 9: Given a target attribute distribution as `meta5`-Uniform, this figure provides a qualitative comparison of samples generated by different methods in the CIFAR experiment reported in Table 1. The top block shows the target attribute distribution. Each block along the vertical direction exhibits samples generated by one method, clustered by the samples' attribute values, and the empirical attribute distribution is also marked.

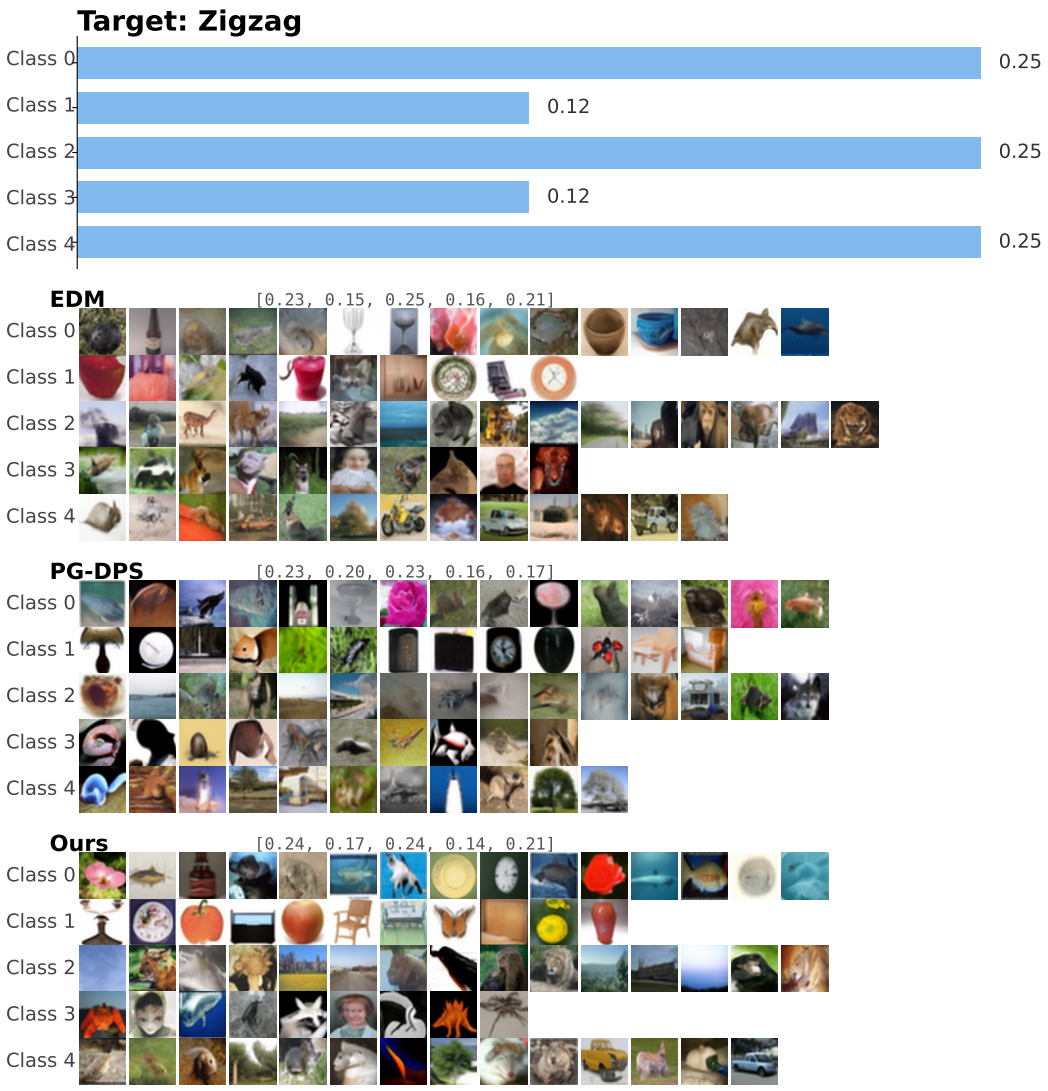


Figure 10: Given a target attribute distribution as `meta5`-Zigzag, this figure provides a qualitative comparison of samples generated by different methods in the CIFAR experiment reported in Table 1. The top block shows the target attribute distribution. Each block along the vertical direction exhibits samples generated by one method, clustered by the samples' attribute values, and the empirical attribute distribution is also marked.

