UNIGUIDE: LEARNING GUIDANCE POLICIES FOR MULTI-OBJECTIVE DIFFUSION SAMPLING

Mahmoud Hegazy[1,*] **Navid Bagheri Shouraki**[2,3,*] **Eric Moulines**[2,4]
Aymeric Dieuleveut[1] **Michael I. Jordan**[5,6] **Yazid Janati**[4,7]

[1]CMAP, Ecole Polytechnique [2]EPITA Research Laboratory [3]Sorbonne University
[4]MBZUAI [5]Inria, Paris [6]UC Berkeley [7]Institute of Foundation Models

ABSTRACT

Guidance is the primary mechanism for steering conditional diffusion models, yet modern samplers rely on coupled, step-dependent design choices that obscure trade-offs between alignment, realism, and diversity. We introduce UniGuide, a unified framework that formulates guided diffusion sampling as a sequential decision-making problem, where guidance actions are selected along the denoising trajectory based on the evolving model state. UniGuide employs a flexible parameterization assigning distinct guidance to the denoiser and the noise-prediction term. To address multi-objective trade-offs, we learn a low-dimensional, preference-conditioned policy that maps intermediate model states to per-step guidance parameters. This enables continuous traversal of a Pareto frontier at inference time via a preference vector, eliminating the need for retuning or manual schedule design. Experiments on conditional image generation demonstrate effective Pareto-optimal guidance with pretrained diffusion backbones.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) provide a general framework for generative modeling by reversing a gradual corruption process that transforms data into noise. By approximating the reverse-time dynamics, they achieve strong empirical performance across conditional generation tasks such as text-to-image synthesis (Rombach et al., 2022; Podell et al., 2023; Esser et al., 2024). Nonetheless, unguided conditional diffusion models often produce samples with weak alignment or limited perceptual quality, making guidance essential for effective generation (Dhariwal & Nichol, 2021; Ho & Salimans, 2022).

Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) is a widely adopted approach, which combines conditional and unconditional denoiser predictions using a guidance scale. While increasing this scale typically improves alignment and perceptual quality, a high guidance scale is known to introduce systematic failures, including reduced diversity, oversimplified structure, and visual artifacts such as oversaturation (Saharia et al., 2022; Sadat et al., 2025a). These effects reveal inherent multi-objective trade-offs in conditional generation, involving competing objectives such as alignment, diversity, and realism (Astolfi et al., 2024; Annadani et al., 2025; Yao et al., 2024).

Although CFG is often described as a single scalar control, modern pipelines rarely use it in this form. As practical implementations rely on noise-dependent schedules, solver-coupled reparameterization, and constraints on update directions to mitigate the adverse effects of strong guidance (Chung et al., 2025; Xi et al., 2024; Kynkäänniemi et al., 2024; Sadat et al., 2025b; Malarz et al., 2025; Jin et al., 2025). Thus these static schedules are typically tuned using heuristic visual fidelity metrics such as FID (Heusel et al., 2017) and PRDC (Precision/Recall/Density/Coverage) (Naeem et al., 2020), making the balance among competing objectives implicit and difficult to control.

These observations motivate this work to move beyond fixed schedules by replacing a single hand-tuned rule with a learned *family* of guidance policies supporting controllable trade-offs at inference

*Authors contributed equally

time. We formalize guidance as an explicit policy that controls diffusion sampling dynamics. Concretely, we introduce a unified parameterization that recovers and interpolates between CFG and CFG++ (Chung et al., 2025) and cast guidance as a multi-objective reinforcement learning problem. In contrast to recent works (Black et al., 2023; Ren et al., 2024; Liu et al., 2025), which aim to fine-tune diffusion models using reinforcement learning, we rely entirely on pretrained diffusion backbones and learn a separate lightweight policy to control the guidance scale. By conditioning on preference vectors, this approach enables inference-time navigation of achievable trade-offs.

Contributions and Outline. Section 2 presents a unified guidance policy design space that separately controls guidance applied to the denoised estimate and the noise-prediction term, recovering CFG and CFG++ as special cases. In Section 3, we frame guided diffusion sampling as a sequential decision-making problem and learn a preference-conditioned policy that selects per-step guidance actions. Section 4 details our training algorithm. Finally, after detailing related works in Section 5, Section 6 contains experiments on class-conditional generation on ImageNet-512 (Deng et al., 2009) and shows improved performance over baselines across the alignment-diversity Pareto-front.

2 Background

Diffusion models. Let p_0 denote the target data distribution. For a noise level $\sigma \in [0, \sigma_{\max}]$, we define the noised variable

$$X_\sigma = X_0 + \sigma\varepsilon\,, \quad X_0 \sim p_0, \quad \varepsilon \sim \mathcal{N}(0, I) \tag{1}$$

and denote its marginal density by p_σ. This construction defines a family of smoothed distributions $\{p_\sigma\}_{\sigma\in[0,\sigma_{\max}]}$, where $p_{\sigma_{\max}}$ approaches a Gaussian distribution for sufficiently large $\sigma_{\max}$.

Diffusion-based generative models (Song & Ermon, 2019; Song et al., 2021b) aim to sample from p_0 by reversing the forward noising process. In the continuous formulation, sampling can be expressed as integrating a reverse-time ordinary differential equation over decreasing noise levels. Following the formulation of Karras et al. (2022), the probability-flow dynamics are given by

$$\frac{\mathrm{d}\mathbf{x}_\sigma}{\mathrm{d}\sigma} = \frac{\mathbf{x}_\sigma - \hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma)}{\sigma}, \tag{2}$$

where $\hat{\mathbf{x}}_0(x, \sigma) = \mathbb{E}[X_0 \,|\, X_\sigma = x]$ is the conditional expectation associated with the joint distribution defined by (1). In practice, the probability-flow ODE is not integrated in closed form. Instead, sampling is performed by discretizing the noise interval $[\sigma_{\max}, 0]$ into a finite sequence $\sigma_T > \sigma_{T-1} > \cdots > \sigma_0$ and applying a numerical integration scheme such as Euler or Heun (Karras et al., 2022). For instance, starting from an initial sample $X_{\sigma_T} \sim \mathcal{N}(0, \sigma_T^2 I)$, discretizing the probability-flow ODE using the Euler method yields the update

$$X_{\sigma_t} = X_{\sigma_{t+1}} + \frac{X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1})}{\sigma_{t+1}} (\sigma_t - \sigma_{t+1})\,. \tag{3}$$

We define the noise prediction at σ_{t+1} by $\hat{\boldsymbol{\varepsilon}}(X_{\sigma_{t+1}}, \sigma_{t+1}) \coloneqq \big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1})\big)/\sigma_{t+1}$, which corresponds to the conditional expectation of ε given $X_{\sigma_{t+1}}$ under the joint model (1). Rearranging Eq. (3) yields the deterministic DDIM update (Song et al., 2021a) of the form

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}} X_{\sigma_{t+1}} + \left(1 - \frac{\sigma_t}{\sigma_{t+1}}\right) \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}).$$

Other solvers, such as Heun's method, modify this update while preserving the underlying ODE.

In many cases, the denoising process is conditioned on additional information. In particular, let $\hat{\mathbf{x}}_0(\mathbf{x}, \sigma|\mathbf{c}) = \mathbb{E}[X_0 \mid X_\sigma = \mathbf{x}, \mathbf{c}]$ denote the conditional denoiser, where $\mathbf{c}$ is the conditioning input (e.g., a class label or text prompt). The unconditional denoiser may be expressed within the same notation using the null-conditioned case, $\hat{\mathbf{x}}_0(\mathbf{x}, \sigma) \coloneqq \hat{\mathbf{x}}_0(\mathbf{x}, \sigma|\varnothing)$. In addition, we can analogously define the conditional noise prediction as $\hat{\boldsymbol{\varepsilon}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}) \coloneqq \big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c})\big)/\sigma_{t+1}$.

Classifier-free guidance (CFG). In practice, sampling from the conditional denoiser alone often yields weak alignment or perceptual quality. Ho & Salimans (2022) address this by amplifying

the effect of conditioning during sampling. During training, the conditioning signal is randomly dropped and replaced by a null context, allowing a single network to approximate both $\hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma|\mathbf{c})$ and $\hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma)$. At inference time, guidance is implemented by forming the guided denoiser:

$$\hat{\mathbf{x}}_0^{\mathsf{cfg}}(\mathbf{x}_\sigma, \sigma|\mathbf{c}; \omega) := \hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma) + \omega\big(\hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma|\mathbf{c}) - \hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma)\big), \tag{4}$$

where $\omega > 1$ is the guidance scale. When combined with an Euler step, this yields the update

$$X_{\sigma_t} = \hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; \omega) + \frac{\sigma_t}{\sigma_{t+1}}\Big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; \omega)\Big). \tag{5}$$

While CFG significantly improved upon earlier guidance approaches, high values of ω are typically required to improve alignment with the conditioning signal and perceptual quality, but they often reduce diversity and may introduce artifacts at large values (e.g., mode collapse, oversaturation) (Xi et al., 2024; Sadat et al., 2025a). At the same time, although CFG is often motivated as sampling from a modified conditional distribution, analyses show that the guided update does not correspond to any valid diffusion process and instead alters the dynamics in a way that is only locally interpretable (Moufad et al., 2025; Bradley & Nakkiran, 2024; Chidambaram et al., 2024). Despite this mismatch, CFG remains highly effective in practice, suggesting that guidance acts as a corrective mechanism for denoiser approximation errors (Bradley & Nakkiran, 2024).

