Adaptive Diffusion Guidance via Stochastic Optimal Control

Iskander Azangulov
Department of Statistics
University of Oxford

Peter Potaptchik
Department of Statistics
University of Oxford

Qinyu Li
Department of Statistics
University of Oxford

Eddie Aamari
DMA, École normale supérieure,
Université PSL, CNRS

George Deligiannidis
Department of Statistics
University of Oxford

Judith Rousseau
CEREMADE, Université Paris-Dauphine,
PSL University, CNRS

Abstract

Guidance is a cornerstone of modern diffusion models, playing a pivotal role in conditional generation and enhancing the quality of unconditional samples. However, current approaches to guidance scheduling—determining the appropriate guidance weight—are largely heuristic and lack a solid theoretical foundation. This work addresses these limitations on two fronts. First, we provide a theoretical formalization that precisely characterizes the relationship between guidance strength and classifier confidence. Second, building on this insight, we introduce a stochastic optimal control framework that casts guidance scheduling as an adaptive optimization problem. In this formulation, guidance strength is not fixed but dynamically selected based on time, the current sample, and the conditioning class, either independently or in combination. By solving the resulting control problem, we establish a principled foundation for more effective guidance in diffusion models.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) have emerged as a dominant paradigm in generative modeling, achieving state-of-the-art results across various domains, including text-to-image generation (Rombach et al., 2022; Ramesh et al., 2022; Zhang et al., 2023), video synthesis (Ho et al., 2022; Singer et al., 2022; Blattmann et al., 2023), and molecular design (Bao et al., 2022; Weiss et al., 2023; Schneuing et al., 2024). These models operate by defining a forward diffusion process X_t that gradually adds noise to data, $X_0 \sim p$, and then generating samples by learning to reverse this noising process. The backward process is driven by the score function $\nabla \log p_t$, where p_t is the density of X_t. In practice, the score is approximated by a neural network trained via score matching (Song et al., 2021).

While incredibly powerful, the raw output from these diffusion models often benefits from refinement to enhance sample quality and ensure alignment with specific desired conditions or characteristics. Such tasks are widespread, from generating images based on text to steering protein generation to meet specific constraints. *Guidance* (Dhariwal and Nichol, 2021; Ho and Salimans, 2022) techniques have, therefore, become an indispensable part of generative modeling, ubiquitously employed across a variety of tasks (Ramesh et al., 2022; Saharia et al., 2022; Schiff et al., 2025).

Among guidance techniques, Classifier-Free Guidance (CFG) (Ho and Salimans, 2022) has become a cornerstone, significantly improving conditional generation. Let c be a conditioning input such as a class label or a textual prompt. CFG involves training a single diffusion model on complementary objectives to simultaneously learn the conditional scores $\nabla \log p(x|c)$ for different conditions c and the unconditional score $\nabla \log p(x)$. During sampling, CFG modifies the standard reverse diffusion process by introducing a guidance strength parameter $w \in \mathbb{R}$. This parameter induces a guided score $\nabla \log p_t(x|c) + w(\nabla \log p_t(x|c) - \nabla \log p_t(x))$, which interpolates between the conditional ($w = 0$) and the unconditional ($w = -1$) scores.

In practice, the guidance strength w is typically chosen

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

to be much greater than 0 to amplify the influence of the conditioning information. For example, in image generation, w commonly ranges from 3 to 16 (Ho and Salimans, 2022; Podell et al., 2023; Li et al., 2024; Stability-AI, 2025), though this is just a heuristic choice and the theoretical understanding of the guiding mechanism is still limited. We discuss this in more detail in the next section.

1.1 Related Works

While choosing constant guidance is standard practice, recent works have explored various heuristics for improving guidance via non-constant weight schedules. For instance, Chang et al. (2023); Gao et al. (2023) proposed schedules where guidance strength increases with the current noise level. Kynkäänniemi et al. (2024) proposed only applying guidance at empirically specified time intervals. Additionally, Shen et al. (2024) proposed using different guidance weights across different semantic regions, arguing that a fixed guidance weight results in spatial inconsistencies.

From a theoretical point of view, the understanding of the guidance mechanism is still limited. A common intuition is that guidance with weight w effectively samples from a "tilted" target distribution proportional to $p(x)p(c|x)^w$, however, as shown in Chidambaram et al. (2024); Bradley and Nakkiran (2024), this is not true. Another point of ambiguity, and a widely held misconception, is whether guided sampling ($w \neq 0$) ensures that samples remain within the true conditional data support, with some believing that guidance may prevent the recovery of this support. This concern about losing the data support is, interestingly, contradictory to the aforementioned "tilted" distribution intuition.

Recently, Skreta et al. (2025) proposed a Feyman-Kac based corrector that runs a Sequential Monte Carlo scheme along guided trajectories to sample exactly from $\propto p(x)p(c|x)^w$. Also, Domingo-Enrich et al. (2025) suggested a Stochastic Optimal Control based method to train a guiding vector, so that the resulting samples are from $\propto p(x)\exp(r(x))$ for an arbitrary terminal reward $r(x)$.

1.2 Our Contributions

This work addresses these limitations on two main fronts. First, we address a critical gap in the current understanding of guidance: the ambiguity of the precise distribution being sampled, and a lack of formal guarantees when $w \neq 0$. Our work establishes that (i) generated samples are guaranteed to remain within the conditional data manifold; and (ii) applying guidance with any positive strength $w > 0$ increases $p(c|x)$ both with high probability and on average.

