ADVERSARIAL LEARNING OF CLASSIFIER-FREE GUIDANCE SCHEDULES

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern text-to-image diffusion models rely on classifier-free guidance (CFG) to achieve high image fidelity and text alignment. However, CFG typically applies a static, global scale across all timesteps, samples, and conditions — a choice that is generally suboptimal and can introduce artifacts, as different states may benefit from different levels of guidance. While time-varying schedules are known to improve quality, designing them by hand is non-trivial and application-dependent. In this paper, we learn the guidance schedule as a function of diffusion time, conditioning and the current noisy sample, in order to better align sampled images with the text prompt. We frame this as a density ratio estimation problem: a discriminator is trained to estimate the time-dependent log-density ratio between the true and guided marginal distributions, while a lightweight generator network predicts the optimal, state-dependent guidance scale. Empirically, our approach outperforms both heuristic CFG schedules and prior methods for learning dynamic guidance on text-to-image generation benchmarks.

1 INTRODUCTION

Diffusion models (Song & Ermon, 2019; Song et al., 2021a;b) and flow matching models (Lipman et al., 2023) have emerged as state-of-the-art generative models for high-fidelity image and video generation. The impressive practical performance of these models is largely driven by the use of Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) during the sampling process.

For flow matching, CFG replaces the conditional velocity approximation $v_\theta(x_t, t, c)$ with a linear combination of conditional and unconditional approximations; *i.e.,* $(1+\omega)v_\theta(x_t, t, c) - \omega v_\theta(x_t, t, \varnothing)$, where ω is the guidance weight. A value of $\omega = 0$ corresponds to conditional sampling, while $\omega = -1$ corresponds to the unconditional one. In practice, most of modern text-to-image (T2I) applications employ a constant guidance weight $\omega \approx 7.5$ (Rombach et al., 2022). Using such high guidance weight increases prompt alignment (i.e. CLIP score) and improves image quality.

Relying on a constant guidance weight ω can lead to artifacts and over-saturation in the generated images (Saharia et al., 2022; Kynkäänniemi et al., 2024; Sadat et al., 2024). Manually designed dynamical guidance schedules (Kynkäänniemi et al., 2024; Wang et al., 2024) have been shown to mitigate these issues. However, these schedules introduce hyperparameters that can be difficult to tune across datasets, and are typically agnostic to the specific conditioning, applying the same guidance scale regardless of the complexity of text prompt or the specific class label.

Recently, efforts have shifted towards learning optimal guidance schedules that adapt to different conditioning information, thereby offering greater flexibility. Galashov et al. (2026) demonstrate that it is possible to learn time- and conditioning-dependent guidance weights that improve the Fréchet Inception Distance (FID) (Heusel et al., 2017) relative to constant CFG and manually designed CFG schedule baselines while maintaining comparable text alignment. Their approach is motivated by *marginal consistency condition* which encourages marginals of the true noising and the guided denoising distributions to match. Their practical implementation, however, relied on a stronger (in a sense that it implies *marginal consistency*), *self-consistency* objective which is while stable to train, failed to surpass constant CFG in text alignment.

In this paper, we shift our focus from satisfying the strict self-consistency condition to the weaker *marginal consistency* condition. Unlike prior work that operates on raw latents via an energy distance

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.

kernel (Székely & Rizzo, 2004) with maximum mean discrepancy (MMD) (Gretton et al., 2012), we employ an adversarial framework to learn the guidance schedule by matching the marginal distribution of the guided trajectories to that of the true data distribution at every timestep.

We demonstrate that this condition can be formulated as an objective that involves density ratio estimation, which naturally fits into a generative adversarial network (GAN) framework (Goodfellow et al., 2014). In our setup, a discriminator estimates the log-density ratio between real and classifier-free guided samples, while a generator learns to predict guidance scales that maximize this ratio. By optimizing for marginal consistency across the sampling trajectory, our method achieves good prompt adherence as well as sample quality, outperforming constant guidance and learned guidance baselines in T2I benchmarks.

Our main contributions are summarized as follows. First, we propose a method to learn guidance schedules based on the marginal consistency condition, encouraging the distribution of guided samples to match the true data distribution at every timestep along the sampling trajectory. Second, we use an adversarial framework to estimate density ratios and learn dynamic time-, conditioning-, and noisy sample-dependent guidance scales. Third, we show that our method outperforms constant and manually selected guidance schedule baselines as well as prior learned schedules on T2I benchmarks.

2 Background

Notation. Following the flow matching (FM) convention (Lipman et al., 2023), our framework uses continuous time $t \in [0, 1]$, where $t = 0$ is pure noise and $t = 1$ is data. Denoising steps proceed from a source timestep s to a target timestep t ($s < t$). Here, p_t denotes the true marginal distribution of the noised data. We also use $p_{t|s}$ to denote the conditional distribution $p_{t|s}(x_t|x_s)$ for any s and t; $p_{1|c}$ to denote $p(x_1|c)$; and $p_{1,c}$ to denote the joint distribution $p(x_1, c)$. We write $p_t^{s,\boldsymbol{\omega}}(x_t)$ for the marginal distribution of the guided particles, obtained by taking a sampling step from time s to t using the dynamically guided vector field (see definitions below). We use the bold notation $\boldsymbol{\omega}$ to denote the schedule of the guidance weights. Finally, $\mathcal{N}(\mu, \Sigma)$ denotes the Gaussian distribution with mean μ and covariance Σ, and $\mathcal{N}(x; \mu, \Sigma)$ represents its density evaluated at x.

Flow matching. The goal of conditional generative modeling is to sample from a target conditional distribution $p_{1|c}$ on $\mathbb{R}^d$, where $c \sim p(c)$ is a conditioning signal (e.g., a text prompt). To achieve this, Flow matching (Lipman et al., 2023) constructs a probability path connecting a standard Gaussian $p_0 = \mathcal{N}(0, \mathrm{Id})$ at $t = 0$ to the data distribution $p_1 = p_{\mathrm{data}}$ at $t = 1$ by using

$$x_t = (1 - t)\,\varepsilon + t\,x_1, \quad \varepsilon \sim \mathcal{N}(0, \mathrm{Id}), \quad x_1 \sim p_{1|c}. \tag{1}$$

This induces the conditional distribution $p_{t|1}(x_t|x_1) = \mathcal{N}(x_t;\, t\,x_1,\, (1 - t)^2\mathrm{Id})$. The conditional velocity field generating this path is $u_t(x_t|x_1) = x_1 - \varepsilon$. A neural network $v_\theta(x_t, t, c)$ is trained to approximate the marginal velocity field by minimizing the following objective

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\,(x_1,c)\sim p_{1,c},\,\varepsilon\sim\mathcal{N}(0,\mathrm{Id})}\left[\left\|v_\theta(x_t, t, c) - (x_1 - \varepsilon)\right\|^2\right]. \tag{2}$$

Conditional sampling. At inference time, the samples are generated by integrating the following the ordinary differential equation (ODE) with learned velocity field starting from $x_0 \sim \mathcal{N}(0, \mathrm{Id})$,

$$\mathrm{d}x_t = v_\theta(x_t, t, c)\mathrm{d}t, \tag{3}$$

from $t = 0$ to $t = 1$. Using Euler solver and time discretization $0 = t_0 < t_1 < \cdots < t_N = 1$, we get

$$x_{t_{k+1}} = x_{t_k} + (t_{k+1} - t_k)\, v_\theta(x_{t_k}, t_k, c), \quad k = 0, \ldots, N - 1. \tag{4}$$

For a single step from time s to time $t > s$, the update is $x_t = x_s + (t - s)\, v_\theta(x_s, s, c)$. The predicted clean data from a noisy sample x_t can be recovered as

$$\hat{x}_1(x_t, t, c) = x_t + (1 - t)\, v_\theta(x_t, t, c). \tag{5}$$

Classifier-Free Guidance (CFG). Let $v_\theta(x_t, t, \varnothing)$ be the unconditional velocity (trained by randomly dropping the conditioning) and ω is the guidance weight. Classifier-Free Guidance (Ho & Salimans, 2022) replaces the conditional velocity with the guided velocity

$$v_\theta^{\mathrm{cfg}}(x_t, t, c; \omega) = v_\theta(x_t, t, c) + \omega\,(v_\theta(x_t, t, c) - v_\theta(x_t, t, \varnothing)), \tag{6}$$

$\omega = -1$ yields unconditional sampling, $\omega = 0$ conditional sampling, while $\omega > 0$ amplifies the conditioning. Typically, $\omega \approx 7.5$ is standard for T2I applications (Saharia et al., 2022; Rombach et al., 2022). However, constant ω across time and conditioning may lead to artifacts such as over-saturation (Kynkäänniemi et al., 2024; Sadat et al., 2024).