These shortcomings motivate CFG++ (Chung et al., 2025), which applies guidance only to the denoised estimate while keeping the renoising term unconditional. Thus, it prevents the guided signal from contaminating the noise estimate. As shown in the appendix A, the CFG++ update can be equivalently written as:

$$X_{\sigma_t} = \hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; \omega) + \frac{\sigma_t}{\sigma_{t+1}}\Big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; 0)\Big). \tag{6}$$

Unified parametrization. Eqs. (5) and (6) show that CFG++ can be viewed as a modification of the CFG Euler step. In particular, Eq. (5) can be expressed as a weighted combination of the guided denoiser $\hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; \omega)$ and the conditional noise prediction $\hat{\boldsymbol{\varepsilon}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c})$. By contrast, CFG++ replaces the conditional noise prediction with the unconditional one, resulting in a weighted combination of $\hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; \omega)$ and $\hat{\boldsymbol{\varepsilon}}(X_{\sigma_{t+1}}, \sigma_{t+1})$.

As such both approaches may be unified through the following parameterization

$$X_{\sigma_t} = \hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; \omega) + \frac{\sigma_t}{\sigma_{t+1}}\Big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}; \lambda)\Big), \tag{7}$$

with CFG recovered by setting $\omega = \lambda$ and CFG++ by setting $\lambda = 0$. Nonetheless, Eq. (7) is fundamentally tied to Euler's discretization of the probability-flow ODE at chosen noise levels $\sigma_T, \ldots, \sigma_0$.

The dependence on the solver may be relaxed by reinterpreting Eq. (7) as Euler's method applied to the continuous-time ODE

$$\frac{\mathrm{d}\mathbf{x}_\sigma}{\mathrm{d}\sigma} = \frac{\mathbf{x}_\sigma - \big((1-\lambda)\hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma) + \lambda\hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma|\mathbf{c})\big)}{\sigma} + \kappa(\sigma)\big(\hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma|\mathbf{c}) - \hat{\mathbf{x}}_0(\mathbf{x}_\sigma, \sigma)\big), \tag{8}$$

where $\kappa(\sigma)$ is defined piecewise on each interval $[\sigma_{t+1}, \sigma_t]$ by $\kappa(\sigma) := {}^{(\omega-\lambda)}/_{(\sigma_t - \sigma_{t+1})}$. This provides a schedule-dependent continuous-time (ODE) interpretation of CFG++ and the unified update, and enables implementing the same guidance mechanism with alternative ODE solvers (e.g., Heun's method). Nonetheless, $\kappa(\sigma)$ is defined using the discrete σ-grid and as such the resulting dynamics are not a continuous-time diffusion process in the strict sense.

3 The UniGuide Framework

Building on the guided ODE formulation in Eq. (8), UniGuide treats guidance selection as a sequential decision problem along the denoising trajectory. At each noise level σ_t, the policy observes a summary of the current diffusion state, sets the guidance parameters (ω, λ) to (ω_t, λ_t), and advances to noise level σ_{t-1} via the ODE solver update. This allows the guidance parameters (ω_t, λ_t) to vary across timesteps, yielding a trajectory-dependent adaptive policy rather than a fixed schedule.

POMDP formulation. In particular, we formulate guidance selection as a *multi-objective, partially observed Markov decision process* (POMDP). The POMDP formalism naturally captures the rich structure of the problem: the policy is stochastic during training, observes only a compact summary of the full latent state, and is conditioned on a continuous preference vector. Although in this work we integrate the probability-flow ODE deterministically (i.e., without noise injection), the POMDP formulation readily extends to SDE-based samplers.

For a conditioning input $\mathbf{c}$ and a noise schedule, the environment latent state at step t is

$$z_t \coloneqq \Big(X_{\sigma_t},\ \sigma_t,\ \sigma_{t-1},\ \hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t|\mathbf{c}),\ \hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t)\Big),$$

which includes the current noisy sample, the current and next noise levels, and the conditional and unconditional denoiser predictions. Given an action a_t, the next state z_{t-1} is obtained by applying the guided ODE update defined in Eq. (7) with a numerical solver. We denote this transition kernel by $p(z_{t-1} \mid z_t, a_t, \mathbf{c})$. Since we restrict our focus to deterministic ODE integration, throughout this paper, the transition kernel is a Dirac delta, i.e., z_{t-1} is a deterministic function of $(z_t, a_t, \mathbf{c})$.

Since the full latent state z_t can be very high-dimensional, the policy instead acts on a low-dimensional observation $o_t \coloneqq \phi(z_t) \in \mathbb{R}^d$ that captures guidance-relevant statistics such as noise level, guidance magnitude, and denoiser agreement. Although ϕ does not take $\mathbf{c}$ as an explicit input, o_t depends on $\mathbf{c}$ implicitly through the conditional and unconditional denoiser predictions contained in z_t. At each step t, we allow the policy to condition on the full observation history $(o_t, \dots, o_T)$.

Multi-objective rewards and preferences. We consider a multi-objective setting in which each completed rollout is evaluated by M reward functions on the final decoded sample, yielding the reward vector $r(X_0, \mathbf{c}) \coloneqq \big[r^{(1)}(X_0, \mathbf{c}), \dots, r^{(M)}(X_0, \mathbf{c})\big]$. Given a preference vector $\mathbf{p} \in \Delta^{M-1} \coloneqq \{\mathbf{p} \in \mathbb{R}^M_+ : \sum_k \mathbf{p}_k = 1\}$, the scalarized training objective is

$$R(X_0, \mathbf{c}; \mathbf{p}) = \sum_{k=1}^{M} \mathbf{p}_k\, r^{(k)}(X_0, \mathbf{c}). \tag{9}$$

We learn a preference-conditioned policy $\pi_\theta(a_t \mid o_{1:t}, \mathbf{p})$ that maps the observation history and the preference vector to a distribution over guidance parameters (ω_t, λ_t). By conditioning on $\mathbf{p}$, a single trained policy parameterizes the Pareto front over competing objectives, allowing the trade-off to be adjusted at inference time without retraining.

Given the dynamics, observations, and policy above, the joint trajectory distribution factorizes as

$$p_{\pi_\theta}(z_{0:T}, o_{1:T}, a_{1:T} \mid \mathbf{c}, \mathbf{p}) = p(z_T) \prod_{t=1}^{T} p(o_t \mid z_t, \mathbf{c})\, \pi_\theta(a_t \mid o_{1:t}, \mathbf{p})\, p(z_{t-1} \mid z_t, a_t, \mathbf{c})\ .$$

Crucially, because each guided update modifies the latent on which all subsequent denoising steps are built, guidance choices early in the trajectory can have compounding effects on the final sample.

State representation. At step t, we compute the observation map ϕ from quantities available in the state z_t. We instantiate ϕ as an 8-dimensional feature vector that captures solver context, guidance magnitude, alignment between conditional and unconditional denoisers, and temporal changes in guidance effects. Full definitions are provided in Appendix B.1.

To motivate our construction of ϕ, we note that its features may be grouped into three interpretable categories. First, as prior work has shown that optimal guidance strength varies with the noise level (Xi et al., 2024; Kynkäänniemi et al., 2024; Jin et al., 2025), we capture the *solver context* via the current and next noise levels (σ_t, σ_{t-1}). Second, *guidance magnitude* is represented by the norm of the conditional-unconditional discrepancy $\|\hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t|\mathbf{c}) - \hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t)\|$ and its log/SNR variants, which have been identified as indicators of guidance sensitivity (Chidambaram et al., 2024; Sadat et al., 2025a; Karras et al., 2022). Finally, *directional agreement* between the conditional and unconditional denoisers is captured via their cosine similarity, the scalar projection, and the norm of the orthogonal residual, which together measure how consistently the two denoisers align along the guidance direction (Castillo et al., 2025; He et al., 2023; Kwon et al., 2025). We additionally include temporal change features that track how guidance effects evolve across the trajectory, following observations that guidance dynamics are non-stationary (Papalampidi et al., 2025; Jin et al., 2025). While prior methods use these signals to design fixed guidance schedules, UniGuide exposes them as input features and learns how to act on them end-to-end.

Policy construction. We leverage a recurrent policy network that processes the full history $o_{t:T} := (o_t, \ldots, o_T)$ to output guidance actions. The policy is explicitly conditioned on the preference vector $\mathbf{p}$, allowing a single trained model to sweep the Pareto front of competing objectives at inference time without retraining.

4 UniGuide Training

We now detail the training procedure for optimizing the preference-conditioned policy using a variation of Group Relative Policy Optimization (GRPO) (Shao et al., 2024) adapted for our setting. A single step of training procedure is summarized in Algorithm 1, which we describe below.