Second, building on these theoretical insights, we introduce a novel framework that recasts guidance scheduling as a Stochastic Optimal Control (SOC) problem. In this framework, the guidance strength is no longer a fixed parameter, but is dynamically adjusted based on time, the current sample state, and the conditioning class c. By formulating the resulting control problem, we establish a principled foundation for effective adaptive guidance. The resulting framework optimizes a scalar-valued function $w = w_t(x, c)$ and only requires estimates of $\nabla \log p_t(x)$ and $\nabla \log p_t(x|c)$, making it applicable in the CFG setting.

Finally, we propose and implement a scalable numerical method based on the adjoint method to efficiently solve this control problem. The open-source code for our framework is publicly available at github.com/imbirik/adaptive-guidance.

Concurrent work. While we were finalizing our paper, a concurrent work (Li and Jiao, 2025) appeared, presenting a result similar to our Corollary 2. Using a different approach, they also show that applying guidance $w > 0$ reduces $\mathbb{E}\, p^{-1}(c|Y_t^w)$ compared to the unguided case $w \equiv 0$.

2 BACKGROUND

2.1 Diffusion Models via Stochastic Differential Equations

Diffusion models define a generative process by reversing a forward stochastic differential equation (SDE) that progressively corrupts data into noise. Let $X_0 \sim p_0 = p$ denote the data distribution in $\mathbb{R}^D$. The forward (noising) process is governed by the Ornstein–Uhlenbeck SDE:

$$dX_t = -X_t dt + \sqrt{2} dB_t, \quad t \in [0, T], \tag{1}$$

where B_t is a standard Wiener process. We denote the density of X_t by p_t and remark that

$$X_t | X_0 := \mathcal{N}\left(e^{-t} X_0, (1 - e^{-2t})\,\mathrm{Id}_D\right). \tag{2}$$

Under mild assumptions (Anderson, 1982), the reverse-time process $\{Y_t\}_{t\in[0,T]} := \{X_{T-t}\}_{t\in[0,T]}$ satisfies the SDE

$$dY_t = [Y_t + 2\nabla \log p_{T-t}(Y_t)]\, dt + \sqrt{2} d\bar{B}_t, \tag{3}$$

where $\bar{B}_t$ is a standard Wiener process in the reverse-time direction. In practice, $\nabla \log p_t$ is approximated by a neural network trained via score matching and sampling $Y_T \sim p_T$ is typically approximated by $Y_T \sim \mathcal{N}(0, I_D)$. We refer to Karras et al. (2022) for further details.

2.2 Guidance

Conditional generation from $p(x|c)$ can be achieved by replacing $\nabla \log p_t(x)$ with the conditional score $\nabla \log p_t(x|c)$. However, in order to increase adherence to the condition c, strong guidance is often employed by introducing a guidance strength $w > 0$ and instead using the guided score

$$\nabla \log p_t(x|c) + w\nabla \log p_t(c|x). \tag{4}$$

So, the backward dynamics is given by

$$dY_t^w = \Big[Y_t^w + 2\nabla \log p_{T-t}(Y_t^w|c) + 2w\nabla \log p_t(c|Y_t^w)\Big]dt + \sqrt{2}dB_t. \tag{5}$$

Guided scores can be obtained either by using learned classifiers $p_t(c|x)$ for noisy data, or by applying Bayes' rule $\nabla \log p_t(c|x) = \nabla \log p_t(x|c) - \nabla \log p_t(x)$, yielding Classifier-Free Guidance (CFG) which directly uses a score network for both the unconditional $\nabla \log p_t(x)$ and the conditional $\nabla \log p_t(x|c)$ scores, thereby avoiding a separate classifier.

2.3 Stochastic Optimal Control

Stochastic optimal control deals with the problem of controlling a system whose state evolves over time according to a stochastic process, with the goal of optimizing a certain objective. Consider a stochastic process Y_t^w whose dynamics can be influenced by a control variable $w_t(Y_t^w)$. Define a reward function $R(Y^w, w)$ that quantifies the desirability of the terminal state Y_T^w and the entire trajectory $(Y_t^w)_{t\in[0,T]}$. We consider the following objective for w

$$R(w) := \mathbb{E}\left[R(Y^w, w)\right] = \mathbb{E}\left[\Phi(Y_T^w) + \int_0^T r(t, Y_t^w, w_t(Y_t^w))dt\right],$$

where Φ is a terminal reward function and r is an instantaneous reward function. The goal is to find the control policy $w_t^*(Y_t^w)$ in some class $\mathbb{U}$ that maximizes this reward. The value function $V(t,x)$ is defined as the optimal expected reward obtained by starting from state x at time t:

$$V_t(x) = \sup_{w\in\mathbb{U}} \mathbb{E}\left[\Phi(Y_T^w) + \int_t^T r(s, Y_s^w, w_s(Y_s^w))ds \,\middle|\, Y_t^w = x\right]. \tag{6}$$

Under suitable regularity conditions, the value function satisfies the Hamilton-Jacobi-Bellman (HJB) equation, a backward-in-time partial differential equation:

$$-\frac{d}{dt}V_t(x) = \sup_{w\in\mathbb{U}}\left\{\mathcal{A}^w V_t(x) + r(t,x,w)\right\}, \tag{7}$$

where $\mathcal{A}^w$ is the infinitesimal generator of the stochastic process Y_t^w, which depends on the control w. Solving the HJB equation yields the optimal value function $V_t^*(x)$ and the optimal control policy

$$w_t^*(x) = \arg\max_{w\in\mathbb{U}}\{\mathcal{A}^w V_t(x) + r(t,x,w)\}. \tag{8}$$

We refer to (Fleming and Soner, 2006) for further details.