Table 16: FFHQ single-attribute sensitivity to sampling steps K.

K	Method	age TV↓	age FD↓	age FID↓	age CMMD↓	gender TV↓	gender FD↓	gender FID↓	gender CMMD↓	race TV↓	race FD↓	race FID↓	race CMMD↓
Uniform													
15	DDIM	.190	.234	50.3	.816	.0427	.0611	51.1	.772	.589	.686	56.9	**.940**
	Ours	**.0240**	**.0280**	**47.7**	**.703**	**.00**	**.00627**	**47.2**	**.609**	**.0594**	**.0651**	**51.9**	1.10
20	DDIM	.200	.241	48.0	.768	.0313	.0460	48.4	.723	.575	.669	53.9	**.886**
	Ours	**.0156**	**.0201**	**46.0**	**.678**	**.00938**	**.0139**	**45.6**	**.600**	**.0573**	**.0652**	**48.7**	.924
25	DDIM	.207	.249	47.2	.750	.0417	.0518	47.3	.705	.560	.650	52.4	.862
	Ours	**.0229**	**.0276**	**45.2**	**.656**	**.00521**	**.00870**	**45.0**	**.586**	**.0281**	**.0379**	**47.6**	**.836**
30	DDIM	.218	.259	47.5	.764	.0458	.0621	47.2	.716	.531	.624	51.4	.864
	Ours	**.0354**	**.0494**	**44.9**	**.667**	**.00313**	**.00737**	**44.5**	**.587**	**.0469**	**.0466**	**48.1**	**.809**
Custom-1													
		[4 : 1 : 3]				[2 : 8]				[4 : 3 : 2 : 1]			
15	DDIM	.398	.490	52.3	1.04	.257	.363	52.1	1.06	.439	.534	51.7	**.932**
	Ours	**.0438**	**.0564**	**49.0**	**.842**	**.00729**	**.00863**	**49.0**	**.861**	**.0500**	**.0565**	**45.3**	.942
20	DDIM	.408	.498	50.1	.982	.269	.378	49.6	1.00	.425	.517	49.4	.881
	Ours	**.0427**	**.0560**	**47.2**	**.814**	**.0125**	**.0174**	**47.8**	**.819**	**.0375**	**.0416**	**43.2**	**.809**
25	DDIM	.416	.507	49.7	.959	.258	.372	48.6	.979	.410	.500	48.7	.861
	Ours	**.0490**	**.0633**	**46.1**	**.791**	**.00938**	**.0165**	**46.5**	**.804**	**.0427**	**.0618**	**43.0**	**.741**
30	DDIM	.426	.517	49.7	.963	.254	.362	48.2	.979	.381	.476	48.6	.870
	Ours	**.0563**	**.0735**	**46.1**	**.815**	**.0146**	**.0212**	**46.6**	**.791**	**.0385**	**.0463**	**41.9**	**.725**

E.2 Additional Results for Face Generation Experiment

E.2.1 Ablations

Diffusion inference steps K We add a similar ablaton on the number of diffusion steps taken during inference. For both the baseline DDIM and our method, we take $K \in \{15, 20, 25, 30\}$; all other hyperparameters are frozen. Results are reported in Table 16. Findings are similar to the CIFAR experiments: the gains of our method are *not* restricted to particular K values.

Oracle Model Architecture We perform a similar oracle model ablation as for the CIFAR experiment, replacing the ResNet oracle with another off-the-shelf ViT classifier (syntheticbot, 2025), pretrained on the FairFace dataset (Karkkainen & Joo, 2021). We freeze hyperparameters for our method as the ones reported in the main table. The results are in Table 18. These results reveal the similar pattern as in the main table: our method achieves better distributional alignment while preserving substantially better sample quality. This indicates that the gains of our method are *not* specific to a ResNet oracle backbone and persist with ViT oracles under the current setup.