Algorithm 1 Preference-Conditioned Training for Guidance Control

Require: number of groups N; group size S; reward models $\{R^{(k)}\}_{k=1}^{M}$; policy π_θ.
Initialize N groups $\{\mathcal{G}_j\}_{j=1}^{N}$ by sampling a unique context $\mathbf{c}_j$ and preference $\mathbf{p}_j$ for each.
for $j = 1$ **to** N **do**
  for $s = 1$ **to** S **do**
    Initialize $X_{\sigma_T} \sim \mathcal{N}(0, \sigma_T^2 I)$.
    for $t = T$ **to** 1 **do**
      Compute $o_t = \phi(z_t)$ and sample $a_t \sim \pi_\theta(\cdot \mid o_{t:T}, \mathbf{p}_j)$.
      Apply guided solver update to obtain $X_{\sigma_{t-1}}$.
    end for
  end for
end for
Build per-timestep rewards $r_t^{(k)}$.
Compute per-objective GAE advantages $A_t^{(k)}$ (Eq. 10) and normalize over the batch.
Compute scalarized advantages $\bar{A}_t = \sum_k p_k A_t^{(k)}$ using the assigned preference vector $\mathbf{p}_j$.
Perform group-relative normalization on $\bar{A}_t$ within each context group.
Update θ using the training loss in Eq. (12).
Output: updated parameters θ.

Training loop. At each iteration, we sample a batch of conditioning contexts and assign a preference vector $\mathbf{p} \in \Delta^{M-1}$ to each context group. A *group* $\mathcal{G}$ is defined as the set of generated trajectories that share the same conditioning context and preference vector. This grouping is essential for computing distributional rewards.

Each rollout starts from Gaussian noise at σ_T and proceeds for T steps. Upon completion, we obtain a final sample X_{σ_0}. However, relying solely on sparse terminal feedback on X_{σ_0} makes credit assignment difficult. To address this, we leverage the denoiser's internal estimate of the clean data, $\hat{X}_0(z_t)$, available at every timestep t. By evaluating rewards on these intermediate predictions, we mirror effective strategies from recent diffusion RL methods (Zhang et al., 2024; Shao et al., 2025; Xie et al., 2025) to provide dense supervision throughout the trajectory.

We employ a set of M reward models $\{R^{(k)}\}_{k=1}^{M}$. In our main experiments (Sec. 6), we instantiate these to capture semantic alignment and batch diversity, but the framework supports any differentiable or non-differentiable metric. Each model $R^{(k)} : \mathcal{G} \to \mathbb{R}^{|\mathcal{G}|}$ maps the entire group of generated samples to a vector of scalar rewards, one for each sample in the group. This formulation allows us to include both per-sample metrics (e.g., alignment scores) and distributional metrics (e.g., diversity), where the reward for an individual sample depends on its relation to the rest of the group. These M objectives are then scalarized via linear preference weighting (Eq. 9).

Advantage computation. We adopt a critic-free training scheme inspired by GRPO (Shao et al., 2024). Specifically, we optimize a clipped surrogate policy-gradient loss using advantages derived directly from rewards, without learning a value function. To balance computational cost with learning signal quality, we evaluated the reward models on the dense predictions $\hat{X}_0(z_t)$ at a subset of timesteps $\mathcal{T}_{\text{dense}} \subset \{1, \ldots, T\}$ and zero-fill the remaining steps to form the full sequence $r_t^{(k)}$. We

then compute the Generalized Advantage Estimation (GAE) for each objective,

$$A_t^{(k)} = \sum_{\ell=0}^{T-1-t} \gamma_{\text{eff}}^{\ell} r_{t+\ell}^{(k)}, \tag{10}$$

To handle varying reward scales, we normalize each objective's advantages $A_t^{(k)}$ across the entire batch: $\tilde{A}_t^{(k)} = (A_t^{(k)} - \mu_k)/\sigma_k$, where μ_k and σ_k are the batch statistics of the k-th reward model across all groups. Next, we scalarize these normalized advantages using the preference vector $\mathbf{p}$ assigned to the sample's group: $\bar{A}_t = \sum_k p_k \tilde{A}_t^{(k)}$. Finally, we apply group-relative normalization within each group $\mathcal{G}$ to reduce variance: $\hat{A}_t = (\bar{A}_t - \mu_{\mathcal{G}})/\sigma_{\mathcal{G}}$. In all normalization steps, we add a small constant to the divisor for numerical stability.

This group-relative normalization is RLOO-style (Kool et al., 2019) in spirit because both methods reduce variance by centering each sample's signal with respect to other samples in the same group; instead of using group mean and variance normalization, RLOO leverages a leave-one-out group-mean baseline $A_i^{\text{RLOO}} = r_i - (|\mathcal{G}| - 1)^{-1} \sum_{j \neq i} r_j$.

Policy Optimization. Let $\rho_t = \pi_\theta(a_t \mid o_{T:t}, \mathbf{p})/\pi_{\theta_{\text{old}}}(a_t \mid o_{T:t}, \mathbf{p})$ be the policy likelihood ratio. We optimize a clipped surrogate objective with multiple update epochs per rollout batch:

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_t\Big[\min\Big(\rho_t \hat{A}_t,\ \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\,\hat{A}_t\Big)\Big]. \tag{11}$$

To improve representation learning, we include an auxiliary dynamics head f_ψ that predicts a subset of next-step state features given the policy's current internal representation and action a_t. Specifically, $\hat{o}_{t+1} = f_\psi(\cdot)$ estimates the transition to the next state. We minimize the mean-squared error $\mathcal{L}_{\text{dyn}} = \mathbb{E}_t[\|\hat{o}_{t+1} - o_{t+1}^{(\mathcal{I})}\|_2^2]$ between this prediction and the actual next-step features restricted to a subset of indices $\mathcal{I}$, where we exclude features that are deterministic functions of the noise schedule.

The total training objective is a weighted sum of the policy loss, entropy regularization, and the auxiliary dynamics loss:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{policy}} - \beta\,\mathbb{E}[H(\pi_\theta)] + \lambda_{\text{dyn}}\mathcal{L}_{\text{dyn}}, \tag{12}$$

where the entropy term regularizes the policy to prevent collapse, with weight β, and λ_{dyn} scales the dynamics loss. We clip gradient norms for stability.

5 Related Work

Conditional guidance and guidance magnitude control. Early conditional diffusion methods relied on classifier guidance (Dhariwal & Nichol, 2021), which improves alignment via gradients from an auxiliary classifier but requires additional training on noisy inputs. Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) removes this requirement by combining conditional and unconditional denoiser predictions at inference time and has become the standard in modern pipelines. Prior work shows that a constant guidance scale is often sub-optimal, as guidance sensitivity varies across noise levels. Noise-dependent schedules (Xi et al., 2024), annealing strategies (Yehezkel et al., 2025), and interval-limited guidance (Kynkäänniemi et al., 2024) can improve the trade-off between alignment, fidelity, and diversity, but typically rely on heuristics and manual tuning. Analyses of high-guidance regimes further document systematic failures such as oversaturation and artifacts, motivating state-dependent magnitude control (Sadat et al., 2025a).

Solver dependence and guidance placement. Guidance behavior depends on the numerical solver, and identical guidance rules can result in different update behavior under deterministic DDIM-style updates and stochastic samplers (Song et al., 2021a). Recent work interprets CFG as a predictor–corrector procedure whose effect is discretization-dependent (Bradley & Nakkiran, 2024). CFG++ makes the point at which guidance is applied explicit by applying guidance to the denoised estimate while keeping the noise prediction unconditional, reducing off-manifold artifacts and improving inversion under DDIM-style solvers (Chung et al., 2025).

Direction control and manifold-based constraints. Several methods modify the direction of guidance to reduce off-manifold drift using projection or damping rules motivated by local score geometry (He et al., 2023; Kwon et al., 2025). However, such geometric arguments are theoretically justified only locally, typically at sufficiently low noise levels, and do not generally extend across the full denoising trajectory.

Learned and adaptive guidance with multi-objective control. Recent work adapts guidance based on the sampling state rather than fixed heuristics. Galashov et al. (2025) formulates guidance selection as a learning problem, while Papalampidi et al. (2025) adjusts CFG magnitude online using internal feedback, showing that effective guidance can depend on both the prompt and the timestep. Separately, Astolfi et al. (2024) argue that conditional generation is inherently multi-objective and introduce an evaluation framework that exposes trade-offs between consistency, realism, and diversity via Pareto fronts. Other approaches explicitly target Pareto-optimal generation by dynamically combining multiple objectives during diffusion sampling (Yao et al., 2024).

Finally, reinforcement learning has been used to fine-tune diffusion models by treating denoising as a multi-step decision process (Black et al., 2023; Fan et al., 2023), but these approaches optimize model parameters for single objectives and do not address inference-time guidance design or preference-controllable trade-offs, which are the focus of our method.