3 THEORETICAL RESULTS FOR GUIDANCE

In this section, we present our two main theoretical results for guidance. The first result establishes a connection between the guidance strength and $\log p_t(c|Y_{T-t}^w)$ – a measure of the alignment of the guided samples with the classifier at time t. Our second result shows that the application of guidance preserves the support of the true conditional distribution, $\operatorname{supp} p(x|c)$. We defer proofs to Appendix A.

In the remainder of this paper, we consider a generalization of the guidance introduced in Section 2.2, by allowing w to depend on time, position, and the conditioning class. To simplify notation, we suppress these dependencies and write $w := w_t(Y_t^w, c)$.

3.1 Analysis of Classifier Confidence Along Guided Trajectories

Let us introduce the backward running "guiding field" G_t:

$$\begin{aligned} G_t(x) &:= \log p_{T-t}(c|x) - \log p(c) \\ &= \log p_{T-t}(x|c) - \log p_{T-t}(x), \end{aligned} \tag{9}$$

where we also suppress the dependence on c in the notation for G_t. So, (10) reads as

$$dY_t^w = \Big[Y_t^w + 2\nabla \log p_{T-t}(Y_t^w|c) + 2w\nabla G_t(Y_t^w)\Big]dt + \sqrt{2}dB_t. \tag{10}$$

Combining Ito's lemma and the Fokker-Planck equation, we obtain our main technical lemma. We defer the proof to the Appendix.

Lemma 1.

$$dG_t(Y_t^w) = (1+2w)\left\|\nabla G_t(Y_t^w)\right\|^2 dt + \sqrt{2}\nabla G_t(Y_t^w)\cdot dB_t. \tag{11}$$

Taking expectations in (11) and using (9), we can derive an expression for the mean likelihood of target class c at time t, $\mathbb{E}\log p_{T-t}(c|Y_t^w)$. It is given by an integral

of the *known* term ∇G_t, along the paths of the guided process Y^w:

$$\mathbb{E}\log p_{T-t}(c|Y_t^w) = C + \mathbb{E}\int_0^t (1+2w)\left\|\nabla G_s(Y_s^w)\right\|^2 ds, \tag{12}$$

where the constant $C = \log p(c) + \mathbb{E}\log\frac{p_T(Y_0^w|c)}{p_T(Y_0^w)}$ does not depend on the choice of the guidance function w. Moreover, we note that the quadratic variation of G_t satisfies $d\langle G\rangle_t = 2\left\|\nabla G_t\right\|^2 dt$. Therefore, the Doléans-Dade stochastic exponential of the martingale $-\sqrt{2}\nabla G_s(Y_s^w)\cdot dB_s$ is equal to

$$S_t := \exp\left(-G_t + \int_0^t 2w_s\left\|\nabla G_s\right\|^2 ds\right) \tag{13}$$

$$= \frac{p(c)}{p(c|Y_t^w)}\exp\left(\int_0^t 2w_s\left\|\nabla G_s\right\|^2 ds\right). \tag{14}$$

S_t is therefore a non-negative martingale on $[0,T)$ and a super-martingale on $[0,T]$. We refer the reader to Karatzas and Shreve (1998) for more details on the required martingale theory. Leveraging the super-martingale property, we obtain the following corollary.

Corollary 2. *By Doob's inequality for any $\delta > 0$ with probability at least $1-\delta$ sampled guided trajectory Y^w satisfies*

$$\frac{p_{T-t}(Y_t^w|c)}{p_{T-t}(Y_t^w)} \geq \delta\exp\left[\int_0^t 2w_s\left\|\nabla G_s(Y_s^w)\right\|^2 ds\right]\Bigg) \tag{15}$$

for all $t\in[0,T]$. Moreover, if we condition on the event $Y_t^w = x$, then for any $\gamma\in[0,T-t)$

$$p_{T-t}^{-1}(c|x) = \mathbb{E}\left[p_{T-t-\gamma}^{-1}(c|Y_{t+\gamma}^w)\exp\left(\int_t^{t+\gamma} 2w\left\|\nabla G_s\right\|^2 ds\right)\Big|Y_t^w = x\right]. \tag{16}$$

Since the term in the integrand is non-negative, and equal to zero if $w\equiv 0$ we get that for any x s.t. $\nabla G_t(x)\neq 0$

$$\mathbb{E}\left[p_{T-t-\gamma}^{-1}(c|Y_{t+\gamma}^w)\Big|Y_t^w = x\right] < p_{T-t}^{-1}(c|x) = \mathbb{E}\left[p_{T-t-\gamma}^{-1}(c|Y_{t+\gamma}^0)\Big|Y_t^0 = x\right].$$

This implies that the addition of non-negative guidance guarantees increased alignment with the class label, compared to simulation without guidance. Moreover, by (15) the term

$$\int_0^t 2w_s\left\|\nabla G_s(Y_s^w)\right\|^2 ds$$

plays the role of the high probablity lower bound for $\log p_{T-t}(c|Y_t^w) - \log p(c)$.

3.2 Support Recovery

In this section, we prove that under mild assumptions the terminal state of the guided process, Y_T^w, lies within $\operatorname{supp} p_0(\cdot|c)$. Specifically, we only assume: (i) the support of the unconditional distribution is compact, i.e. $\operatorname{supp} p_0 \subseteq B(0,R)$ for some $R>0$; and (ii) the guidance is uniformly bounded, i.e. $0 < w \leq C_{\max}$. Notably, we allow $p_0(\cdot|c)$ to be singular, and our result holds even when the guidance is not constant; (iii) The conditional class has positive weight $p(c) > 0$.