Dynamic guidance schedules. A natural extension of CFG is to allow the guidance weight ω to vary across the sampling trajectory, i.e., $\omega = \omega(t)$. Several heuristic schedules have been proposed, including limiting guidance to a time interval (LIG; Kynkäänniemi et al. 2024), clamp-linear schedules (CLG; Wang et al. 2024), and time-dependent rescaling (Sadat et al., 2025). While these can improve over constant guidance, they still apply the same schedule globally across all samples and conditionings, and require manual tuning for each model and application.

Learnable guidance schedules and marginal consistency. Rather than hand-designing a schedule, one can *learn* guidance weights $\omega_{c,(s,t)}$ as functions of the conditioning c and the timesteps s, t with $s < t$ (time t is the target and time s is the source time). One could attempt to learn ω with guided velocity $v_\theta^{\text{cfg}}(x_s, s, c; \omega)$ equation 6 from equation 2. However, this will necessarily lead to $\omega = 0$ since v_θ is already a minimizer of equation 2. Recent work in learned guidance has explored enforcing *consistency conditions* to learn ω. Let the true marginal at time t be given by

$$p_t(x_t) = \int p_{t|1}(x_t|x_1)\, p_{1,c}(x_1, c)\, \mathrm{d}x_1\, \mathrm{d}c. \tag{7}$$

Sampling $(x_1, c) \sim p_{1,c}$, noising to time s via $x_s \sim p_{s|1}(\cdot|x_1)$, and taking a guided step from s to t using (equation 6) with weight ω, will lead to a *guided marginal*

$$p_t^{s,\boldsymbol{\omega}}(x_t) = \int \left[\int p_{t|s,c}^{(\theta,\boldsymbol{\omega})}(x_t|x_s, c)\, p_{s|1}(x_s|x_1)\, \mathrm{d}x_s \right] p_{1,c}(x_1, c)\, \mathrm{d}x_1\, \mathrm{d}c, \tag{8}$$

where $p_{t|s,c}^{(\theta,\boldsymbol{\omega})}(x_t|x_s, c)$ is the guided transition kernel from time s to t; for a deterministic Euler solver, $p_{t|s,c}^{(\theta,\boldsymbol{\omega})}(x_t|x_s, c) = \delta\Big(x_t - x_s - (t-s)\, v_\theta^{\text{cfg}}(x_s, s, c;\, \boldsymbol{\omega})\Big)$. We say that the guided process satisfies *marginal consistency* if, for all $0 \le s < t \le 1$,

$$p_t^{s,\boldsymbol{\omega}}(x_t) \approx p_t(x_t). \tag{9}$$

Galashov et al. (2026) enforced a stronger *self-consistency* variant of equation 9—matching $p_{t|1,c}$ and $p_{t|1,c}^{s,\boldsymbol{\omega}}$ (the inner integral in equation 8)—using MMD (Gretton et al., 2012) with energy kernel (Székely & Rizzo, 2004), achieving strong FID improvements but not consistently outperforming constant guidance on text alignment metrics. They argued that the gradients of the objective based on *marginal consistency* suffer from high variance due to a marginalization over (x_1, c) and the use of the MMD with a fixed energy kernel defined on raw high dimensional data. In the next section, we propose an adversarial approach to enforce marginal consistency via density ratio estimation, yielding superior text-to-image performance.

3 Learning to Guide with Density Ratio Estimation

We introduce our method for learning the dynamic guidance weights $\boldsymbol{\omega}$ for diffusion sampling with a generative adversarial network (GAN). To enforce the *marginal consistency* condition, we derive an adversarial objective based on density ratio estimation using a learned discriminator. We assume access to a pre-trained conditional flow matching model, $v_\theta(x_t, t, c)$, and an unconditional model, $v_\theta(x_t, t, \varnothing)$.

3.1 Guidance learning objective

In standard CFG (equation 6), a static scalar ω controls the guidance strength uniformly across all timesteps and inputs. We parametrize the guidance weights to be a function of the times (s, t), the conditioning c, and the current noisy sample x_s i.e. $\boldsymbol{\omega}_{x_s,c,(s,t)}$. Intuitively, adding more information to the guidance weights might allow the model to more finely adjust the guidance weights and adapt to a specific noise trajectory.

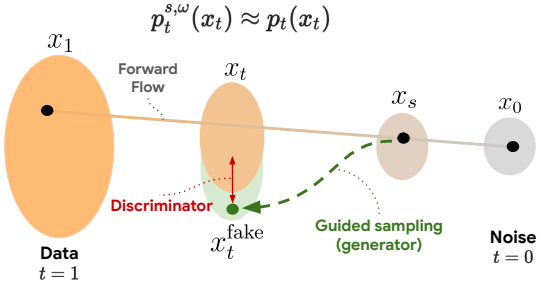


Figure 1: Marginal consistency via GAN. The first of two independent samples, x_1^1 is noised to the time t with the forward process and denotes the real samples. Second sample x_1^2 is noised to the time s and then denoised with guidance to the time t which denotes our generated path. The discriminator is then trained to discriminate between these two.

Adversarial objective. For $(s,t) \sim p(s,t)$ (see Section 3.2 for more details), the *marginal consistency* condition dictates $p_t^{s,\boldsymbol{\omega}}(x_t) \approx p_t(x_t)$, i.e. the distribution of our guided particles should match the true data distribution. This can be achieved by minimizing the Kullback-Leibler (KL) divergence

$$\mathbb{E}_{(s,t)\sim p(s,t)} \left[\mathrm{KL}\left(p_t^{s,\boldsymbol{\omega}}(x_t) \| p_t(x_t) \right) \right]. \tag{10}$$

This divergence can be rewritten in terms of the density ratio $r_t^{\boldsymbol{\omega}}(x_t) = p_t(x_t)/p_t^{s,\boldsymbol{\omega}}(x_t)$ as:

$$\mathrm{KL}\left(p_t^{s,\boldsymbol{\omega}}(x_t) \| p_t(x_t) \right) = -\mathbb{E}_{p_t^{s,\boldsymbol{\omega}}} \left[\log r_t^{\boldsymbol{\omega}}(x_t) \right]. \tag{11}$$

We train a generator to predict $\boldsymbol{\omega}$ to minimize this divergence and a discriminator to estimate this log-density ratio, providing training signal to the generator. The approach is illustrated in Figure 1.

Discriminator. We define a discriminator $d_\phi(x_t, s, t, c)$ with parameters ϕ which needs to distinguish between true marginal samples $x_t^{\text{real}} \sim p_t(x_t)$ and guided marginal samples $x_t^{\text{fake}}(\boldsymbol{\omega}) \sim p_t^{s,\boldsymbol{\omega}}(x_t)$. We optimize this via a standard binary cross-entropy loss (minus sign due to maximization):

$$\mathcal{L}_{disc}(\phi) = -\left\{ \mathbb{E}_{p_t} \left[\log \sigma(d_\phi) \right] + \mathbb{E}_{p_t^{s,\boldsymbol{\omega}}} \left[\log(1 - \sigma(d_\phi)) \right] \right\}, \tag{12}$$

where $\sigma(\cdot)$ is the sigmoid function. At convergence, the optimal discriminator $d^\star$ is a Bayes classifier. Extracting the logit from this optimal discriminator directly recovers the log-density ratio:

$$d^\star(x_t, s, t, c) = \log \left(\frac{\sigma(d^\star)}{1 - \sigma(d^\star)} \right) = \log \left(\frac{p_t(x_t)}{p_t^{s,\boldsymbol{\omega}}(x_t)} \right) = \log r_t^{\boldsymbol{\omega}}(x_t) \approx d_\phi(x_t, s, t, c). \tag{13}$$

We apply a standard R_1 gradient penalty (Mescheder et al., 2018) to the discriminator to ensure smooth ratio estimation with a constant $\gamma > 0$ (hyperparameter):

$$\mathcal{L}_{\text{reg}}(\phi) = \frac{\gamma}{2} \mathbb{E}_{x_t^{\text{real}}} [\|\nabla_{x_t} d_\phi(x_t^{\text{real}}, s, t, c)\|^2]. \tag{14}$$

The total loss for the discriminator is

$$\mathcal{L}_D(\phi) = \mathcal{L}_{disc}(\phi) + \tfrac{\gamma}{2} \mathbb{E}_{x_t^{\text{real}}} [\|\nabla_{x_t} d_\phi(x_t^{\text{real}}, s, t, c)\|^2] \tag{15}$$

Generator. We define a generator or a guidance network $\boldsymbol{\omega}^\psi_{x_s, c, (s,t)} = \boldsymbol{\omega}(x_s, s, t, c; \psi) > 0$ with parameters ψ, which outputs positive guidance weights. Employing these weights via guided velocity equation 6 specifies guided marginal distribution $p_t^{s,\boldsymbol{\omega}}(x_t)$ equation 8. We write here $x_t^{\text{fake}}(\psi) \sim p_t^{s,\boldsymbol{\omega}}$ to highlight the dependence on parameters ψ. The generator is trained by optimizing

$$\mathcal{L}_{gen}(\psi) = \mathbb{E}_{p_t^{s,\boldsymbol{\omega}}} \left[-d_\phi(x_t^{\text{fake}}(\psi), s, t, c) \right]. \tag{16}$$

To prevent the guidance scale from exploding or over-correcting out of distribution, we introduce a stabilizing L_2 penalty to the generator loss with time-dependent regularization weight $\lambda(t)$,

$$\mathcal{R}(\boldsymbol{\omega}, t) = \lambda(t) \boldsymbol{\omega}^2. \tag{17}$$

In practice, we set $\lambda(t) = at^2$, where a is a hyperparameter.