6 Experiments

We evaluate UniGuide on class-conditional ImageNet-512 generation using two pretrained EDM2 backbones of different capacities—EDM2-S and EDM2-XXL (Karras et al., 2024)—operating in the latent space. All experiments use the 2nd-order Heun sampler with 32 deterministic steps and the standard EDM noise schedule ($\sigma_{\min} = 2 \times 10^{-3}$, $\sigma_{\max} = 80$, $\rho = 7$). We compare against four guidance baselines: CFG (Ho & Salimans, 2022), LI-CFG (Kynkäänniemi et al., 2024), CFG++ (Chung et al., 2025), and CFGIG (Moufad et al., 2025). All baselines share the same sampler and step count; their guidance hyperparameters are individually tuned to minimize FID and are listed in Appendix B.

Reward models. UniGuide is trained with two reward objectives (M=2). The *alignment* reward uses a pretrained ImageNet ResNet-50 classifier (Salimans et al., 2016) and returns the predicted probability of the target class; we optionally clip this score at a fixed maximum to prevent reward hacking. The *diversity* reward is a convex mixture of semantic and perceptual within-group distances: $r_{\text{div}} = 0.5\, r_{\text{CLIP}} + 0.5\, r_{\text{LPIPS}}$, where r_{CLIP} is the mean pairwise cosine distance between CLIP image embeddings (Radford et al., 2021) and r_{LPIPS} is the mean pairwise LPIPS distance (Zhang et al., 2018), both computed within each context group. All rewards are evaluated on decoded images. Diversity rewards are computed over each group (samples sharing the same class label and preference vector).

Policy architecture and training. The guidance policy processes the observation history $(o_T, \ldots, o_t)$ with a 2-layer LSTM (Hochreiter & Schmidhuber, 1997). The LSTM hidden state is modulated by the preference vector $\mathbf{p}$ via a FiLM layer (Perez et al., 2018), followed by a 4-layer MLP that outputs the mean and diagonal covariance of a Gaussian over the two-dimensional action (ω_t, λ_t). Actions are tanh-squashed and affinely rescaled to $[0, 5]$ to enforce bounded, nonnegative guidance scales; the exact Jacobian correction is included when computing log-probabilities (see Appendix B.2). We train with the clipped surrogate objective (Eq. 11) for 800 iterations (EDM2-S) or 1 000 iterations (EDM2-XXL) using Adam. For EDM2-S we use a learning rate of 10^{-4} with warmup and cosine annealing; for EDM2-XXL we use $3{\times}10^{-5}$ with a constant schedule. Each training step samples 8 (EDM2-S) or 32 (EDM2-XXL) classes with 16 samples per class, and preferences are drawn from a curriculum of evenly spaced α values. Dense rewards are evaluated at steps $\{10, 20, 31\}$. The EDM2-S policy was trained on a single NVIDIA H100 80 GB GPU; the EDM2-XXL policy on 4 NVIDIA A100-SXM4 80 GB GPUs. Full hyperparameters are given in Tables 2 and 3 (Appendix B).

Evaluation metrics. For evaluation, we report $\text{FID}_{\text{DINOv2}}$ (Heusel et al., 2017) (a DINOv2-based Fréchet distance, lower is better), Human Preference Score reward model HPSv2 (Wu et al., 2023),

and PRDC (Precision, Recall, Density, Coverage) (Naeem et al., 2020) computed in DINOv2 feature space against a fixed ImageNet reference set. Precision and density measure alignment and fidelity, while recall and coverage capture distributional diversity. We additionally report classifier alignment and the combined diversity metric on the Pareto-front plots. For UniGuide results, we report them along the α value used by the policy, where α and $1-\alpha$ are the weights assigned to classifier alignment and diversity respectively.

Table 1: Quantitative comparison on ImageNet-512 (512×512) for EDM2-S and EDM2-XXL backbones. We report $\text{FD}_{\text{DINOv2}}$ (lower is better), HPSv2, and PRDC metrics. UniGuide operates at a balanced preference (α=0.5). Bold and underline denote best and second-best, respectively. Even at this balanced operating point, UniGuide achieves the highest precision and density, indicating strong alignment.

	Quality metrics					
Algorithm	$\text{FID}_{\text{DINOv2}}\downarrow$	HPS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
EDM2-S						
CFG	85.79	0.222	0.61	0.56	0.58	0.55
LI-CFG	<u>78.24</u>	0.223	0.62	**0.60**	0.58	<u>0.56</u>
CFG++	93.45	0.222	0.60	0.54	0.57	0.52
CFGIG	**73.20**	<u>0.227</u>	<u>0.64</u>	<u>0.59</u>	<u>0.62</u>	**0.58**
UniGuide	79.34	**0.243**	**0.69**	0.36	**0.72**	0.55
EDM2-XXL						
CFG	55.65	0.224	0.67	0.66	0.70	0.65
LI-CFG	**38.41**	0.230	0.70	<u>0.68</u>	0.77	**0.70**
CFG++	47.34	0.229	<u>0.71</u>	0.61	<u>0.80</u>	0.67
CFGIG	<u>41.02</u>	<u>0.230</u>	0.70	**0.68**	0.77	**0.70**
UniGuide	47.18	**0.245**	**0.77**	0.54	**0.95**	<u>0.69</u>

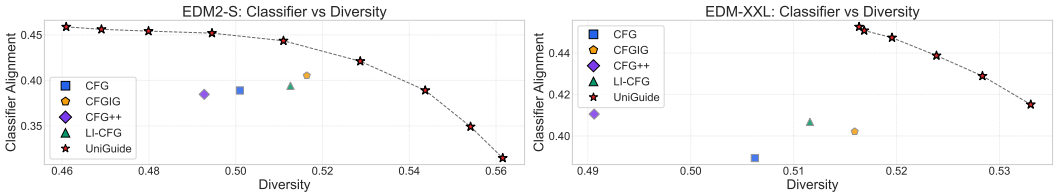


Figure 1: Alignment–diversity Pareto fronts on ImageNet-512 for EDM2-S (left) and EDM2-XXL (right). Each baseline is a single fixed operating point; UniGuide traces a continuous, preference-conditioned curve by sweeping $\alpha \in [0, 1]$. UniGuide dominates the Pareto-front for both backbones.

Preliminary results. Table 1 compares UniGuide at a balanced preference (α=0.5) against all baselines on both backbones. Even at this balanced operating point, UniGuide achieves the highest precision and density on both EDM2-S and EDM2-XXL, as well as the best HPSv2. Figure 1 shows that UniGuide's Pareto front strictly dominates all baselines across the full alignment–diversity trade-off on both backbones, while each baseline occupies only a single fixed operating point.

Figure 2 visualizes the learned guidance schedules across the 32 denoising steps. The schedules are *preference-dependent* (higher α yields stronger guidance), *non-monotonic* and *backbone-dependent*. Figure 3 illustrates this trade-off qualitatively by plotting the images across different α values

7 Conclusion

We presented UniGuide, a unified framework that casts diffusion guidance as a multi-objective reinforcement learning problem, enabling a single preference-conditioned policy to navigate the alignment–diversity Pareto front at inference time. Preliminary results on class-conditional ImageNet-512 show that UniGuide dominates standard guidance baselines across two EDM2 backbones of different capacities, while learning non-trivial guidance schedules. These results are preliminary; we note that linear scalarization only covers the convex part of the Pareto frontier, and

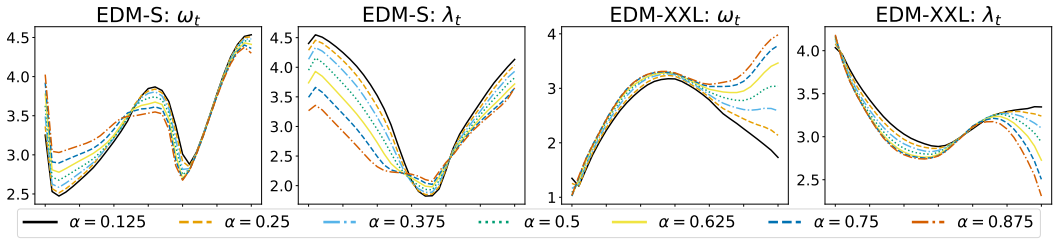


Figure 2: Learned mean guidance schedules $\bar{\omega}_t$ (left) and $\bar{\lambda}_t$ (right) across denoising steps for different preference weights α. The policy learns non-trivial, non-monotonic schedules that differ qualitatively across backbones.