Below we provide a sketch of the proof, while the full proof can be found in the Appendix A. First, we note that under these conditions, for all $t\in[0,T)$, by Tweedie's formula (Robbins, 1956)

$$\left\|\nabla G_t(x)\right\|^2 = \frac{e^{2(t-T)}\left\|\mathbb{E}[X_0|X_{T-t}=x,c] - \mathbb{E}[X_0|X_{T-t}=x]\right\|}{(1-e^{2(t-T)})^2} \leq \frac{4e^{2(t-T)}}{(1-e^{2(t-T)})^2}R^2. \tag{17}$$

Since the true conditional backward process Y_t^0 differs from guided process Y_t^w by the drift term $w_t\nabla G_t$. Girsanov's theorem combined with the data-processing inequality bounds the KL divergence

$$\mathsf{KL}(Y_{T-\varepsilon}^w\|Y_{T-\varepsilon}^0) \leq \mathsf{KL}(Y_{[0,T-\varepsilon]}^w\|Y_{[0,T-\varepsilon]}^0) = \mathbb{E}\int_0^{T-\varepsilon} w_t^2\left\|\nabla G_t(Y_t^w)\right\|^2 dt,$$

for all $\varepsilon>0$, since Novikov's condition holds due to (17). The integrand coincides with (12) up to a scalar $0 < \frac{w_t^2}{1+2w_t} < C_{\max}$, so

$$\mathsf{KL}(Y_{T-\varepsilon}^w\|Y_{T-\varepsilon}^0) \leq \mathbb{E}\int_0^{T-\varepsilon}\frac{w_t^2}{1+2w_t}(1+2w_t)\|\nabla G_t(Y_t^w)\|^2 dt \leq -C_{\max}\log p(c).$$

Therefore, for any $\varepsilon>0$, the KL divergence between $Y_{T-\varepsilon}^w$ and $Y_{T-\varepsilon}^0$ is uniformly bounded. Then support recovery follows from Donsker and Varadhan's variational formula.

Theorem 3. *Assume that there are $R, C_{\max} < \infty$ such that almost surely $\operatorname{supp} p_0 \subseteq B(0,R)$ and $w < C_{\max}$. Then almost surely $Y_T^w \in \operatorname{supp} p(\cdot|c)$.*

Remark 4. *Theorem 3 still holds for negative guidance as long as $w > -\frac{1}{2}+\varepsilon$ uniformly in w for some $\varepsilon>0$.*

4 ADAPTIVE GUIDANCE LEARNING

As shown in the previous section, applying guidance encourages the growth of $\mathbb{E}\log p(c|Y_T^w)$. However, by

itself, the maximization of $\mathbb{E}\log p(c|Y_T^w)$ might be an ill-defined problem. For example, in a simple Gaussian mixture case, this objective is maximized at infinity. Moreover, too much guidance can push trajectories to low-probability regions of $p_t(\cdot|c)$, where the score function is poorly learned.

Both of these problems are addressed if we additionally assume that the guided trajectory distribution Y^w and the unguided one Y are close in KL divergence. Therefore, we introduce our SOC objective that balances these goals

$$R(w) := \mathbb{E}\log p(c|Y_T^w) - \alpha\,\mathsf{KL}(Y^w_{[0,T]} \parallel Y_{[0,T]}), \quad (18)$$

where $\alpha > 0$ governs the tradeoff. This leads to the following SOC problem:

$$w^* = \arg\max_{w\in\mathbb{U}} \mathbb{E}\log p(c|Y_T^w) - \alpha\,\mathsf{KL}(Y^w_{[0,T]} \parallel Y_{[0,T]}), \quad (19)$$

where $\mathbb{U}$ is some class of guidance strength functions. In the most general case, which is the focus of this section, $\mathbb{U}$ is the space of adapted scalar-valued processes. By (12) and Girsanov's theorem,

$$R(w) = \mathbb{E}\int_0^T (1 + 2w_t - \alpha w_t^2)\,\|\nabla G_t(Y_t^w)\|^2\,dt + C, \quad (20)$$

where C is a constant that does not depend on the choice of guidance strength w. We introduce the value function

$$V_t(x) = \sup_w \mathbb{E}\left[\int_t^T (1 + 2w_t - \alpha w_t^2)\,\|\nabla G_s(Y_s^w)\|^2\,ds | Y_t^w = x\right]$$

The associated HJB equation is given by

$$-\frac{d}{dt}V_t(x) = \sup_{w\in\mathbb{R}}\Big[\langle x + 2\nabla\log p_{T-t}(x|c) + 2w\nabla G_t(x), \nabla V(t,x)\rangle + \Delta V(t,x) + (1 + 2w_t - \alpha w_t^2)\,\|\nabla G_t\|^2\Big],$$

with terminal condition $V_T \equiv 0$.

For fixed t, x, the objective in the supremum is a simple quadratic in $w_t(x)$. Thus

$$w_t^*(x) = \frac{\nabla G_t \cdot \nabla V_t + \|\nabla G_t\|^2}{\alpha\,\|\nabla G_t\|^2} \quad (21)$$

and

$$-\frac{d}{dt}V_t(x) = \frac{1}{\alpha}\left\{\frac{\nabla G_t}{\|\nabla G_t\|}\cdot\nabla V_t + \|\nabla G_t\|\right\}^2 + \|\nabla G_t\|^2 + \Delta V(t,x) + \langle x + 2\nabla\log p_{T-t}(x|c), \nabla V(t,x)\rangle. \quad (22)$$

The resulting HJB equation can be estimated using finite difference methods or Physics-Informed Neural Networks (PINN) based objectives. Where the latter involves parameterization of the value function $V_t(x)$ as a neural network and its optimization by minimization of the squared difference between LHS and RHS of (22).