Reward-based regularization. Similar to prior work (Galashov et al., 2026), we introduce a reward-based loss term. We compute the first-order approximation of the clean data using $v_t^{\text{cfg},\boldsymbol{\omega}}$ as

$$\hat{x}_1(\boldsymbol{\omega}) = x_s + (1-s)\cdot v_t^{\text{cfg},\boldsymbol{\omega}}(x_s, s, 1, c; \psi) \tag{18}$$

We define the reward loss as the negative expectation of a practitioner-defined reward function $R(\hat{x}_1, c)$, such as the CLIP score for text-to-image models,

$$\mathcal{L}_{\text{Reward}}(\boldsymbol{\omega}) = -\mathbb{E}\left[R(\hat{x}_1(\boldsymbol{\omega}), c)\right]. \tag{19}$$

The total loss for the generator is given as

$$\mathcal{L}_{\text{G}}(\psi) = \mathcal{L}_{gen}(\psi) + \mathcal{R}(\boldsymbol{\omega}, t) + \eta\mathcal{L}_{\text{Reward}}(\boldsymbol{\omega}), \tag{20}$$

where $\eta > 0$ is a hyperparameter. In our framework, adversarial loss and regularization loss terms ensure that the samples remain on the data manifold while the reward term pushes for better alignment.

3.2 Guidance Learning Algorithm Settings

Time distribution $p(s,t)$. We use the same time distribution $p(s,t)$ as previously leveraged in Galashov et al. (2026) in our experiments, and we re-write it in a flow matching notation. The target time is given by $t \sim \mathcal{U}[\zeta+\delta,\ 1-\zeta]$, where ζ is a small constant which serves as boundary buffers that prevent optimization from encountering numerical singularities or vanishing signals found at extreme noise or pure data limits. In our experiment we set it to small constant (10^{-2}) without further tuning. The step size between the source and target timesteps is $\Delta s \sim \mathcal{U}[\delta,\ t-\zeta]$. Therefore, the source time is $s = t - \Delta s$. Similar to prior observations, we observe that $\delta = 0.1$ slightly outperforms $\delta = 0.01$, even if inference utilizes smaller steps. See Table 4 for an ablation on δ. This is likely because a larger δ provides a more robust signal to the discriminator. To ensure that the generator interpolates smoothly to the smaller step sizes utilized at inference, we make our guidance weight predictor more robust by conditioning on coordinate-invariant geometric statistics rather than raw high dimensional latents and Fourier embeddings for timesteps. See our discussion on network parametrization below. In Appendix C.9, we discuss alternative choices of time distribution.

Network Parametrization. The discriminator $d_\phi(x_t, s, t, c)$ is a convolutional network with parameters ϕ. The guidance network $\boldsymbol{\omega}(x_s, s, t, c; \psi)$ is a lightweight multi-layer perceptron (MLP) with parameters ψ. Because x_s is high dimensional latent, feeding raw latents into the MLP is computationally inefficient. We instead compute a vector of summary statistics that captures the relationship between the current state, the conditioning, and the sampling trajectory. Specifically, we extract the following statistics: 1) The log norm of the conditional velocity $\|v_\theta(x_s, s, c)\|$ and the CFG direction $\|v_\delta\|$, where $v_\delta = v_\theta(x_s, s, c) - v_\theta(x_s, s, \varnothing)$. 2) The cosine similarity between the current latent x_s and the guidance direction v_δ, calculated as $\frac{x_s \cdot v_\delta}{\|x_s\|\|v_\delta\|}$. This captures how well the current sample aligns with the intended conditioning signal. 3) The log distance to the data manifold $\log(1-s)$ and the log step size of the current jump $\log(t-s)$. The MLP takes a concatenated feature vector with these values as an input. In addition, we find it useful to feed in Fourier embeddings of s, t and $t-s$ to the generator. This helps the network interpolate smoothly to smaller step sizes utilized at inference. We apply softplus activation at the output to ensure strictly non-negative guidance weights, mirroring the standard CFG behavior. For more architectural details, see Section D.

Training strategy. The training of our guidance scheduler follows an adversarial paradigm that alternates between updating the discriminator and the generator *i.e.,*the guidance MLP. To ensure stable convergence and accurate density ratio estimation, we adopt the Two-Time-Scale Update Rule (TTUR) (Heusel et al., 2017), performing a single gradient step for each network alternately.

A key distinction in our training loop is the use of independent image-conditioning pairs, (x_1^1, c^1) and (x_1^2, c^2), drawn from the dataset $\mathcal{D}$. We use the first pair to generate the "real" marginal target x_t^{real} and the second pair to define the "fake" guided sample x_t^{fake}. This independent sampling is fundamental to enforcing marginal consistency; it prevents the discriminator from trivially learning to identify the specific noise or structure of a single trajectory. Instead, it forces the discriminator to evaluate whether the distribution of the guided particles matches the global data distribution at that specific timestep. While this independent sampling is theoretically more aligned with the marginal consistency objective, we observe in practice that the framework is robust to this choice, yielding comparable performance even when correlated pairs are used. See Table 13 for quantitative results.

Putting together the above components, we summarize our method in Algorithm 1.

Algorithm 1 Algorithm to learn parameter of CFG with Flow Matching model.

Require: Pretrained FM model v_θ, a T2I dataset, time distribution $p(s,t)$, γ discriminator regularization parameter, λ generator regularization parameter, η generator reward score parameter. Discriminator $d_\phi(x_t, s, t, c)$ with parameters ϕ and guidance network $\boldsymbol{\omega}(x_s, s, t, c; \psi)$ with parameters ψ. Learning rates α_D and α_G for discriminator and generator.

while not converged **do**

Sample times $s, t \sim p(s,t)$

(**Real path**) Sample $(x_1^1, c^1) \sim \mathcal{D}$, $\epsilon \sim \mathcal{N}(0, I)$, $x_t^{\text{real}} = (1-t)\epsilon + t x_1^1$

Sample $(x_1^2, c^2) \sim \mathcal{D}$, $z \sim \mathcal{N}(0, I)$, $x_s = (1-s)z + s x_1^2$

Compute $v_s^c = v_\theta(x_s, s, c^2)$ and $v_s^u = v_\theta(x_s, s, \varnothing)$

Compute $v_s^{\text{cfg},\boldsymbol{\omega}}(\psi) = v_s^c + \boldsymbol{\omega}(x_s, s, t, c^2; \psi)(v_s^c - v_s^u)$

(**Fake path**) $x_t^{\text{fake}}(\psi) = x_s + v_s^{\text{cfg},\boldsymbol{\omega}}(\psi)(t-s)$ ▷ Euler/DDIM ODE sampling

$\phi \leftarrow \phi - \alpha_D \nabla \mathcal{L}_D(\phi)$ ▷ Update GAN discriminator ϕ (equation 15)

$\psi \leftarrow \psi - \alpha_G \nabla \mathcal{L}_G(\psi)$ ▷ Update GAN generator ψ (equation 20)

4 Related Work

Classifier-free guidance and Variants. Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) has become the standard for conditional sampling in diffusion and flow models. A significant amount of effort has been directed towards understanding the underlying mechanics of this success. Chidambaram et al. (2024) find that CFG might fail to sample from the intended tilted distribution, instead driving the trajectory towards "archetypes" of the class in the distribution. Further analysis in high-dimensional settings (Pavasovic et al., 2025) and general data distributions (Li & Jiao, 2025) has clarified how CFG alters the geometry of the score-field. From a practical perspective, several training-free strategies have been proposed to mitigate oversaturation and artifacts associated with high constant guidance scales. CFG++ (Chung et al., 2025) introduces manifold constraints to the sampling process, while Sadat et al. (2024) decompose network predictions into parallel and orthogonal components. Autoguidance (Karras et al., 2024) rethinks the guidance direction by using a lower-quality version of the model as the unconditional reference.

Heuristic-based dynamic schedules have also seen rapid development. Kynkäänniemi et al. (2024) apply guidance on a limited interval to improve image fidelity and alignment, while Wang et al. (2024) provide an extensive analysis of guidance dynamics. We adapt clamp-linear schedule from this work as a baseline. Malarz et al. (2025) use the density of a beta distribution to modulate the guidance signal. Other training-free methods include rectified guidance (Xia et al., 2025), geo-guide Poleski et al. (2025), and adaptive guidance (Castillo et al., 2025). While these methods avoid the cost of training, they generally rely on global schedules that are agnostic to the specific semantic complexity of the conditioning.