E.2.2 Additional Metric

CMMD for Face Images FID is not necessarily the most ideal image quality metric for FFHQ. As such, we add the CLIP embedding Maximum Mean Discrepancy (CMMD) (Jayasumana et al., 2024) to evaluate the sample quality for the face image generation. CMMD is a suitable *population-level* measurement here. For CMMD calculation, we adopt the default RBF Gaussian kernel with default bandwidth $\sigma = 10$ as in (Jayasumana et al., 2024); the reference sets of dataset samples are the same as those we used for computing FID. Tables 20 and 21 present the CMMD results. The overall pattern aligns with the FID results in Tables 2 and 3, further supporting that our method better preserve sample quality than the baselines under the tested settings.

Table 17: FFHQ joint age×gender×race sensitivity to sampling steps K.

K	METHOD	TV↓	FD↓	FID↓	CMMD↓
		Uniform Joint			
15	DDIM	.588	.320	49.6	.987
	OURS	**.134**	**.0672**	**43.6**	**.969**
20	DDIM	.575	.314	47.1	.936
	OURS	**.121**	**.0616**	**41.9**	**.812**
25	DDIM	.566	.308	46.3	.912
	OURS	**.112**	**.0606**	**41.3**	**.762**
30	DDIM	.549	.299	46.1	.919
	OURS	**.124**	**.0621**	**41.7**	**.741**
		Custom Joint			
		$[5:2:3] \times [3:7] \times [4:3:2:1]$			
15	DDIM	.480	.309	46.3	**.962**
	OURS	**.156**	**.0857**	**45.6**	1.02
20	DDIM	.495	.308	44.6	.916
	OURS	**.138**	**.0868**	**43.3**	**.839**
25	DDIM	.493	.305	44.3	.895
	OURS	**.142**	**.0895**	**42.2**	**.747**
30	DDIM	.478	.299	44.5	.902
	OURS	**.146**	**.0912**	**42.4**	**.766**

Table 18: Quantitative evaluation metrics on face generation with **single-attribute** control using the pretrained ViT oracle.

	age					**gender**					**race**				
METHOD	TV↓	JS↓	χ^2↓	FD↓	FID↓	TV↓	JS↓	χ^2↓	FD↓	FID↓	TV↓	JS↓	χ^2↓	FD↓	FID↓
						Uniform									
PG-DPS	.535	.444	.329	.617	92.6	.247	.181	.0649	.340	53.9	.745	.605	.590	.788	62.4
OURS	**.0167**	**.0127**	**3.24e-04**	**.0367**	**47.1**	**.0167**	**.0118**	**2.78e-04**	**.0147**	**47.3**	**.0500**	**.0434**	**.00376**	**.0272**	**49.7**
						Custom-1									
	$[4:1:3]$					$[2:8]$					$[4:3:2:1]$				
PG-DPS	.849	.667	.729	.978	63.3	.571	.416	.326	.784	81.6	.398	.403	.255	.448	80.1
OURS	**.0313**	**.0281**	**.00158**	**.0417**	**47.6**	**.0219**	**.0198**	**7.80e-04**	**.0507**	**46.4**	**.243**	**.181**	**.0645**	**.104**	**47.0**
						Custom-2									
	$[2:3:4]$					$[7:3]$					$[1:1:4:4]$				
PG-DPS	.740	.589	.572	.855	119	.236	.170	.0574	.321	159	.868	.687	.761	.902	260
OURS	**.0358**	**.0278**	**.00154**	**.0240**	**48.6**	**.00625**	**.00484**	**4.68e-05**	**.0126**	**50.8**	**.0785**	**.0728**	**.0105**	**.0818**	**49.3**

Table 19: Quantitative results of face generation controlling the **joint distribution** over age, gender, and race with the pretrained ViT oracle.

	age×gender×race				
METHOD	TV↓	JS↓	χ^2↓	FD↓	FID↓
	Uniform Joint				
PG-DPS	.865	.699	.771	.515	70.0
OURS	**.0935**	**.0794**	**.0126**	**.0134**	**45.0**
	Custom Joint				
	$[5:2:3] \times [3:7] \times [4:3:2:1]$				
PG-DPS	.905	.728	.828	.685	67.8
OURS	**.136**	**.143**	**.0363**	**.0319**	**43.3**

Table 20: CMMD on FFHQ face generation with single-attribute targets (lower is better).