However, both methods work only in low-dimensional cases, and do not allow for solutions that only depend on time t or class c. In the next section, we discuss an alternative, scalable approach based on the adjoint state method.

We emphasize that, unlike methods that learn a new guidance *direction*, our method focuses on tuning only the *scalar* guidance strength w, which allows for simpler network architectures and potentially less training data.

5 OPTIMIZATION OF STOCHASTIC OPTIMAL CONTROL OBJECTIVE

The primary objective of this section is to solve the SOC problem defined in (19) for general $\mathbb{U}$, i.e. find an optimal, adaptive guidance strength $w^* \in \mathbb{U}$ maximizing (20). We parameterize this guidance strength w_θ (potentially dependent on time, state, and conditioning prompt c) using a neural network whose parameters θ are optimized via stochastic gradient descent (SGD).

To begin with we recall the exact formula (20) of our objective $R(w)$

$$R(w) = \mathbb{E}\int_0^T (1 + 2w_t - \alpha w_t^2)\,\|\nabla G_t(Y_t^w)\|^2\,dt + C,$$

The main challenge is that the change of the guidance w_θ not only changes the instant reward $1 + 2w_t(\theta) - \alpha w_t^2(\theta))^2$ but also changes the distribution over trajectories Y^w. To capture this we introduce the *sensitivity process*:

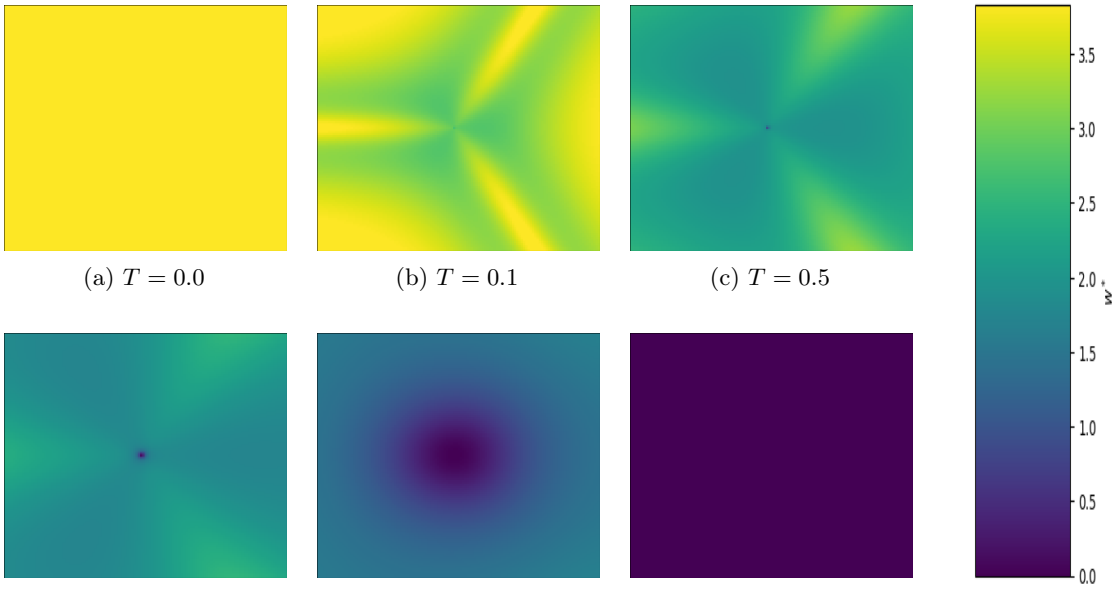
$$Z_t(\tau) = \frac{\partial Y_t^w}{\partial w_\tau}$$

capturing how the value of the process Y_t^w at time $t > \tau$ changes w.r.t. to change of the process at initial time τ. Applying the chain rule we obtain

$$\frac{\partial R(w)}{\partial w_\tau} = 2\,\mathbb{E}(1 - \alpha w_\tau)\,\|\nabla G_t(Y_t^w)\|^2 \quad (23)$$

$$+\mathbb{E}\int_\tau^T 2(1 + 2w_t - \alpha w_t^2)\nabla^T G_t(Y_t^w)\nabla^2 G_t(Y_t^w) Z_t(\tau)\,dt \quad (24)$$

The general approach to computing the second integral is to introduce the adjoint process. First, using the



(a) $T = 0.0$ (b) $T = 0.1$ (c) $T = 0.5$

(d) $T = 1.0$ (e) $T = 2.5$ (f) $T = 5.0$

Figure 1: Optimal guidance w^* obtained by solving the HJB equation.

chain rule, we write the SDE on the sensitivity process

$$dZ_t(\tau) = \Big[Z_t(\tau) + 2\nabla^2 \log p_{T-t}(Y_t^w|c) Z_t(\tau) + 2\delta(\tau - t)\nabla G_\tau(Y_\tau^w) + 2w_t \nabla^2 G_t(Y_t^w) Z_t(\tau)\Big] dt$$

implying that $Z_t(\tau)$ is driven by the ODE

$$\dot{Z}_t(\tau) = \Big[\mathrm{Id} + 2\nabla^2 \log p_{T-t}(Y_t^w|c) + 2w_t \nabla^2 G_t(Y_t^w)\Big] Z_t(\tau),$$

with initial condition $Z_\tau = 2\nabla G_\tau(Y_\tau^w)$. We consider an adjoint backward process λ_t with initial condition $\lambda_T = 0$ and satisfying the ODE

$$\dot{\lambda}_t = -\underbrace{2(1 + 2w_t - \alpha w_t^2)\nabla^T G_t(Y_t^w)\nabla^2 G_t(Y_t^w)}_{:=B_t} - \underbrace{\Big[\mathrm{Id} + 2\nabla^2 \log p_{T-t}(Y_t^w|c) + 2w_t \nabla^2 G_t(Y_t^w)\Big]^T}_{:=A_t} \lambda_t$$