Our work belongs to the emerging category of learned guidance schedules. Unlike Felix et al. (2025) which uses internal model predictions; *i.e.,*feedback during inference, we optimize a dedicated guidance scheduler. Closely related to our work is MMD guidance (Galashov et al., 2026). The key distinction is that we enforce a weaker marginal consistency condition instead of the strict self-consistency condition in an adversarial framework.

Another line of work that is related to our work is discriminator guidance (Kim et al., 2022; Verine et al., 2025). These methods use a discriminator as corrector to provide auxiliary score gradient, thereby increasing inference-time complexity. On the other hand, we use the discriminator strictly as a training supervisor to learn optimal guidance schedule. As our generator is a lightweight MLP, our method can achieve better alignment with the prompts while maintaining the sampling efficiency of the original backbone.

Adversarial training in Generative Modeling. Our method leverages the principles of Generative Adversarial Networks (Goodfellow et al., 2014) to learn dynamic guidance schedules. GANs were traditionally used as standard alone generative models, but later on they have also been integrated into likelihood-based settings, especially in the context of diffusion models, to enhance sampling efficiency and quality. Diffusion GAN (Wang et al., 2023) and Denoising Diffusion GAN (Xiao et al., 2022) use a discriminator to match the conditional or marginal distributions of denoising steps,

thereby speeding up sampling while maintaining visual fidelity. In this work, we do not use the adversarial loss to train the primary generative backbone. Instead, we use it to provide a training signal for the guidance scheduler.

5 EXPERIMENTS

Datasets and Models. We evaluate the performance of our method on the standard text-to-image generation task using 10K prompts from MS-COCO 2014 dataset (Lin et al., 2014) at 512×512 resolution. Our experiments utilize two sizes of a custom-trained flow matching model (Lipman et al., 2023) based on the Multimodal Diffusion Transformer backbone (Labs et al., 2025) with parameter counts of 740M (MMDiT-XS) and 1.05B (MMDiT-S). During all subsequent training of the guidance network, this remains frozen. For CFG, instead of using empty conditioning $\varnothing$ in equation 6, we use a fixed negative prompt c_{neg} = "blurred, blurry, disfigured, ugly, tiling, poorly drawn" and therefore replace the unconditional velocity term $v_\theta(x_t, t, \varnothing)$ by $v_\theta(x_t, t, c_{\text{neg}})$. We found that it led to higher quality images than $\varnothing$ and we use it for our learned models and the baselines. We provide detailed hyperparameter settings for optimizer as well as details of the model architecture in Section D.

Baselines. We evaluate our method against a diverse set of guidance strategies: 1) *Fixed Guidance*: We consider unguided ($\omega = 0$) and the standard constant CFG ($\omega = 7.5$) as our primary foundational baselines. 2) *Heuristic-based guidance schedules*: We evaluate state-of-the-art manually designed schedules, including Limited Interval Guidance (LIG) (Kynkäänniemi et al., 2024) and Clamp-linear schedules (CLG) (Wang et al., 2024). For these baselines, we perform extensive hyperparameter tuning to find the optimal ranges for MS-COCO, as detailed in Section C. 3) *Learned Self-Consistency (MMD+SC) / Marginal-Consistency (MMD+MC)*: We compare against the approach by Galashov et al. (2026), which utilizes Maximum Mean Discrepancy and an energy kernel to enforce self-consistency / marginal-consistency.

Evaluation Metrics. We employ a suite of metrics to better capture the trade-off between image fidelity and text alignment. We use FID (Heusel et al., 2017) to measure the distributional distance between generated and real images. We report the CLIP score (Hessel et al., 2021; Radford et al., 2021) to measure alignment between the generated images match the input prompts. We include LAION Aesthetics metric (Schuhmann, 2022) to capture visual quality. Finally, we also report Human Preference Score v2 (HPSv2) (Wu et al., 2023) and PickScore (Kirstain et al., 2023) to measure alignment with human preferences, and prompt adherence.

5.1 TEXT-TO-IMAGE (T2I) GENERATION

Results. We summarize the quantitative performance of our method on MS-COCO 512×512 benchmark in Table 1 for MMDiT-XS and in Table 2 for MMDiT-S. We used the same generator architecture for MMD+SC/MMD+MC baseline and we add more extensive comparisons in Appendix C.1, where we replicated the same setup as in (Galashov et al., 2026). Across both model sizes, our approach based on adversarial framework that enforces marginal consistency (GAN + MC) consistently achieves better metrics compared to the baselines in almost all alignment and human preference metrics. Specifically, our method yields superior CLIP, Aesthetic, and HPSv2 scores compared to constant guidance and manually tuned heuristic schedules.

Table 1: **MS COCO** 512×512. Performance of different methods on a small, MMDiT-XS model. We highlight the best metric in **bold**.

Method	Guidance Weight	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
Unguided	$\omega = 0$	27.65	0.2742	4.79	0.2547	0.1997
Constant	$\omega = 7.5$	29.73	0.3044	5.29	0.2817	**0.2186**
LIG	$\omega(t) = 12.5, t \in [0.1, 0.9]$	**25.57**	0.3032	5.30	0.2812	0.2171
CLG	$\omega = 15.0$	26.38	0.3038	5.26	0.2807	0.2175
MMD + SC	$\boldsymbol{\omega}(x_s, s, t, c)$	28.38	0.3042	5.26	0.2813	0.2181
MMD + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	28.11	0.3037	5.25	0.2807	0.2179
GAN + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	31.31	**0.3048**	**5.32**	**0.2829**	**0.2186**



(a) *A black dog lays his head on a book of maps.*



(b) *A side by side comparison of the same room in the past and present.*



(c) *People are cross country skiing through a field.*

Figure 2: **T2I Results on MS-COCO with MMDiT-S**. (left-to-right) We provide results of images generated from the given text prompt with CFG $\omega = 7.5$, Clamp linear guidance (CLG), Limited Interval guidance (LIG), MMD guidance, and our method. Qualitatively, our method (rightmost) exhibits better adherence to text prompts, precisely capturing nuanced details such as the dog's interaction with a book of maps in (a), the temporal dichotomy of the split room in (b), and the specific winter-field context in (c) that baselines simplify or omit.

Table 2: **MS COCO** 512×512. Performance of different methods on a large, MMDiT-S model. We highlight the best metric in **bold**.

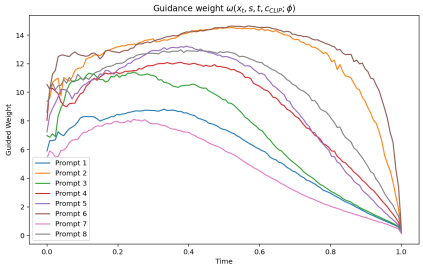
Method	Guidance Weight	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
Unguided	$\omega = 0$	24.3	0.279	4.84	0.2588	0.2024
Constant	$\omega = 7.5$	31.11	0.3063	5.31	0.285	**0.2214**
LIG	$\omega(t) = 15.0, t \in [0.1, 0.9]$	**24.95**	0.3042	5.31	0.2843	0.2194
CLG	$\omega = 14.0$	27.14	0.3055	5.27	0.2838	0.2203
MMD + SC	$\boldsymbol{\omega}(x_s, s, t, c)$	28.55	0.3067	5.29	0.2847	0.2208
MMD + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	29.43	0.3062	5.28	0.2836	0.2205
GAN + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	31.7	**0.3068**	**5.34**	**0.2856**	**0.2214**

As shown in Table 1, MMDiT-XS model benefits from our learned schedule. While the LIG achieves the lowest FID (25.57), it does so at the cost of lower alignment. Our method achieves a CLIP score of 0.3048 and an Aesthetic score of 5.32, the highest in the group. The gains observed in the smaller model consistently carry over to the larger MMDiT-S architecture as shown in Table 2. GAN + MC outperforms baselines on CLIP (0.3068), Aesthetic (5.34), and HPSv2 (0.2856).

This suggests that learning a dynamic, conditioning-dependent schedule via marginal consistency allows the model to better navigate the trade-off between image realism and adherence to complex text prompts. Our approach also demonstrates a clear advantage over self-consistency framework of Galashov et al. (2026) on alignment and human preference metrics. Finally, as observed in Tables 1 and 2, our method also results in a marginal increase in FID. We attribute this to the direction of the divergence we optimize: the generator loss $\mathbb{E}_{p_t^{s,\omega}}[-d_\phi] \approx \mathrm{KL}(p_t^{s,\boldsymbol{\omega}} \,\|\, p_t)$ is a reverse KL, which is mode-seeking and favors precision (alignment and aesthetics) over coverage, which FID penalizes.