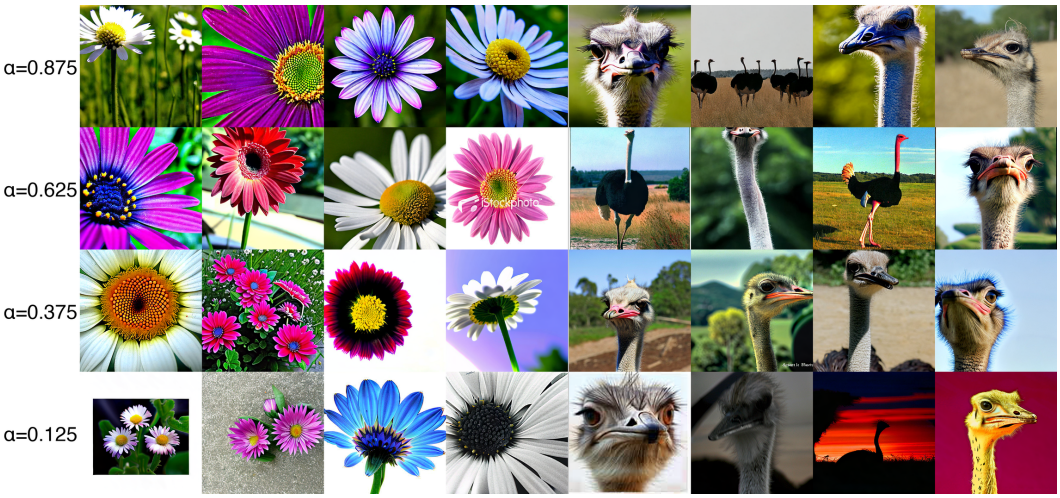


Figure 3: Effect of preference conditioning on EDM2-XXL for ImageNet classes *daisy* (left block) and *ostrich* (right block). Each row corresponds to a different preference weight α; as α decreases from 0.875 (top, high alignment) to 0.125 (bottom, high diversity), the policy shifts from highly aligned samples toward greater visual diversity, all using the same trained policy.

that extending the framework to non-linear scalarizations, text-conditioned generation, and larger numbers of reward models are promising directions for future work.

Ethics Statement

This work focuses on the alignment of generative models with preferences and constraints. While the proposed method is intended for beneficial use cases such as improving image quality and diversity, we acknowledge that steering generative models could potentially be used to generate harmful or biased content. Our experiments are conducted on standard benchmark datasets (ImageNet) and utilize existing, publicly available backbones. We believe that improving the interpretability and control of diffusion models is a key step toward their safe deployment.

Acknowledgments

The work of Aymeric Dieuleveut and Mahmoud Hegazy is supported by French State aid managed by the Agence Nationale de la Recherche (ANR) under the France 2030 program with the reference ANR-23-PEIA-005 (REDEEM project), and ANR-23-IACL-0005, in particular the Hi!Paris FLAG chair. Additionally, this project and the work of Navid Bagheri Shouraki and Eric Moulines, were funded by the European Union (ERC-2022-SYG-OCEAN-101071601). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor

the granting authority can be held responsible for them. This publication is part of the Chair Markets and Learning, supported by Air Liquide, BNP PARIBAS ASSET MANAGEMENT Europe, EDF, Orange and SNCF, sponsors of the Inria Foundation. This work was granted access to the HPC resources of IDRIS under the allocations 2025-AD011016484 and 2026-AD011017341 by GENCI.

REFERENCES

Yashas Annadani, Syrine Belakaria, Stefano Ermon, Stefan Bauer, and Barbara E Engelhardt. Preference-guided diffusion for multi-objective offline optimization. *arXiv preprint arXiv:2503.17299*, 2025.

Pietro Astolfi, Marlene Careil, Melissa Hall, Oscar Mañas, Matthew Muckley, Jakob Verbeek, Adriana Romero Soriano, and Michal Drozdzal. Consistency-diversity-realism pareto fronts of conditional image generative models, 2024. *URL https://arxiv. org/abs/2406.10429*, 2024.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*, 2024.

Angela Castillo, Jonas Kohler, Juan C Pérez, Juan Pablo Pérez, Albert Pumarola, Bernard Ghanem, Pablo Arbeláez, and Ali Thabet. Adaptive guidance: Training-free acceleration of conditional diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1962–1970, 2025.

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

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

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

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Alexandre Galashov, Ashwini Pokle, Arnaud Doucet, Arthur Gretton, Mauricio Delbracio, and Valentin De Bortoli. Learn to guide your diffusion model. *arXiv preprint arXiv:2510.00815*, 2025.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, et al. Manifold preserving guided diffusion. *arXiv preprint arXiv:2311.16424*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Cheng Jin, Qitan Shi, and Yuantao Gu. Stage-wise dynamics of classifier-free guidance in diffusion models. *arXiv preprint arXiv:2509.22007*, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! 2019.

Mingi Kwon, Jaeseok Jeong, Yi Ting Hsiao, Youngjung Uh, et al. Tcfg: Tangential damping classifier-free guidance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2620–2629, 2025.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=nAIhvNy15T.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

Dawid Malarz, Artur Kasymov, Maciej Zięba, Jacek Tabor, and Przemysław Spurek. Classifier-free guidance with adaptive scaling. 2025.

Badr Moufad, Yazid Janati, Alain Durmus, Ahmed Ghorbel, Eric Moulines, and Jimmy Olsson. Conditional diffusion models with classifier-free gibbs-like guidance. *arXiv preprint arXiv:2505.21101*, 2025.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning*, pp. 7176–7185. PMLR, 2020.

Pinelopi Papalampidi, Olivia Wiles, Ira Ktena, Aleksandar Shtedritski, Emanuele Bugliarello, Ivana Kajic, Isabela Albuquerque, and Aida Nematzadeh. Dynamic classifier-free diffusion guidance via online feedback. *arXiv preprint arXiv:2509.16131*, 2025.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

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

Seyedmorteza Sadat, Otmar Hilliges, and Romann M Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Seyedmorteza Sadat, Manuel Kansy, Otmar Hilliges, and Romann M. Weber. No training, no problem: Rethinking classifier-free guidance for diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=b3CzCCCILJ.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Yawen Shao, Jie Xiao, Kai Zhu, Yu Liu, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Anchoring values in temporal and group dimensions for flow matching model alignment. *arXiv preprint arXiv:2512.12387*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

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

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 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.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Wang Xi, Nicolas Dufour, Nefeli Andreou, Cani Marie-Paule, Victoria Fernandez Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *Transactions on Machine Learning Research*, 2024.

Shaoan Xie, Lingjing Kong, Xiangchen Song, Xinshuai Dong, Guangyi Chen, Eric P Xing, and Kun Zhang. Step-aware policy optimization for reasoning in diffusion large language models. *arXiv preprint arXiv:2510.01544*, 2025.

Yinghua Yao, Yuangang Pan, Jing Li, Ivor Tsang, and Xin Yao. Proud: Pareto-guided diffusion model for multi-objective generation. *Machine Learning*, pp. 1–28, 2024.

Shai Yehezkel, Omer Dahary, Andrey Voynov, and Daniel Cohen-Or. Navigating with annealing guidance scale in diffusion space. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Ziyi Zhang, Li Shen, Sen Zhang, Deheng Ye, Yong Luo, Miaojing Shi, Bo Du, and Dacheng Tao. Aligning few-step diffusion models with dense reward difference learning. *arXiv preprint arXiv:2411.11727*, 2024.

A Details on the unified parametrization

Unified parametrization update. Starting from the unified parameterization in Eq. (7), expanding gives

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}} X_{\sigma_{t+1}} + \Big(1-\omega-\frac{\sigma_t}{\sigma_{t+1}}(1-\lambda)\Big)\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}) + \Big(\omega-\frac{\sigma_t}{\sigma_{t+1}}\lambda\Big)\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c}). \tag{13}$$

(1) Setting $\omega = \lambda$ simplifies Eq. (13) to

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}} X_{\sigma_{t+1}} + \left(1-\frac{\sigma_t}{\sigma_{t+1}}\right)\Big[(1-\omega)\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}) + \omega\,\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c})\Big],$$

which can be written using the CFG denoiser as

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}} X_{\sigma_{t+1}} + \left(1-\frac{\sigma_t}{\sigma_{t+1}}\right)\hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c};\omega).$$

This is exactly the Euler update with the CFG denoiser.

(2) Setting $\lambda = 0$ simplifies Eq. (13) to

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}}\Big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1})\Big) + (1-\omega)\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}) + \omega\,\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c}), \tag{14}$$

which can be written using the CFG denoiser as

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}}\Big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1})\Big) + \hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c};\omega).$$

This is the CFG++ (Chung et al., 2025) update under the Euler discretization of the probability-flow ODE (Eq. (2)).

With $\lambda = 0$, the CFG++ update in Eq. (14) can be rewritten as

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}} X_{\sigma_{t+1}} + \Big(\big(1-\frac{\sigma_t}{\sigma_{t+1}}\big)-\omega\Big)\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}) + \omega\,\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c}). \tag{15}$$

By factoring out $1-\frac{\sigma_t}{\sigma_{t+1}}$ from the last two terms, we will have:

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}} X_{\sigma_{t+1}} + \left(1-\frac{\sigma_t}{\sigma_{t+1}}\right)\Big[(1-\omega_{t+1}^{++})\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}) + \omega_{t+1}^{++}\hat{\mathbf{x}}_0(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c})\Big],$$

which can be written as a CFG denoiser update with a specific guidance scale

$$X_{\sigma_t} = \frac{\sigma_t}{\sigma_{t+1}} X_{\sigma_{t+1}} + \left(1-\frac{\sigma_t}{\sigma_{t+1}}\right)\hat{\mathbf{x}}_0^{\mathsf{cfg}}(X_{\sigma_{t+1}},\sigma_{t+1}|\mathbf{c};\omega_{t+1}^{++}).$$

Matching coefficients requires

$$\left(1-\frac{\sigma_t}{\sigma_{t+1}}\right)\omega_{t+1}^{++} = \omega.$$

Therefore, CFG++ can be interpreted as a DDIM-style sampler whose update is driven by a classifier-free guided denoiser with noise-level-dependent scaling with $\omega_{t+1}^{++} = \omega\,\sigma_{t+1}/\big(\sigma_{t+1}-\sigma_t\big)$.