Then, integrating by parts, we see that

$$\langle \lambda_\tau, 2\nabla G_\tau(Y_\tau^w)\rangle = \int_\tau^T 2(1 + 2w_t - \alpha w_t^2)\nabla^T G_t(Y_t^w)\nabla^2 G_t(Y_t^w) Z_t dt$$

that allows us to simplify (31) and get

$$\frac{\partial R(w)}{\partial w_\tau} = 2\,\mathbb{E}(1 - \alpha w_\tau)\,\|\nabla G_t(Y_t^w)\|^2 + 2\langle \lambda_\tau, \nabla G_\tau(Y_\tau^w)\rangle$$

Finally, if the guidance $w = w_\theta$ is parameterized by a neural network with parameters θ, the full gradient is obtained via the chain rule:

$$\frac{dR}{d\theta} = \mathbb{E}\int_0^T \frac{\partial R(w)}{\partial w_\tau}\frac{\partial w_\tau}{\partial \theta}\, d\tau.$$

In practice, we avoid the explicit and computationally expensive formation of $A_t, \nabla^2 G_t \in \mathbb{R}^{D\times D}$ by leveraging Vector-Jacobian Products (VJPs). Pseudocode for this efficient implementation is provided in Appendix C.

6 EXPERIMENTS

HJB simulation. As a first example, we consider a mixture $p(x)$ of four Isotropic Gaussians with variance 0.5. One is centered at the origin, and the other three are centered on an equilateral triangle around the origin. The distribution $p(x|c)$ is the Gaussian at the origin. To obtain the optimal guidance strength in this case, we solve the Hamilton-Jacobi-Bellman PDE directly. We visualize the optimal guidance weights for different times in Figure 1. We would like to highlight that the optimal schedule depends on both time t and space x, and is not a constant.

EDM2 We evaluated our guidance optimization framework on the EDM2 model (Karras et al., 2024) (CC BY-NC-SA 4.0), which was pre-trained on the ImageNet dataset (Deng et al., 2009). Specifically, we conducted experiments using a size-M model at a 512×512 resolution and a size-S model at 64×64. We fine-tuned the time-dependent guidance $w_\theta(t)$ by maximizing the reward (20) via the Adjoint Method (detailed in Section 5) with regularization parameters $\alpha \in \{2.5, 5.0, 10.0\}$. The guidance scale was optimized

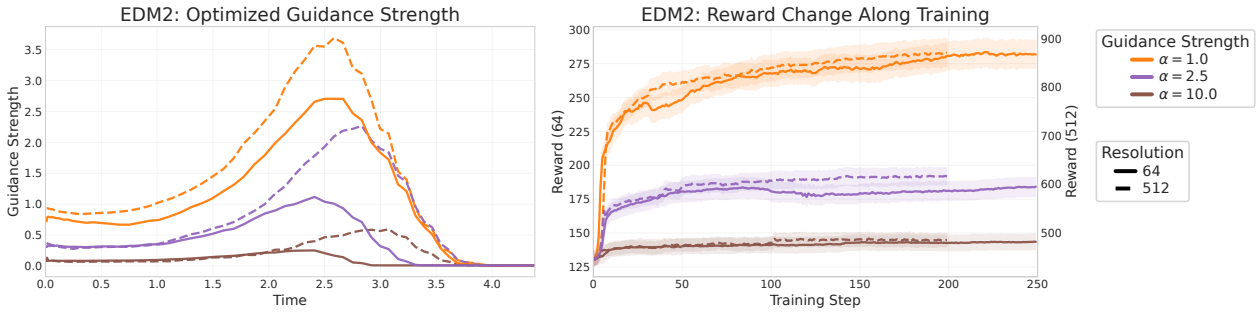


Figure 2: EDM2. Left: Guidance learned at the end of the training. Right: Change of the reward function $R(w_\theta)$ during the training with ± 1 standard deviation.

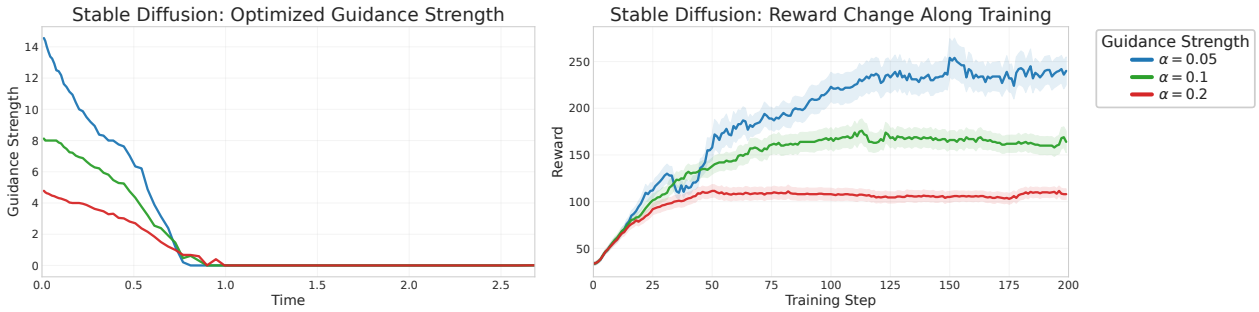


Figure 3: Stable Diffusion. Left: Guidance learned at the end of the training. Right: Change of the reward function $R(w_\theta)$ during the training with ± 1 standard deviation.

for 25 epochs (20 epochs for the 512×512 model) using 10 gradient steps per epoch, processing all 1,000 labels once per epoch. Additional implementation details are provided in Appendix B.