This mirrors the standard guidance trade-off: constant CFG ($\omega = 7.5$) already degrades FID relative to unguided sampling, and Inception-based FID is known to be misaligned with human preference in high-guidance regimes (Jayasumana et al., 2024).

MMDiT-S samples are shown in Figures 2, 4 and 5. Additional samples are given in Appendix E. Our method generates images that are more realistic and better aligned with the text prompts. The learned guidance weights are shown in Figure 3. We observe high variability depending on the prompt.



Prompt 1. A black dog lays his head on a book of maps.
Prompt 2. A side by side comparison of the same room in the past and present.
Prompt 3. People are cross country skiing through a field.
Prompt 4. A person holds a flip phone displaying the screen.
Prompt 5. A cake has had a slice taken out of it and put onto the plate.
Prompt 6. A display case in front of a store filled with umbrellas.
Prompt 7. A black and white dog laying on a bed.
Prompt 8. A kitchen that has various paintings on the walls and a large fish aquarium against the wall.

Figure 3: **Learned guidance weights on MS COCO** 512×512 trained with Algorithm 1 on MMDiT-S. Please refer to Figures 2, 4 and 5 for the corresponding images.

5.2 Ablations

Conditioning Information. We ablate the inputs provided to the guidance network $\boldsymbol{\omega}$. While Galashov et al. (2026) found passing timesteps (s, t) and conditioning c was superior, Table 3 demonstrates that additionally providing state statistics of x_s consistently boosts prompt alignment and human preference scores across scales, at a slight cost to FID.

Time Distribution & Additional Studies. In Table 4, we vary δ on MMDiT-XS and find performance remains largely robust across intervals. Additional studies on sampling step counts (Section C.4), generator network architectures (Section C.5), noise coupling (Section C.6), pair sampling (Section C.7), and alternative temporal distributions (Section C.9) are detailed in the Appendix.

Table 3: **Conditioning input ablation** (MS-COCO).

Conditioning	FID ↓	CLIP ↑	Aes. ↑	HPS ↑	Pick ↑
XS, $\boldsymbol{\omega}(s, t, c)$	**26.91**	0.3037	5.27	0.2811	0.2178
XS, $\boldsymbol{\omega}(x_s, s, t, c)$	31.31	**0.3048**	**5.32**	**0.2829**	**0.2186**
S, $\boldsymbol{\omega}(s, t, c)$	**28.00**	0.3063	5.28	0.2841	0.2206
S, $\boldsymbol{\omega}(x_s, s, t, c)$	31.70	**0.3068**	**5.34**	**0.2856**	**0.2214**

Table 4: **Impact of** δ (MMDiT-XS).

δ	FID ↓	CLIP ↑	Aes. ↑	HPS ↑	Pick ↑
0.01	31.24	0.3048	5.32	0.2827	0.2189
0.10	31.31	0.3048	5.32	0.2829	0.2186
0.20	30.36	0.3049	5.31	0.2826	0.2185

6 Discussion and Limitations

We introduce an adversarial method to learn dynamic guidance schedules by enforcing a marginal consistency condition along the diffusion sampling trajectory. We train a lightweight MLP through an objective that matches the marginal densities of the guided distribution and true data distribution along the sampling trajectory. Empirically, our method outperforms constant CFG schedule, heuristic-based dynamic schedules, as well as, prior learned guidance approaches on human preference and text alignment metrics on the MS-COCO 512×512 benchmark. Despite its empirical success, our method has several limitations that provide avenues for future work. First, while our learned schedule introduces negligible computational overhead during inference, the training procedure is complex. Optimizing an adversarial game over a diffusion trajectory requires careful choice of training objectives, optimization strategies such as Two-Time-Scale Update Rule (TTUR). Second, the learned guidance scheduler is backbone-specific. As the network relies on the local geometric statistics of a specific pre-trained flow matching model (e.g., MMDiT), the scheduler cannot be transferred zero-shot to a different architecture or velocity field without re-training. Finally, our objective for T2I incorporates a CLIP reward loss. There are avenues to explore alternate reward signals that might overcome drawbacks of CLIP reward (Kang et al., 2025), which we leave as future work.

AI USE STATEMENT

In this work, we used AI tools to proofread the code, to find and fix bugs, and to write some parts of code. On top of that, we used AI to proofread the paper, find grammatical errors and typos and improve writing of some sentences in the manuscript. We have reviewed all AI-assisted work – AI found bugs which we verified were bugs and we fixed them manually, AI wrote some small chunks of code (self-contained functions of boiler-plate code), which we manually verified. We have not used generative AI tools for deriving the methodology, ideas and experimental protocols in this work, nor to conduct any experiments. We did not use generative AI for bibliography and related work, nor we used it to write the mathematical claims – all these were done by hand. The rest of the required tasks such as generation of synthetic data sets, providing critical ingredients for proving mathematical claims, assistance in the writing of proofs, assistance in translation, and cleaning and reformatting the dataset are not applicable to this work. We also did not use AI to interpret the results of this work, or perform any qualitative and thematic data analysis. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 843–852, 2023. 14

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. 6

Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? a fine-grained analysis in a simple setting. In *Advances in Neural Information Processing Systems*, 2024. 6

Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. CFG++: Manifold-constrained classifier free guidance for diffusion models. In *International Conference on Learning Representations*, 2025. 6

Koulischer Felix, Deleu Johannes, Demeester Thomas, and Ambrogioni Luca. Feedback guidance of diffusion models. In *Advances in Neural Information Processing Systems*, 2025. 6

Alexandre Galashov, Ashwini Pokle, Arnaud Doucet, Arthur Gretton, Mauricio Delbracio, and Valentin De Bortoli. Learn to guide your diffusion model. In *International Conference on Learning Representations*, 2026. 1, 3, 5, 6, 7, 8, 9, 14, 15, 21

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2014. 2, 6

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012. 2, 3

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Empirical Methods in Natural Language Processing*, 2021. 7

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 1, 5, 7

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1, 2, 6

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9307–9315, 2024. 9

Raphi Kang, Yue Song, Georgia Gkioxari, and Pietro Perona. Is CLIP ideal? no. can we fix it? yes! In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22436–22446, 2025. 9

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Advances in Neural Information Processing Systems*, 2024. 6

Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv preprint arXiv:2211.17091*, 2022. 6

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023. 7

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Advances in Neural information Processing Systems*, 2024. 1, 3, 6, 7, 14, 15

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742. 7

Gen Li and Yuchen Jiao. Provable efficiency of guidance in diffusion models for general data distribution. In *International Conference on Machine Learning*, 2025. 6

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014. 7

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. 1, 2, 7

Dawid Malarz, Artur Kasymov, Maciej Zięba, Jacek Tabor, and Przemysław Spurek. Classifier-free guidance with adaptive scaling. *arXiv preprint arXiv:2502.10574*, 2025. 6

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*. PMLR, 2018. 4

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 21

Krunoslav Lehman Pavasovic, Jakob Verbeek, Giulio Biroli, and Marc Mézard. Understanding classifier-free guidance: High-dimensional theory and non-linear generalizations. *arXiv preprint arXiv:2502.07849*, 2025. 6

Mateusz Poleski, Jacek Tabor, and Przemyslaw Spurek. Geoguide: Geometric guidance of diffusion models. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 297–305, 2025. 6

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 21

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*, 2021. 7, 14

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*, 2022. 1, 3

Seyedmorteza Sadat, Otmar Hilliges, and Romann M Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations*, 2024. 1, 3, 6

Seyedmorteza Sadat, Manuel Kansy, Otmar Hilliges, and Romann M Weber. No training, no problem: Rethinking classifier-free guidance for diffusion models. In *International Conference on Learning Representations*, 2025. 3

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, and Tim Salimans. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. 1, 3

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 14

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024. 14

Christoph Schuhmann. LAION-Aesthetics. https://laion.ai/blog/laion-aesthetics/, 2022. Accessed: 2025-06-27. 7

Yifei Shen, Xinyang Jiang, Yifan Yang, Yezhen Wang, Dongqi Han, and Dongsheng Li. Understanding and improving training-free loss-based diffusion guidance. In *Advances in Neural Information Processing Systems*, 2024. 14

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. 1

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. 1

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. 1

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023. 14

G. Székely and M. Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5, 2004. 2, 3

Alexandre Verine, Ahmed Mehdi Inane, Florian Le Bronnec, Benjamin Negrevergne, and Yann Chevaleyre. Improving discriminator guidance in diffusion models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 222–238. Springer, 2025. 6

Xi Wang, Nicolas Dufour, Nefeli Andreou, Marie-Paule Cani, Victoria Fernández Abrevaya, David Picard, and Vicky Kalogeiton. Analysis of classifier-free guidance weight schedulers. *Transactions on Machine Learning Research*, 2024. 1, 3, 6, 7, 14, 16

Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-GAN: Training GANs with diffusion. In *International Conference on Learning Representations*, 2023. 6