Heun Method. The Heun method is a second-order predictor–corrector scheme for numerically solving the probability flow ODE (2). It improves upon the Euler discretization by approximating the integral using the trapezoidal rule, averaging the vector field evaluated at the beginning and end of each step. We first define the ODE drift

$$d(\mathbf{x},\sigma) \;:=\; \frac{\mathbf{x}-\hat{\mathbf{x}}_0(\mathbf{x},\sigma)}{\sigma}.$$

Given a discretization $\sigma_{t+1} > \sigma_t$, the Heun update proceeds as follows.

- **Predictor:** Take a trial Euler step using slope $d_1 = d(X_{\sigma_{t+1}},\sigma_{t+1})$,

$$\hat{X}_{\sigma_t} = X_{\sigma_{t+1}} + (\sigma_t - \sigma_{t+1})\,d_1.$$

- **Corrector:** Estimate the slope at the predicted point, $d_2 = d(\hat{X}_{\sigma_t}, \sigma_t)$.
- **Update:** Compute the final sample using the trapezoidal rule,

$$X_{\sigma_t} = X_{\sigma_{t+1}} + \frac{\sigma_t - \sigma_{t+1}}{2}(d_1 + d_2).$$

Writing $d(\mathbf{x}, \sigma) = \big(\mathbf{x} - \hat{\mathbf{x}}_0(\mathbf{x}, \sigma)\big)/\sigma$ gives the explicit form

$$X_{\sigma_t} = X_{\sigma_{t+1}} + \frac{\sigma_t - \sigma_{t+1}}{2}\left[\frac{\hat{X}_{\sigma_t} - \hat{\mathbf{x}}_0(\hat{X}_{\sigma_t}, \sigma_t)}{\sigma_t} + \frac{X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1})}{\sigma_{t+1}}\right].$$

Adaptive guidance interpretation. We relate the unified update in Eq. (7) to a standard Euler update with an effective guidance scale. Let $r := \sigma_t/\sigma_{t+1}$ and define the guidance vector $d_{t+1} := \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}) - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1})$. Expanding Eq. (7) gives

$$\begin{aligned} X_{\sigma_t} &= \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}) + \omega d_{t+1} + r\Big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}) - \lambda_t d_{t+1}\Big) \\ &= (1-r)\hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}) + (\omega - r\lambda_t)d_{t+1} + rX_{\sigma_{t+1}}\,. \end{aligned} \tag{16}$$

A standard Euler update with scale $\tilde{\omega}$ is

$$X_{\sigma_t} = \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}) + \tilde{\omega} d_{t+1} + r\Big(X_{\sigma_{t+1}} - \hat{\mathbf{x}}_0(X_{\sigma_{t+1}}, \sigma_{t+1}) - \tilde{\omega} d_{t+1}\Big), \tag{17}$$

which expands to the same form as Eq. (16) with $(\omega - r\lambda_t)$ replaced by $\tilde{\omega}(1-r)$. Matching coefficients yields

$$\omega - r\lambda_t = \tilde{\omega}(1-r) \implies \tilde{\omega} = \frac{\omega - r\lambda_t}{1-r} = \frac{\omega - (\sigma_t/\sigma_{t+1})\lambda_t}{1 - \sigma_t/\sigma_{t+1}}\,. \tag{18}$$

So again, we can check that CFG++ ($\lambda = 0$) is equivalent to standard CFG with step-dependent scale $\tilde{\omega} = \omega\,\sigma_{t+1}/(\sigma_{t+1} - \sigma_t)$.

Heun update with unified guidance. Using the Adaptive Guidance interpretation, the derivation of the Heun sampler becomes trivial. We simply apply the standard Heun scheme, but for each step, we use the effective guidance scale $\tilde{\omega}$ calculated for that step interval. For a step $\sigma_{t+1} \to \sigma_t$:

1. **Calculate Adaptive Scale:** Compute $\tilde{\omega}$ using Eq. (18).
2. **Predictor:** Compute slope d_1 using $\tilde{\omega}$ and take trial step:

$$d_1 = \Big(\hat{X}_{\sigma_{t+1}} - \big(\hat{\mathbf{x}}_0(\hat{X}_{\sigma_{t+1}}, \sigma_{t+1}) + \tilde{\omega}(\hat{\mathbf{x}}_0(\hat{X}_{\sigma_{t+1}}, \sigma_{t+1}|\mathbf{c}) - \hat{\mathbf{x}}_0(\hat{X}_{\sigma_{t+1}}, \sigma_{t+1}))\big)\Big)/\sigma_{t+1}\,.$$

$$\tilde{X}_{\sigma_t} = \hat{X}_{\sigma_{t+1}} + (\sigma_t - \sigma_{t+1})d_1\,.$$

3. **Corrector:** Compute slope d_2 at the new point using the **same** adaptive scale $\tilde{\omega}$:

$$d_2 = \Big(\tilde{X}_{\sigma_t} - \big(\hat{\mathbf{x}}_0(\tilde{X}_{\sigma_t}, \sigma_t) + \tilde{\omega}(\hat{\mathbf{x}}_0(\tilde{X}_{\sigma_t}, \sigma_t|\mathbf{c}) - \hat{\mathbf{x}}_0(\tilde{X}_{\sigma_t}, \sigma_t))\big)\Big)/\sigma_t\,.$$

4. **Update:** $\hat{X}_{\sigma_t} = \hat{X}_{\sigma_{t+1}} + \frac{\sigma_t - \sigma_{t+1}}{2}(d_1 + d_2)$.

B Experiments Details

Following Karras et al. (2024), the noise levels $\{\sigma_t\}_{t=0}^{T-1}$ are spaced according to a power-law schedule that interpolates between σ_0 (the lowest noise level) and $\sigma_{\max}$ (the highest):

$$\sigma_t = \left(\sigma_0^{1/\rho} + \frac{t}{T-1}(\sigma_{\max}^{1/\rho} - \sigma_0^{1/\rho})\right)^{\rho}\,, \tag{19}$$

where ρ controls the density of steps across noise levels: larger ρ concentrates more steps at lower noise levels, where finer detail is resolved. We use the default EDM values σ_0=0.002, $\sigma_{\max}$=80, ρ=7.

Table 2: UniGuide policy training hyperparameters for each backbone.

Parameter	EDM2-S	EDM2-XXL
Policy architecture (shared)		
LSTM layers / hidden	2 / 256	
MLP layers / hidden	4 / 256	
Preference embedding dim	128	
FiLM conditioning	preference + state	
Action dim	2 (ω_t, λ_t)	
$[a_{\min}, a_{\max}]$	$[0,\ 5]$	
Dynamics head hidden	256	
Optimization		
Training iterations	800	1 000
Learning rate	10^{-4}	$3{\times}10^{-5}$
LR schedule	warmup 32 + cosine	constant
PPO epochs	4	
Clip ϵ	0.2	
Entropy coeff. β	0.05	
Grad norm clip	5.0	
GAE & rewards		
γ_{eff}	0.95	
λ_{dyn}	0.4	0.2
Dense reward steps	$\{10, 20, 31\}$	terminal only
Data & preferences		
Classes per step	8	32
Samples per class	16	
Compute		
GPU	1× H100 80 GB	4× A100 80 GB

Table 3: Baseline guidance hyperparameters (FID-optimized). All methods use the 2nd-order Heun sampler with 32 steps ($\sigma_{\min}$=0.002, $\sigma_{\max}$=80, ρ=7).

Method	Parameter	EDM2-S	EDM2-XXL
CFG	ω	1.4	1.2
LI-CFG	ω	2.1	2.0
	σ_{lo}	0.28	0.19
	σ_{hi}	2.9	1.61
CFG++	λ	0.35	0.35
CFGIG	ω	2.3	2.0
	ω_{init}	1.0	1.0
	init steps	12	12
	n_{gibbs} / σ_{gibbs}	2 / 2.0	2 / 1.0

B.1 State representation.