Table 1 compares the Fréchet Inception Distance (FID) and external classifier, InceptionV3 (Szegedy et al., 2016), confidence between our trained guidance schedule and a constant α^{-1} baseline. This baseline represents the naive optimal strategy $w = \arg\max(1 + 2w - \alpha w^2)$ derived from Equation (20). As shown, the trained schedule yields slightly higher classifier confidence at the cost of a marginally higher FID. Figure 2 illustrates the reward curve alongside the learned time-dependent guidance. The optimized guidance follows a bell-shaped trajectory, starting near α^{-1} at $t = 0$ and decaying close to zero for $t > 4$.

Stable Diffusion We extended our evaluation by training time-dependent guidance for Stable Diffusion 1.5 (Rombach et al., 2022) (CreativeML-OpenRail-M license) on both the ImageNet and COCO-captions (Chen and Zitnick, 2015) (CC BY 4.0) datasets. The training procedure mirrored the EDM2 setup, with full details available in Appendix B.

For both datasets, the optimized guidance profiles differ considerably, as depicted in Figures 3 and 5. In all configurations, the guidance strength begins near α^{-1} at $t = 0$, monotonically decreases to 0 over the interval $[0, 1]$, and remains at 0 for $t > 1$. A plausible explanation for this behavior is the difference in conditioning mechanisms: while EDM2 relies on discrete, well-separated class labels, Stable Diffusion utilizes prompts represented as continuous embeddings in a high-dimensional space. In the first case, applying guidance may cease to provide further benefits once the conditional class is established earlier in the process.

7 LIMITATIONS AND FUTURE WORK

A natural progression of this work is to train a comprehensive model where guidance dynamically depends on all variables: time, prompt, and the current state. Currently, the primary obstacle to this approach is the observed instability in the gradients of the score network. We leave this issue for future research.

Additionally, a limitation of our proposed guidance optimization method is its assumption of perfect score estimation by the diffusion model. In practice, this assumption is violated, which may inadvertently compound the errors of the estimator.

Table 1: Trained guidance perfomance for EDM2 model. Compared to constant α^{-1} guidance, the corresponding trained one ensures slightly higher Classifier (Inception V3) confidence on the cost of slightly higher FID.

Model Size	α	(↓) **FID**	(↑) **Classifier Confidence**
S	10	4.86 (4.14)	0.821 (0.817)
S	5	7.16 (5.67)	0.836 (0.831)
S	2.5	10.01 (8.45)	0.842 (0.841)
M	10	4.46 (3.99)	0.844 (0.843)
M	5	6.62 (5.52)	0.835 (0.829)
M	2.5	9.13 (8.26)	0.844 (0.843)

8 CONCLUSION

In this work, we address critical theoretical gaps in the understanding of guidance within diffusion models. Our analysis rigorously establishes that guidance not only enhances alignment with conditioning signals but also crucially preserves the support of the conditional data manifold. Leveraging these foundational insights, we formulate the scheduling of guidance strength as a stochastic optimal control problem and propose practical optimization algorithms based on adjoint state method. This approach facilitates the development of dynamic, sample- and time-dependent guidance policies that can be efficiently trained and deployed. Overall, our work provides a robust theoretical and algorithmic foundation for adaptive guidance, opening new directions for more controlled and effective generation in diffusion models.

Acknowledgements

We thank Modal for providing the computational infrastructure used in our experiments.
IA was supported by the Engineering and Physical Sciences Research Council [grant number EP/T517811/1]. PP and QL are supported by the EPSRC CDT in Modern Statistics and Statistical Machine Learning (EP/S023151/1)
GD was supported by the Engineering and Physical Sciences Research Council [grant number EP/Y018273/1]. JR received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 834175)

References

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

Bao, F., Zhao, M., Hao, Z., Li, P., Li, C., and Zhu, J. (2022). Equivariant energy-guided sde for inverse molecular design. *arXiv preprint arXiv:2209.15408*.

Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. (2023). Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

Boucheron, S., Lugosi, G., and Massart, P. (2013). *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.

Bradley, A. and Nakkiran, P. (2024). Classifier-free guidance is a predictor-corrector. *arXiv preprint arXiv:2408.09000*.

Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.-H., Murphy, K., Freeman, W. T., Rubinstein, M., et al. (2023). Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*.

Chen, X. and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Chidambaram, M., Gatmiry, K., Chen, S., Lee, H., and Lu, J. (2024). What does guidance do? a fine-grained analysis in a simple setting. *arXiv preprint arXiv:2409.13074*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

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

Domingo-Enrich, C., Drozdzal, M., Karrer, B., and Chen, R. T. Q. (2025). Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control.

Fleming, W. and Soner, H. (2006). *Controlled Markov Processes and Viscosity Solutions*. Applications of mathematics. Springer.

Gao, S., Zhou, P., Cheng, M.-M., and Yan, S. (2023). Mdtv2: Masked diffusion transformer is a strong image synthesizer. *arXiv preprint arXiv:2303.14389*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. (2022). Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646.

Karatzas, I. and Shreve, S. (1991). *Brownian Motion and Stochastic Calculus*. Graduate Texts in Mathematics (113) (Book 113). Springer New York.