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. 7

Mengfei Xia, Nan Xue, Yujun Shen, Ran Yi, Tieliang Gong, and Yong-Jin Liu. Rectified diffusion guidance for conditional generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13371–13380, 2025. 6

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022. 6

Yilun Xu, Weili Nie, and Arash Vahdat. One-step diffusion models with f-divergence distribution matching. *arXiv preprint arXiv:2502.15681*, 2025. 14

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems*, 2024a. 14

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b. 14

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23174–23184, 2023. 14

ORGANIZATION OF THE APPENDIX

In Appendix A, we provide image samples for MMDiT-S. In Appendix B, we present the extended related work. In Appendix C, we present additional results and ablations which were not included in the main paper. It includes additional experimental comparisons to baselines in Appendix C. Moreover, it includes additional ablations over the hyperparameters for the considered baselines, see Appendix C.2. Appendix C.3 shows results for the parameters of the time distribution $p(s, t)$ from Section 3.2. On top of that, in Appendix C.4, we show how the number of sampling steps affects performance of our guidance method. Moreover, in Appendix C.5, we show the impact of the generator network architecture for learning the guidance weights. Furthermore, in Appendix C.6, we study the importance of the correlation in the noise distribution of clean data for the Algorithm 1. In Appendix C.7, we study the impact of how the pairs (x_1, c) are sampled. Finally, in Appendix C.9 we discuss alternative choices for the time distribution. In Appendix D, we provide the experimental details and in Appendix E, we show additional image samples from various models.

A MMDIT-S SAMPLES

B EXTENDED RELATED WORK

Approaches for General Guidance. Beyond the scope of CFG, a broader class of methods aim to guide diffusion model by leveraging external models or constraints without requiring any re-training. Universal guidance (Bansal et al., 2023) pioneered this area by demonstrating that the gradients of any differentiable model such as the CLIP feature extractor (Radford et al., 2021), a segmentation network, and an object detection network, can be used guide the diffusion sampling process towards specific attributes. Loss-based guidance methods such as FreeDoM (Yu et al., 2023) and subsequent works (Shen et al., 2024) treat the guidance process as a form of energy-based optimization where the latent state is iteratively udpated to minimize a user-defined loss.

Distillation. Our approach also shares conceptual similarities with the literature on diffusion distillation, especially adversarial distillation methods. Distribution matching distillation (DMD, DMDv2) (Yin et al., 2024b;a) and latent adversarial diffusion distillation (LADD) (Sauer et al., 2024) utilize a discriminator to match the marginal distribution of generated samples to the real data distribution, eliminating the blurriness from trajectory-based distillation methods such as Progressive distillation (Salimans & Ho, 2022) and consistency models (Song et al., 2023). f-distill (Xu et al., 2025) further generalizes this by minimizing f-divergence via an adversarial game.

C ADDITIONAL RESULTS AND ABLATIONS

C.1 EXTENSIVE COMPARISONS TO BASELINES

To complement the results from Section 5.1, we add additional comparison to (Galashov et al., 2026) baseline in Table 5 and in Table 6.

C.2 TEXT-TO-IMAGE GENERATION BASELINE ABLATIONS

In Table 7, we present results on MMDiT-XS model of limited interval guidance (Kynkäänniemi et al., 2024) baseline with a large sweep over the hyperparameters such as the guidance weight and the interval. Furthermore, in Table 8, we present results for clap-linear guidance schedule (Wang et al., 2024).

C.3 ABLATION ON TIME DISTRIBUTION $p(s, t)$ FROM SECTION 3.2

In Table 9, we present results with different hyperparameters of the time distribution $p(s, t)$ discussed in Section 3.2. We see that overall the performance of the method is quite robust to these parameters – the CLIP score does not seem to be affected much by these parameters.

Table 5: **MS COCO** 512×512. Performance of different methods on a small, MMDiT-XS model. We highlight the best metric in **bold**. † indicates that we follow the setup from Galashov et al. (2026).

Method	Guidance Weight	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
Unguided	$\omega = 0$	27.65	0.2742	4.79	0.2547	0.1997
Constant	$\omega = 7.5$	29.73	0.3044	5.29	0.2817	0.2186
LIG	$\omega(t) = 12.5, t \in [0.1, 0.9]$	**25.57**	0.3032	5.30	0.2812	0.2171
CLG	$\omega = 15.0$	26.38	0.3038	5.26	0.2807	0.2175
MMD + SC	$\boldsymbol{\omega}(s, t, c)^\dagger$	28.39	0.3039	5.25	0.28	0.2179
MMD + SC	$\boldsymbol{\omega}(s, t, c)$	27.72	0.3039	5.26	0.2810	0.2179
MMD + SC	$\boldsymbol{\omega}(x_s, s, t, c)$	28.38	0.3042	5.26	0.2813	0.2181
MMD + MC	$\boldsymbol{\omega}(s, t, c)^\dagger$	26.38	0.3034	5.23	0.2799	0.2173
MMD + MC	$\boldsymbol{\omega}(s, t, c)$	26.22	0.3037	5.23	0.2799	0.2172
MMD + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	28.11	0.3037	5.25	0.2807	0.2179
GAN + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	31.31	**0.3048**	**5.32**	**0.2829**	**0.2186**

Table 6: **MS COCO** 512×512. Performance of different methods on a large, MMDiT-S model. We highlight the best metric in **bold**. † indicates that we follow the setup from Galashov et al. (2026).

Method	Guidance Weight	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
Unguided	$\omega = 0$	24.3	0.279	4.84	0.2588	0.2024
Constant	$\omega = 7.5$	31.11	0.3063	5.31	0.285	**0.2214**
LIG	$\omega(t) = 15.0, t \in [0.1, 0.9]$	**24.95**	0.3042	5.31	0.2843	0.2194
CLG	$\omega = 14.0$	27.14	0.3055	5.27	0.2838	0.2203
MMD + SC	$\boldsymbol{\omega}(s, t, c)^\dagger$	28.5	0.3062	5.27	0.2835	0.2204
MMD + SC	$\boldsymbol{\omega}(s, t, c)$	28.72	0.3060	5.27	0.2835	0.2205
MMD + SC	$\boldsymbol{\omega}(x_s, s, t, c)$	28.55	0.3067	5.29	0.2847	0.2208
MMD + MC	$\boldsymbol{\omega}(s, t, c)^\dagger$	28.8	0.3063	5.27	0.2835	0.220
MMD + MC	$\boldsymbol{\omega}(s, t, c)$	27.74	0.3062	5.26	0.2832	0.2202
MMD + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	29.43	0.3063	5.28	0.2836	0.2205
GAN + MC	$\boldsymbol{\omega}(x_s, s, t, c)$	31.7	**0.3068**	**5.34**	**0.2856**	**0.2214**

Table 7: **MS COCO** 512×512. Performance of limited interval guidance (Kynkäänniemi et al., 2024) on the MMDiT-XS model with 128 sampling steps. We highlight the best metrics in **bold**.

Guidance Weight	Interval	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	Pickscore ↑
7.5	$[0.1, 0.9]$	23.97	0.3018	5.26	0.2795	0.2166
7.5	$[0.1, 0.8]$	24.00	0.3018	5.26	0.2795	0.2166
7.5	$[0.2, 0.9]$	**19.65**	0.2969	5.17	0.2734	0.2124
7.5	$[0.2, 0.8]$	19.66	0.297	5.17	0.2735	0.2124
10.0	$[0.1, 0.9]$	24.70	0.3024	5.29	0.2805	0.217
10.0	$[0.1, 0.8]$	24.83	0.2977	5.29	0.2806	0.217
10.0	$[0.2, 0.9]$	20.32	0.3024	5.20	0.2747	0.2129
10.0	$[0.2, 0.8]$	20.3	0.2976	5.21	0.2747	0.2129
12.5	$[0.1, 0.9]$	25.57	**0.3032**	5.30	0.2812	**0.2171**
12.5	$[0.1, 0.8]$	25.68	0.2985	5.30	0.2812	**0.2171**
12.5	$[0.2, 0.9]$	21.48	0.3031	5.23	0.2755	0.2131
12.5	$[0.2, 0.8]$	21.57	0.2985	5.23	0.2755	0.2132
15.0	$[0.1, 0.9]$	25.31	0.3029	**5.31**	**0.2815**	0.2170
15.0	$[0.1, 0.8]$	25.42	0.3029	**5.31**	**0.2815**	**0.2171**
15.0	$[0.2, 0.9]$	21.0	0.2982	5.24	0.2760	0.2132
15.0	$[0.2, 0.8]$	21.0	0.2981	5.24	0.2760	0.2132

Table 8: **MS COCO** 512×512. Performance of clamp-linear guidance schedule $w_t = \max(c_1, \frac{2w_1 t}{T})$ (Wang et al., 2024) on the MMDiT-XS model with 128 sampling steps. We highlight the best metrics in **bold**.