We define the guidance difference

$$d_t \coloneqq \hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t|\mathbf{c}) - \hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t).$$

Let $\langle \cdot, \cdot \rangle$ denote the Euclidean inner product and let $\|\cdot\|$ denote the Euclidean norm. For compactness, define $x_c = \hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t|\mathbf{c})$ and $x_u = \hat{\mathbf{x}}_0(X_{\sigma_t}, \sigma_t)$. The 8-dimensional observation $o_t = \phi(z_t)$ is

$$\phi_0 = -2\log(\sigma_t) \quad \phi_1 = -2\log(\sigma_{t-1}) \quad \phi_2 = \log\big(\|d_t\|\big) \quad \phi_3 = \log\big(\|d_t\|\big) - \log(\sigma_t)$$
$$\phi_4 = \tfrac{\langle x_c, x_u \rangle}{\|x_c\|\,\|x_u\|} \quad \phi_5 = \Big\langle x_u,\, \tfrac{x_c}{\|x_c\|} \Big\rangle \quad \phi_6 = \Big\| x_u - \phi_5 \tfrac{x_c}{\|x_c\|} \Big\| \quad \phi_7 = \log\big(\|d_t\|\big) - \log\big(\|d_{t-1}\|\big)$$

Here ϕ_0, ϕ_1 encode solver context (current and next noise levels), ϕ_2 is the log guidance magnitude, and ϕ_3 is a log magnitude normalized by the current noise level (an SNR proxy). The similarity

features are ϕ_4 (cosine similarity between conditional and unconditional predictions), ϕ_5 (projection of the unconditional prediction onto the conditional direction), and ϕ_6 (magnitude of the orthogonal residual after that projection). Finally, ϕ_7 captures temporal change in guidance magnitude across steps. In implementation, all norms are normalized by the square root of the dimension to improve transfer of hyperparameters across resolutions and backbones.

B.2 Policy network architecture

We implement π_θ as a 2-layer LSTM that processes the observation history $(o_t, \dots, o_T)$. The resulting hidden state is modulated by the preference vector $\mathbf{p}$ via a FiLM layer, and the conditioned representation is passed through a 4-layer MLP that outputs the mean and diagonal covariance of a Gaussian distribution over the guidance actions. Samples are tanh-squashed and affinely rescaled to enforce bounded, nonnegative scales, and the exact Jacobian correction of this transformation is included when computing log-probabilities.

Concretely, for each action dimension we sample

$$u_t \sim \mathcal{N}(\mu_\theta(o_{t:T}, \mathbf{p}), \Sigma),$$
$$a_t = \frac{\tanh(u_t) + 1}{2}(a_{\max} - a_{\min}) + a_{\min},$$

where $a_{\min} \geq 0$ and $a_{\max} > a_{\min}$ are fixed bounds for (ω_t, λ_t). We include the exact Jacobian correction induced by the tanh transform when computing $\log \pi_\theta(a_t \mid o_{T:t}, \mathbf{p})$. Concretely, if u_t is the pre-squash Gaussian sample, then

$$\log \pi_\theta(a_t \mid o_{T:t}, \mathbf{p}) = \log \mathcal{N}(u_t; \mu_\theta, \Sigma) - \sum_{i=1}^{2} \left[\log\left(\frac{a_{\max} - a_{\min}}{2} \big(1 - \tanh^2(u_{t,i})\big) \right) \right],$$

which matches the implementation in our policy.

B.3 Reward computation

We use a set of alignment and diversity rewards. All rewards are computed on decoded images normalized to $[0, 1]$. Diversity rewards are computed within each group (same context/prompt) and then assigned to all samples in that group.

Alignment rewards.

- **Classifier alignment.** Uses a pretrained ImageNet ResNet-50 and returns the predicted probability of the target class.
- **HPSv2 alignment.** Uses the HPSv2.1 text–image score computed by a pretrained ViT-H/14 model with prompt templates.

Diversity rewards. We now define the diversity reward assigned to sample i in group $\mathcal{G}$

- **LPIPS diversity.** The mean pairwise LPIPS distance within each group:

$$r_{\text{lpips},i} = \frac{1}{|g| - 1} \sum_{\substack{j \neq i \\ x_j \in \mathcal{G}}} \text{LPIPS}(x_i, x_j).$$

- **CLIP diversity.** The mean pairwise cosine distance between CLIP image embeddings within each group:

$$r_{\text{clip},i} = \frac{1}{|g| - 1} \sum_{\substack{j \neq i \\ x_j \in \mathcal{G}}} \big(1 - \cos(f_i, f_j)\big).$$

PRDC (Precision/Recall and Density/Coverage) is a distributional metric defined in a pretrained feature space (DINOv2 or InceptionV3) against a fixed ImageNet reference set. Let $\mathcal{G} = \{f_i\}_{i=1}^n$ be generated features and $\mathcal{R} = \{r_j\}_{j=1}^m$ be reference features. Let $d(\cdot,\cdot)$ denote the feature-space distance, and let $\rho_k(x;\mathcal{S})$ be the distance from x to its k-th nearest neighbor in set $\mathcal{S}$. The PRDC metrics are

$$\text{Precision}(\mathcal{G},\mathcal{R}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[d(f_i,\mathcal{R}) \le \rho_k(f_i;\mathcal{R})]\,,$$

$$\text{Recall}(\mathcal{G},\mathcal{R}) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}[d(r_j,\mathcal{G}) \le \rho_k(r_j;\mathcal{G})]\,,$$

$$\text{Density}(\mathcal{G},\mathcal{R}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{k}\sum_{j=1}^{m} \mathbb{1}[d(f_i,r_j) \le \rho_k(r_j;\mathcal{R})]\,,$$

$$\text{Coverage}(\mathcal{G},\mathcal{R}) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}\Big[\min_i d(r_j,f_i) \le \rho_k(r_j;\mathcal{R})\Big]\,.$$

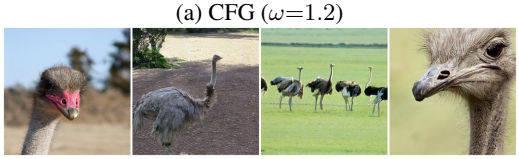
Here, precision is the fraction of generated samples that fall within the reference manifold; recall is the fraction of reference samples covered by generated ones. Density measures local concentration of generated samples around reference neighborhoods, and coverage measures how broadly generated samples span the reference manifold.

C Additional Results

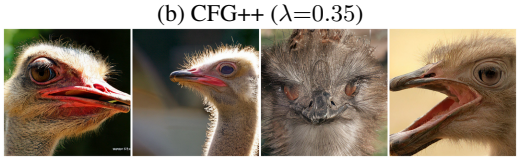
C.1 Baseline comparison on EDM2-XXL

(a) CFG (ω=1.2)

(b) CFG++ (λ=0.35)



(c) LI-CFG (ω=2.0)



(d) CFGIG (ω=2.0)

Figure 4: Baseline samples on EDM2-XXL for ImageNet class 9 (*ostrich*).

(a) CFG (ω=1.2)

(b) CFG++ (λ=0.35)

(c) LI-CFG (ω=2.0)

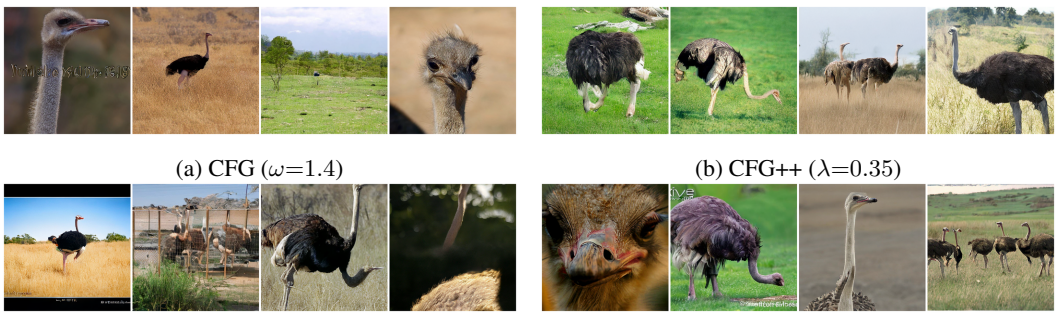
(d) CFGIG (ω=2.0)

Figure 5: Baseline samples on EDM2-XXL for ImageNet class 11 (*goldfinch*).

(a) CFG (ω=1.2) (b) CFG++ (λ=0.35)

(c) LI-CFG (ω=2.0) (d) CFGIG (ω=2.0)

Figure 6: Baseline samples on EDM2-XXL for ImageNet class 94 (*hummingbird*).



(a) CFG (ω=1.4) (b) CFG++ (λ=0.35)

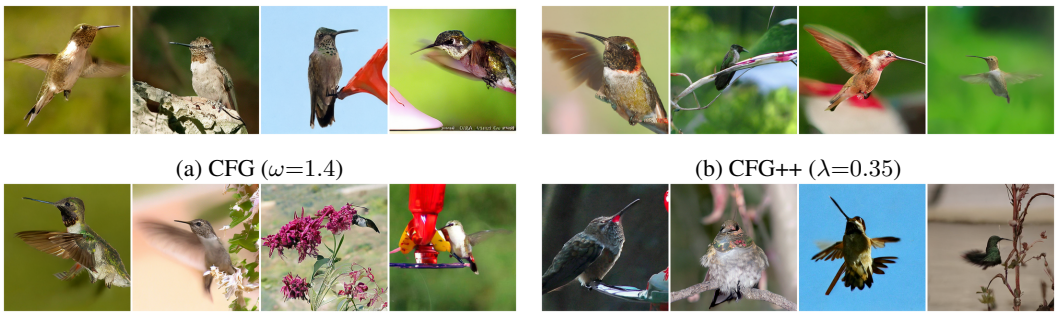
(c) LI-CFG (ω=2.1) (d) CFGIG (ω=2.3)

Figure 7: Baseline samples on EDM2-S for ImageNet class 9 (*ostrich*).