METHOD	Age CMMD↓	Gender CMMD↓	Race CMMD↓
		Uniform	
DDIM	.750	.705	.862
PG-DPS	1.24	.899	1.26
OURS	**.656**	**.586**	**.836**
		Custom-1	
	$[4:1:3]$	$[2:8]$	$[4:3:2:1]$
DDIM	.959	.979	.861
PG-DPS	1.62	1.76	1.92
OURS	**.791**	**.804**	**.741**
		Custom-2	
	$[2:3:4]$	$[7:3]$	$[1:1:4:4]$
DDIM	.990	.969	.949
PG-DPS	1.34	1.89	3.11
OURS	**.782**	**.803**	**.855**

Table 21: CMMD on FFHQ face generation with joint age×gender×race targets (lower is better).

METHOD	CMMD↓
	UniformJoint
DDIM	.912
PG-DPS	1.68
OURS	**.762**
	CustomJoint
	$[5:2:3] \times [3:7] \times [4:3:2:1]$
DDIM	.895
PG-DPS	1.19
OURS	**.747**

E.2.3 Additional Samples

We present additional qualitative samples in Figure 11, Figure 12, Figure 13.

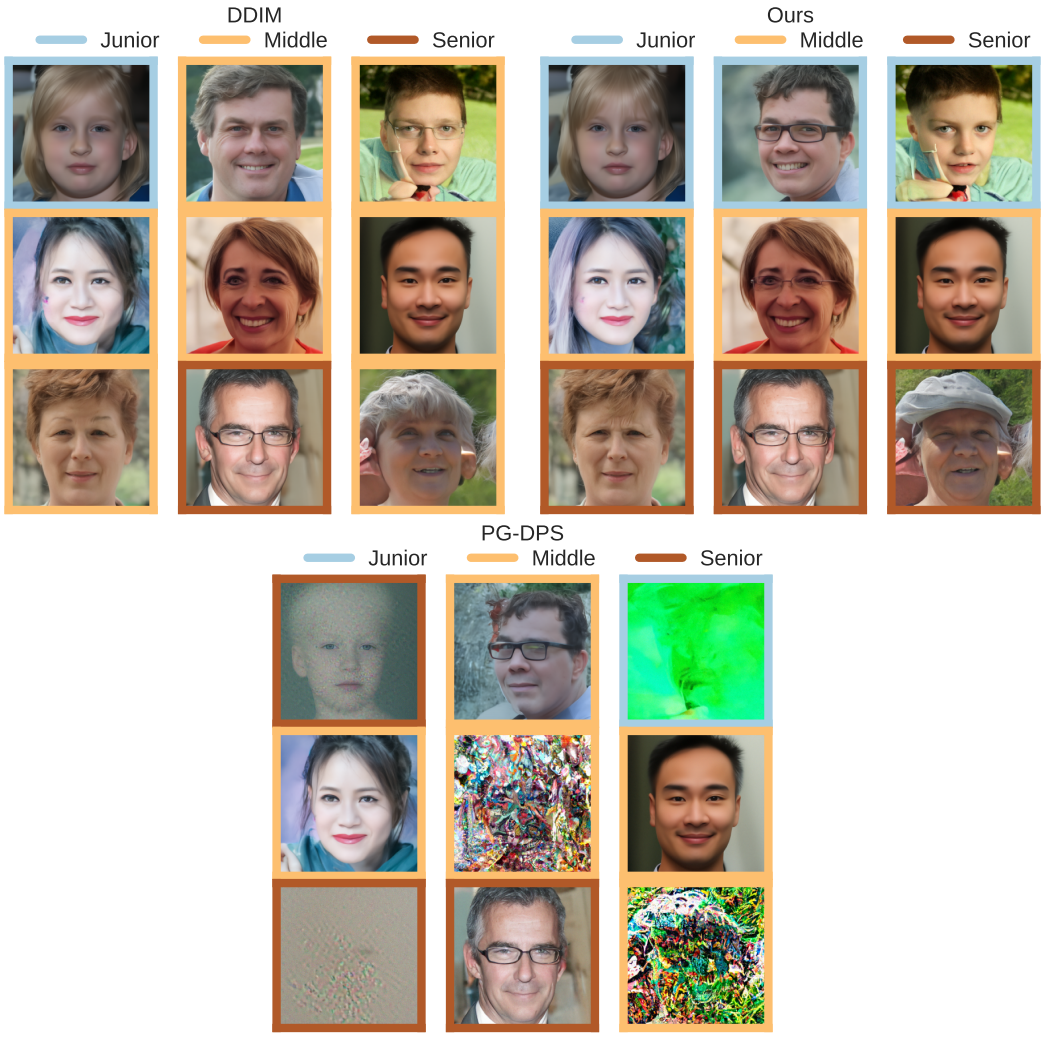


Figure 11: Additional samples from all methods in the face generation experiment.