Guidance Weight ω_1	c_1	**FID** ↓	**CLIP** ↑	**Aesthetic** ↑	**HPSv2** ↑	**Pickscore** ↑
7.5	1	**22.37**	0.3012	5.17	0.2769	0.2167
7.5	2	23.74	0.302	5.18	0.2778	0.2161
7.5	4	26.82	0.303	5.23	0.2798	0.2174
10.0	1	23.15	0.3022	5.21	0.2782	0.2165
10.0	2	24.24	0.3027	5.21	0.2788	0.2166
10.0	4	26.55	0.3033	5.24	0.2801	0.2174
12.5	1	23.75	0.3029	5.23	0.2791	0.2162
12.5	2	24.56	0.3032	5.24	0.2795	0.2168
12.5	4	26.64	0.3036	5.25	0.2804	**0.2175**
15.0	1	24.37	0.3033	5.25	0.2798	0.2167
15.0	2	24.88	0.3036	5.25	0.28	0.2169
15.0	4	26.38	**0.3038**	**5.26**	**0.2807**	**0.2175**

Table 9: Performance metrics for MMDiT-XS for different values of δ values in time distribution $p(s,t)$ and different CLIP reward scale η.

δ	Reward scale η	**FID** ↓	**CLIP** ↑	**Aesthetic** ↑	**HPSv2** ↑	**Pick Score** ↑
0.01	2.50	31.54	0.3049	5.3287	0.2830	0.2188
0.10	2.50	31.15	0.3047	5.3145	0.2826	0.2188
0.20	2.50	30.88	0.3049	5.3190	0.2828	0.2187
0.30	2.50	31.16	0.3049	5.3207	0.2829	0.2189
0.01	0.25	31.24	0.3048	5.3179	0.2827	0.2189
0.10	0.25	31.31	0.3048	5.3236	0.2829	0.2186
0.20	0.25	30.36	0.3049	5.3118	0.2826	0.2185
0.30	0.25	31.17	0.3049	5.3202	0.2828	0.2189



(a) *A car and a public transit vehicle on a road.*



(b) *A person holds a flip phone displaying the screen.*



(c) *A cake has had a slice taken out of it and put onto the plate.*



(d) *A display case in front of a store filled with umbrellas.*



(e) *A white dog sitting on a ledge of a window.*



(f) *A black and white dog laying on a bed.*

Figure 4: **T2I Results on MS-COCO with MMDiT-S**. (left-to-right) We provide results of images generated from the given text prompt with CFG $\omega = 7.5$, CLG, LIG, MMD and our method.



(a) *A kitchen that has various paintings on the walls and a large fish aquarium against the wall.*



(b) *A couple of people are flying a kite on top of a sandy beach.*

Figure 5: **T2I Results on MS-COCO with MMDiT-S**. (left-to-right) We provide results of images generated from the given text prompt with CFG $\omega = 7.5$, CLG, LIG, MMD and our method.

C.4 Impact of number of sampling steps

In Table 10, we show the impact of number of sampling steps on performance of our method. We see that overall, doing more sampling steps leads to better perceptual metrics and to a better prompt alignment.

Table 10: Impact of number of sampling steps on the performance of our GAN-based guidance scheduler. We use MMDiT-S backbone.

Sampling steps	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
128	31.70	**0.3068**	**5.34**	**0.2856**	**0.2214**
64	30.55	**0.3068**	5.32	0.2853	0.2212
32	**28.45**	0.3062	5.31	0.2835	0.2196

C.5 Impact of generator architecture

In Table 11, we present results of our method where we vary the architecture of our generator. See Section D for more details on the generator architecture. We see, that light-weight MLP generator which receives the statistics of x_s, leads to overall better performance compared to a more complex Convolutional neural net which receives raw latents x_s.

Table 11: Impact of generator architecture and choice of inputs (*i.e.,*raw x_s vs statistics) with MMDiT-XS backbone.

Architecture	x_s input type	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
Convolutional	x_s latents	**26.38**	0.3041	5.26	0.2807	0.2170
MLP	x_s statistics	31.31	**0.3048**	**5.32**	**0.2829**	**0.2186**

C.6 Impact of how the noise is sampled during training

In Algorithm 1, the real target x_t^{real} and the proposal base x_s can be constructed using either coupled ($\epsilon = z$) or independent ($\epsilon \neq z$) Gaussian noise vectors. We evaluate this design choice in Table 12 using an MMDiT-XS backbone. The empirical metrics indicate that enforcing true independence between the target and proposal noise states provides minor improvements in downstream generation

quality (e.g., lower FID and improved HPSv2). This confirms that the marginal consistency objective is robust to the underlying noise coupling strategy, as the network generalizes effectively without requiring correlated trajectory pairings during training.

Table 12: Impact of how the noise sampled on the performance of our GAN-based guidance scheduler. We use MMDiT-XS backbone.

Independent noise	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
True	**31.31**	**0.3048**	**5.32**	**0.2829**	**0.2186**
False	31.41	**0.3048**	**5.32**	0.2827	**0.2186**

C.7 Impact of how image-conditioning pairs are sampled

The ablation study in Table 13 demonstrates that our method is quite robust to the sampling strategy of image-conditioning pairs. While independent sampling is theoretically more aligned with the marginal consistency objective, it yields nearly identical performance to correlated sampling across most metrics.

Table 13: Impact of choosing independent image-conditioning pairs on performance of our method with MMDiT-XS backbone.

Independent (x_1, c)	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
✗	31.41	**0.3048**	**5.32**	**0.2927**	**0.2186**
✓	**31.31**	**0.3048**	**5.32**	0.2829	**0.2186**

C.8 Impact of CLIP reward loss

We perform an ablation of our method without CLIP reward loss Equation (19). We notice that the metrics such as CLIP score and human preference metrics drop marginally in the absence of this loss for MMDiT-XS model. We summarize these results in Table 14.

Table 14: Impact of the CLIP reward loss (Equation (19)) on the performance of our method with MMDiT-XS backbone.

CLIP reward	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
✓	31.41	**0.3048**	**5.32**	**0.2927**	**0.2186**
✗	**31.11**	0.3047	5.31	0.2825	0.2184

C.9 Alternative choices for the distribution $p(s,t)$

The distribution $p(s,t)$ from Section 3.2 samples t uniformly and then draws the step size Δs uniformly conditional on t. The set of valid time pairs forms a triangular (trapezoid) region

$$\mathcal{T} = \big\{(s,t) \;:\; \zeta + \delta \le t \le 1 - \zeta,\;\; \zeta \le s \le t - \delta\big\}. \tag{21}$$

However, $p(s,t)$ from Section 3.2 does not lead to a uniform distribution over this region.

Uniform distribution over $\mathcal{T}$. We now discuss a choice where we sample uniformly over $\mathcal{T}$. In order to achieve this, we: (1) draw $u \sim \mathcal{U}[0,1]$; (2) compute $t = \zeta + \delta + (1 - 2\zeta - \delta)\sqrt{u}$; and (3) sample $s \sim \mathcal{U}[\zeta,\, t - \delta]$.

Now, we will write a quick derivation for that. A uniform joint density over $\mathcal{T}$ requires $p(s,t) = 1/|\mathcal{T}|$ for all $(s,t) \in \mathcal{T}$. Factoring the joint as $p(s,t) = p(t)\, p(s \mid t)$ and choosing $s \mid t \sim \mathcal{U}[\zeta,\, t - \delta]$, i.e. $p(s \mid t) = 1/w(t)$ where $w(t) = t - \delta - \zeta$ is the width of the valid strip at t, we obtain

$$p(s,t) = p(t) \cdot \frac{1}{w(t)}. \tag{22}$$

Setting this equal to $1/|\mathcal{T}|$ and solving for $p(t)$ gives

$$p(t) = \frac{w(t)}{|\mathcal{T}|} = \frac{t - \delta - \zeta}{|\mathcal{T}|}, \tag{23}$$

confirming that t must be sampled with probability proportional to the strip width $w(t)$. The area of $\mathcal{T}$ is

$$|\mathcal{T}| = \int_{\zeta+\delta}^{1-\zeta} w(t)\,\mathrm{d}t = \int_{\zeta+\delta}^{1-\zeta} (t - \delta - \zeta)\,\mathrm{d}t = \frac{(1 - 2\zeta - \delta)^2}{2}. \tag{24}$$

The CDF of t under equation 23 is

$$F(t) = \frac{(t - \zeta - \delta)^2}{(1 - 2\zeta - \delta)^2}, \tag{25}$$

which inverts to

$$t = \zeta + \delta + (1 - 2\zeta - \delta)\sqrt{u}, \qquad u \sim \mathcal{U}[0, 1]. \tag{26}$$

Given t, the source time $s \mid t \sim \mathcal{U}[\zeta, t - \delta]$. Substituting back, the joint density is $p(s, t) = p(t)/w(t) = 1/|\mathcal{T}|$, which is constant over $\mathcal{T}$.