(a) CFG (ω=1.4) (b) CFG++ (λ=0.35)

(c) LI-CFG (ω=2.1) (d) CFGIG (ω=2.3)

Figure 8: Baseline samples on EDM2-S for ImageNet class 11 (*goldfinch*).



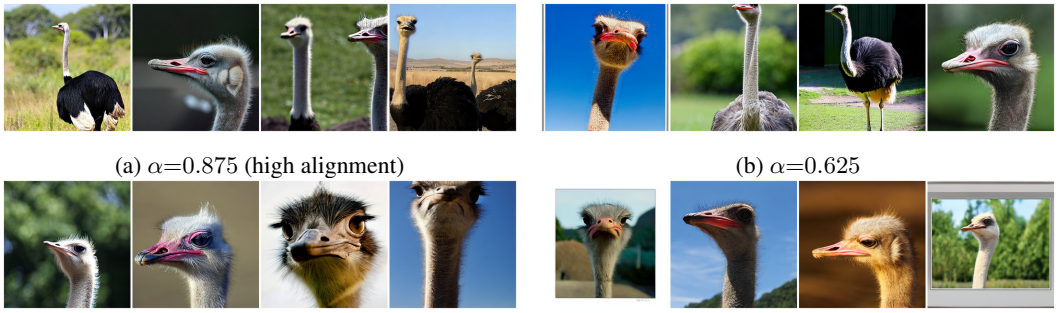
(a) CFG (ω=1.4) (b) CFG++ (λ=0.35)

(c) LI-CFG (ω=2.1) (d) CFGIG (ω=2.3)

Figure 9: Baseline samples on EDM2-S for ImageNet class 94 (*hummingbird*).

C.2 BASELINE COMPARISON ON EDM2-S

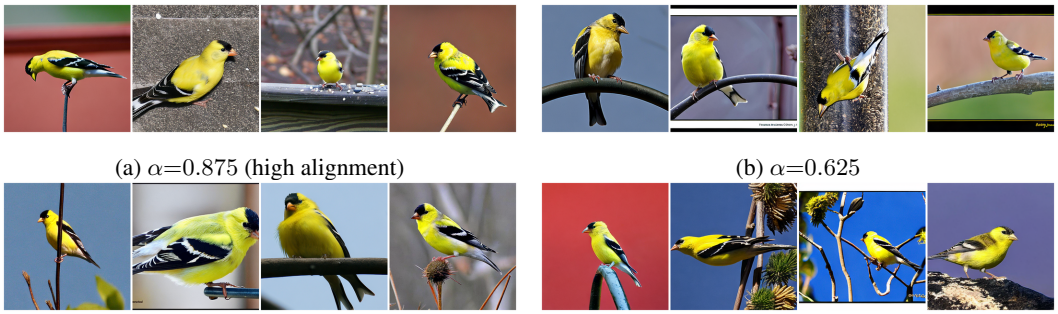
C.3 UNIGUIDE PREFERENCE SWEEPS ON EDM2-XXL



(a) α=0.875 (high alignment) (b) α=0.625

(c) α=0.375 (d) α=0.125 (high diversity)

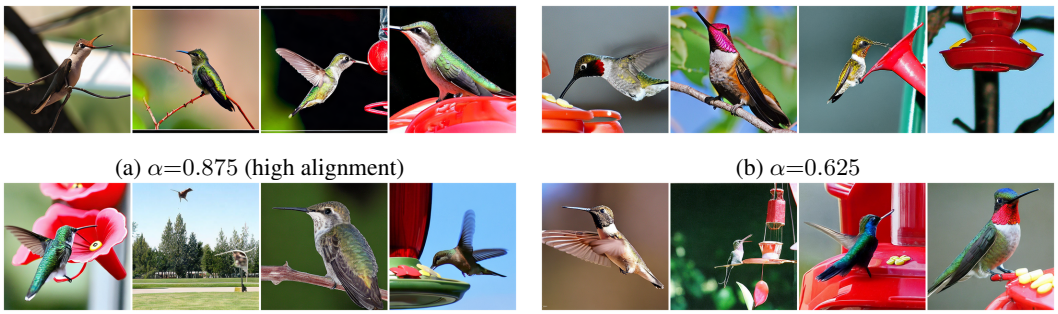
Figure 10: UniGuide preference sweep on EDM2-XXL for ImageNet class 9 (*ostrich*).



(a) α=0.875 (high alignment) (b) α=0.625

(c) α=0.375 (d) α=0.125 (high diversity)

Figure 11: UniGuide preference sweep on EDM2-XXL for ImageNet class 11 (*goldfinch*).

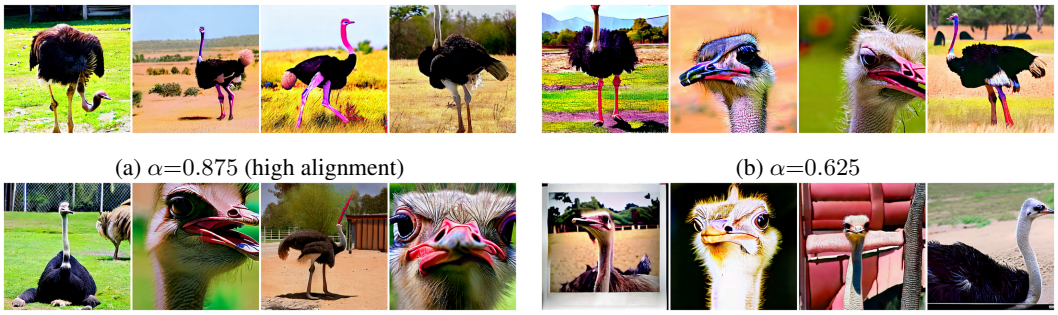


(a) α=0.875 (high alignment) (b) α=0.625

(c) α=0.375 (d) α=0.125 (high diversity)

Figure 12: UniGuide preference sweep on EDM2-XXL for ImageNet class 94 (*hummingbird*).

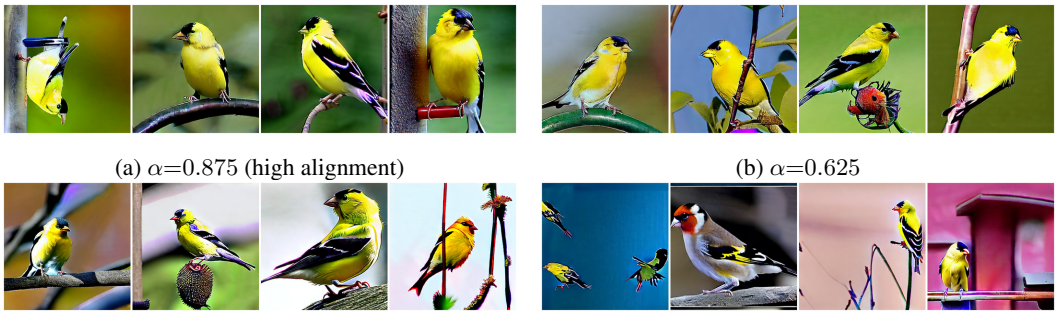
C.4 UNIGUIDE PREFERENCE SWEEPS ON EDM2-S



(a) α=0.875 (high alignment) (b) α=0.625

(c) α=0.375 (d) α=0.125 (high diversity)

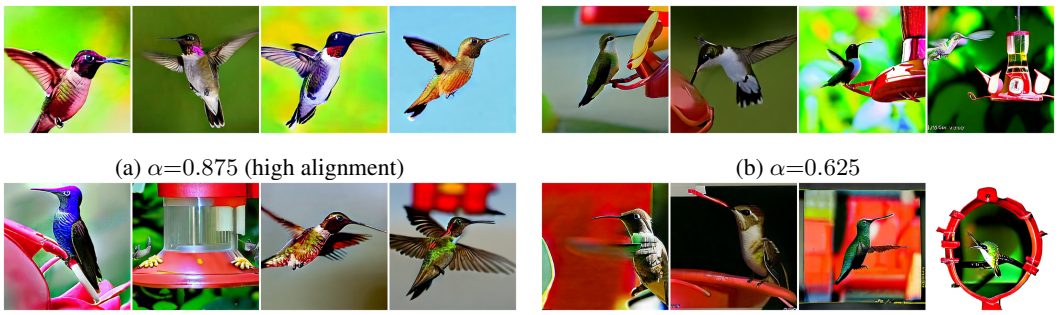
Figure 13: UniGuide preference sweep on EDM2-S for ImageNet class 9 (*ostrich*).



(a) α=0.875 (high alignment) (b) α=0.625

(c) α=0.375 (d) α=0.125 (high diversity)

Figure 14: UniGuide preference sweep on EDM2-S for ImageNet class 11 (*goldfinch*).



(a) α=0.875 (high alignment) (b) α=0.625

(c) α=0.375 (d) α=0.125 (high diversity)

Figure 15: UniGuide preference sweep on EDM2-S for ImageNet class 94 (*hummingbird*).