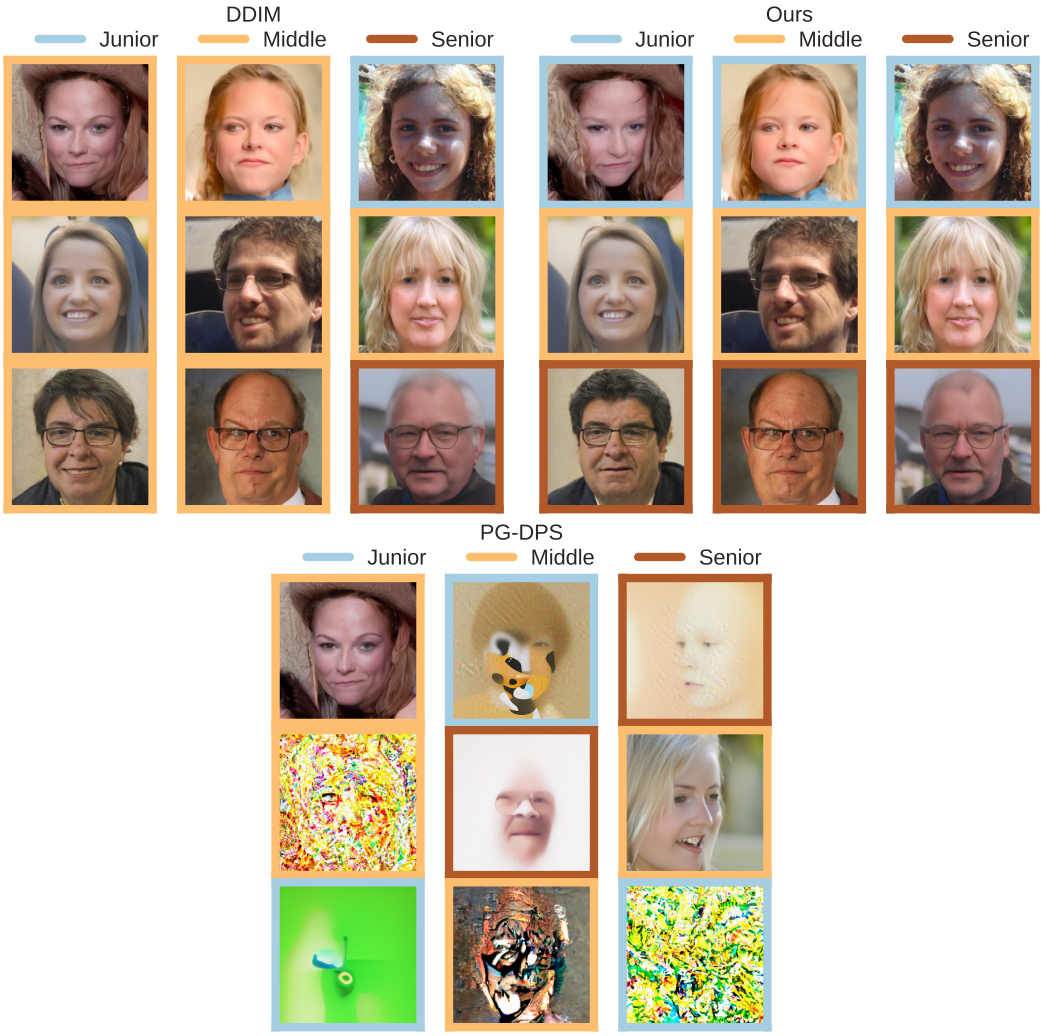


Figure 12: Additional samples from all methods in the face generation experiment.