Beta-trapezoid distribution over $\mathcal{T}$. The uniform-joint sampler above can be generalized by replacing the uniform quantile $u \sim \mathcal{U}[0, 1]$ with a Beta-distributed one, $u \sim \mathrm{Beta}(\alpha, \beta)$, and feeding it through the same inverse CDF equation 26. When $\alpha = \beta = 1$, the Beta distribution reduces to $\mathcal{U}[0, 1]$, exactly recovering the uniform-joint distribution. Since the inverse CDF $t = \zeta + \delta + (1 - 2\zeta - \delta)\sqrt{u}$ is monotonically increasing in u, biasing u toward larger values translates directly into biasing t toward larger values. Setting $\alpha > \beta$ increases the mean of the Beta distribution (recall $\mathbb{E}[u] = \alpha/(\alpha + \beta)$), which shifts mass toward larger t (closer to data), while $\beta > \alpha$ shifts mass toward smaller t (closer to noise).

Results. We present the results in Table 15. The last 3 rows represent the distribution $p(s, t)$ from Section 3.2. We see that $\alpha > \beta$, which shifts mass toward larger t (closer to data), leads to worse results, while $\alpha \leq \beta$ performs well. Overall, $(\alpha, \beta) = (1, 2)$ leads to the best results, which are comparable to those obtained with the distribution $p(s, t)$ from Section 3.2.

Table 15: **Beta-trapezoid time distribution ablation on MMDiT-XS.** We vary the Beta distribution parameters (α, β) and minimum step size δ. When $\alpha = \beta = 1$, the distribution reduces to uniform sampling over the triangular support $\mathcal{T}$. The last 3 rows correspond to the distribution $p(s, t)$ from Section 3.2.

α	β	δ	FID ↓	CLIP ↑	Aesthetic ↑	HPSv2 ↑	PickScore ↑
1	1	0.01	30.85	0.3048	5.32	0.2827	0.2184
		0.10	30.79	0.3038	5.31	0.2819	0.2179
		0.20	31.52	0.3048	5.32	0.2828	0.2186
1	2	0.01	31.87	0.3046	5.32	0.2826	0.2187
		0.10	32.00	0.3048	5.33	0.2829	0.2188
		0.20	31.54	0.3048	5.32	0.2827	0.2185
2	1	0.01	30.58	0.3033	5.31	0.2824	0.2183
		0.10	28.71	0.3013	5.26	0.2795	0.2165
		0.20	30.57	0.3033	5.30	0.2814	0.2176
-	-	0.01	31.24	0.3048	5.32	0.2827	0.2189
		0.10	31.31	0.3048	5.32	0.2829	0.2186
		0.20	30.36	0.3049	5.31	0.2826	0.2185

D Experimental Details

We specify architectural details of discriminator and guidance network below. in addition, we state our optimizer hyperparameter settings in Table 16.

Discriminator architecture. The discriminator $d_\phi(x_t, s, t, c)$ is a convolutional network inspired by DCGAN architecture (Radford et al., 2015). It follows a modular downsampling structure consisting of five primary modules. Each module utilizes a Conv-Residual-AdaGN-Leaky-ReLU block. For convolutional layers, we utilize 4×4 kernels with a stride of 2 and explicit padding of 1 for the first three stages to progressively reduce spatial resolution while doubling feature depth. The final two stages utilize 2×2 kernels to reach a 1×1 spatial bottleneck. Each block incorporates a residual connection with a 1×1 projection convolution to match feature dimensions. We also apply spectral normalization (Miyato et al., 2018) to all convolutional layers to enforce Lipschitz continuity and prevent GAN instability. The discriminator starts with a base feature width of 64, expanding to 1024 in the final layer. The Leaky ReLU uses slope of 0.2.

The network also accepts two time steps, the source time s and target time t. Each temporal scalar is projected into a 64-dimensional space via separate MLP-based time encoders. These embeddings are concatenated and processed through a two-layer MLP with SiLU activation to create a joint temporal embedding $\mathbf{e}_{\text{joint}}$. The CLIP text embeddings c_{CLIP} are projected via a dedicated MLP to match the hidden dimension of the temporal features. These three embeddings are concatenated and passed through a two layer MLP with SiLU activations, and then injected into each convolutional block using Adaptive Group Normalization (AdaGN).

We use the same discriminator architecture for both MMDiT-XS and MMDiT-S. Overall, this discriminator has 5M trainable parameters. We also use this same architecture for the generator architecture for the results in Table 11.

Guidance network architecture. The guidance network $\boldsymbol{\omega}(x_s, s, t, c; \psi)$ is a MLP that predicts guidance scale. We utilize Fourier Embeddings to project the source time s, target time t, and the temporal gap $(t - s)$ into a 256-dimensional space. We use standard sinusoidal frequencies with a $1/10000$ scaling factor. These embeddings are concatenated and passed through a linear layer with Layer Normalization and SiLU activation. We also input the CLIP text embeddings after projecting them into a 256-dimensional hidden space via a Dense-LayerNorm-SiLU block. We also condition the network on a 5-dimensional vector of coordinate-invariant geometric statistics as described in section 3.2 of the main paper. These statistics are projected to a 128-dimensional feature vector. The fused features are passed into a central backbone consisting of a projection layer followed by two Residual Blocks. Each residual block follows the structure:

$$\text{Output} = \text{SiLU}(x + \text{LayerNorm}(\text{Dense}(\text{SiLU}(\text{LayerNorm}(\text{Dense}(x)))))) \tag{27}$$

The final layer is a linear projection to a single scalar, followed by a softplus activation to ensure non-negative guidance scale. We utilize a zero-initialization strategy for the final weight matrix. The bias of the final layer is initialized to $\text{softplus}^{-1}(\alpha)$, where α is the starting guidance multiplier (e.g., 1.0). This ensures that at the start of training, the model defaults to a standard constant schedule before learning to optimize the trajectory.

Training details. The training details are specified in Table 16. All models are trained for 60000 iterations. During training, we track CLIP score on a subset of 3000 images from MS-COCO 512×512. We use this metric to select the best checkpoint for every studied method.

Baselines training details. For MMD+SC and MMD+MC baselines, unless specified otherwise, we follow the same training protocol as above and use the guidance network $\boldsymbol{\omega}(x_s, s, t, c)$. The only difference is that we do not employ discriminator and we do not use regularization equation 17. On top of that, we consider the same experimental setup where we do not include x_s into the guidance network $\boldsymbol{\omega}(s, t, c)$. Finally, we also include results with $\boldsymbol{\omega}(s, t, c)^\dagger$, where we follow the setup from Galashov et al. (2026).

E IMAGE SAMPLES

The image samples for MMDiT-XS are given in Figure 6. The image samples for MMDiT-S are given in Figure 4, Figure 5, Figure 7.

Table 16: **Training and Sampling Details.** Hyperparameter settings for MMDiT-XS, MMDiT-S.

Config	**MMDiT-XS**	**MMDiT-S**
Discriminator Training		
Batch size	128	128
Optimizer	Adam	Adam
Learning rate	1e-5	1e-5
(β_1, β_2)	(0.5, 0.99)	(0.5, 0.99)
Warmup steps	5000	5000
Training steps	60000	60000
R_1 penalty γ	0.1	0.1
R_1 penalty interval	5	4
Generator Training		
Batch size	128	128
Optimizer	Adam	Adam
Learning rate	2e-5	2e-5
(β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)
Warmup steps	5000	5000
Training steps	60000	60000
Regularization weight a	0.1	1e-5
Clip reward weight η	0.25	0.25
Sampling		
Sampler type	Euler	Euler
Sampling steps	128	128



(a) *A city bus is riding down the empty street.*



(b) *Stuffed teddy bear sitting next to garbage can on the side of the road.*



(c) *The train moves thru this part of the city.*



(d) *The dark mirror shows a bathroom in its reflection.*



(e) *A computer monitor sitting on top of a computer desk.*



(f) *Luggage sitting on and around airline luggage cards on the tarmac.*



(g) *A batter swings at the ball during a baseball game.*

Figure 6: **T2I Results on MS-COCO with MMDiT-XS**. (left-to-right) We provide results of images generated from the given text prompt with CFG $\omega = 7.5$, CLG, LIG, MMD and our method.



(a) *Two zebras in the zoo by some trees.*



(b) *A man riding a snowboard down the side of a snow covered ski slope.*



(c) *A grey cat sitting by a round mirror.*



(d) *A small board sits on the water next to a city.*



(e) *Wild animals graze in field in front of a lake and snow covered mountains.*



(f) *A group of stuffed animals are arranged together.*

Figure 7: **Variation with seeds: T2I Results on MS-COCO with MMDiT-S**. (left-to-right) We provide results of images generated from the given text prompt with different seeds (0,1,2,3) with our method.