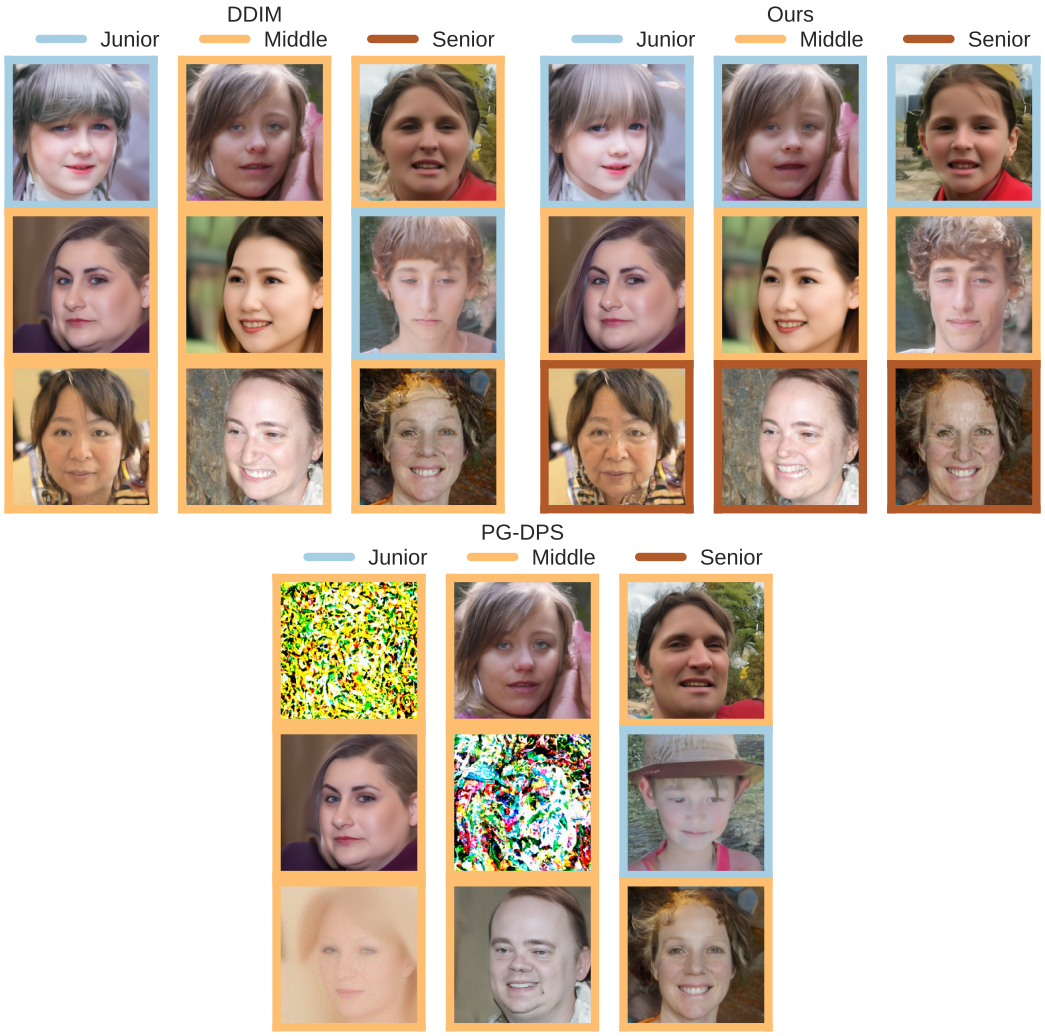


Figure 13: Additional samples from all methods in the face generation experiment.

F Discussions and Limitations

Computational Cost As analyzed in Appendix **B.3**, the computational complexity of our method scales linearly with the number of optimization iterations I, inference steps K, and batch size M. The primary bottleneck is the per-step VJP computation, which necessitates differentiating through the diffusion model. Empirically, we find $I \approx 6 - 12$ generally suffices, and early stopping can further reduce runtime when the distributional objective plateaus. While our approach incurs a higher inference cost than the compared baselines, it *effectively eliminates the need for expensive model fine-tuning or retraining* and *yields stronger distribution-level alignment* in our experiments. We view this computational overhead as a necessary trade-off for high-fidelity distributional matching without model parameter updates.

Batch Size for Empirical Distribution Estimation Our method needs a reasonable batch size M for estimating the empirical attribute distribution. As shown in the ablation study in Section E.1, when the batch size M is too small to provide a reliable empirical distribution for an attribute distribution with a potentially large support size, the distribution alignment performance would degrade. However, the same issue exists for other approaches including training-based methods such as Parihar et al. (2024).