Karatzas, I. and Shreve, S. E. (1998). *Brownian Motion and Stochastic Calculus*, volume 113 of *Graduate Texts in Mathematics*. Springer New York.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models.

Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. (2024). Analyzing and improving the training dynamics of diffusion models. In *Proc. CVPR*.

Kynkäänniemi, T., Aittala, M., Karras, T., Laine, S., Aila, T., and Lehtinen, J. (2024). Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*.

Li, G. and Jiao, Y. (2025). Provable efficiency of guidance in diffusion models for general data distribution.

Li, M., Cai, T., Cao, J., Zhang, Q., Cai, H., Bai, J., Jia, Y., Li, K., and Han, S. (2024). Distrifusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7183–7193.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. (2023). Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Revuz, D. and Yor, M. (2013). *Continuous Martingales and Brownian Motion*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg.

Robbins, H. E. (1956). An empirical bayes approach to statistics. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494.

Schiff, Y., Sahoo, S. S., Phung, H., Wang, G., Boshar, S., Dalla-torre, H., de Almeida, B. P., Rush, A., Pierrot, T., and Kuleshov, V. (2025). Simple guidance mechanisms for discrete diffusion models.

Schneuing, A., Harris, C., Du, Y., Didi, K., Jamasb, A., Igashov, I., Du, W., Gomes, C., Blundell, T. L., Lio, P., et al. (2024). Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4(12):899–909.

Shen, D., Song, G., Xue, Z., Wang, F.-Y., and Liu, Y. (2024). Rethinking the spatial inconsistency in classifier-free diffusion guidance.

Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al. (2022). Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*.

Skreta, M., Akhound-Sadegh, T., Ohanesian, V., Bondesan, R., Aspuru-Guzik, A., Doucet, A., Brekelmans, R., Tong, A., and Neklyudov, K. (2025). Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Stability-AI (2025). Guide to stable diffusion cfg scale parameter.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pages 2818–2826.

Weiss, T., Mayo Yanes, E., Chakraborty, S., Cosmo, L., Bronstein, A. M., and Gershoni-Poranne, R. (2023). Guided diffusion for inverse molecular design. *Nature Computational Science*, 3(10):873–882.

Zhang, L., Rao, A., and Agrawala, M. (2023). Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3836–3847.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A MISSING PROOFS AND CALCULATIONS

A.1 Proof of Lemma 1, Section 3

Proof. Consider a forward process

$$dX_t = -X_t dt + \sqrt{2} dB_t, \quad t \in [0, T]. \tag{25}$$

Let p_t be the density of the marginal X_t at time t, Fokker-Plank equation states

$$\frac{d}{dt} p_t(x) = \nabla \cdot (x p_t) + \Delta p_t = d p_t + \langle x, \nabla p_t \rangle + \Delta p_t.$$

Dividing both parts by p_t

$$\frac{d}{dt} \log p_t(x) = d + \langle x, \nabla \log p_t \rangle + \frac{\Delta p_t}{p_t} = d + \langle x, \nabla \log p_t \rangle + \Delta \log p_t + \|\nabla \log p_t\|^2$$

So, substituting $p_t(x)$ and $p_t(x|c)$ we get

$$\begin{aligned} \frac{d}{dt} G_t(x) &= \frac{d}{dt} \Big[\log p_{T-t}(x|c) - \log p_{T-t}(x) \Big] \\ &= - \langle x, \nabla G_t(x) \rangle - \Delta G_t(x) + \|\nabla G_t(x)\|^2 - 2 \langle \nabla \log p_{T-t}(x|c), \nabla G_t(x) \rangle . \end{aligned} \tag{26}$$

Let $\mu_t^w(x) = x + 2\nabla \log p_{T-t}(x|c) + 2w \nabla G_t(x)$ denote the drift of Y_t^w. Applying Ito's lemma, and then substituting (26) we finish the proof

$$\begin{aligned} dG_t(Y_t^w) &= \left[\frac{d}{dt} G_t(Y_t^w) + \langle \mu_t^w(Y_t^w), \nabla G_t(Y_t^w) \rangle + \Delta G_t(Y_t^w) \right] dt + \sqrt{2} \nabla G_t(Y_t^w) \cdot dB_t \\ &= (1 + 2w_t) \|\nabla G_t(Y_t^w)\|^2 dt + \sqrt{2} \nabla G_t(Y_t^w) \cdot dB_t. \end{aligned}$$

□

A.2 Proof of Corollary 2, Section 3

Proof. First, we verify that S_t is the stochastic exponential of the martingale $-\sqrt{2} \nabla G_s(Y_s^w) \cdot dB_s$. Recall that

$$S_t = \exp\left(-G_t + \int_0^t 2w_s \|\nabla G_s\|^2 ds \right).$$

Let

$$I_t := -G_t + \int_0^t 2w_s \|\nabla G_s\|^2 ds,$$

note that

$$d\langle I \rangle_t = d\langle G \rangle_t = 2 \|\nabla G_t\|^2 dt.$$

By Ito's lemma and Lemma 1 we get

$$\begin{aligned} dS_t &= S_t \, dI_t + \frac{1}{2} S_t \, d\langle I \rangle_t = S_t \Big(-dG_t + 2w_t \|\nabla G_t\|^2 dt \Big) + \frac{1}{2} S_t \cdot 2 \|\nabla G_t\|^2 dt \\ &= -S_t \sqrt{2} \nabla G_t \cdot dB_t, \end{aligned}$$

implying that S_t is indeed the exponent of $-\sqrt{2} \nabla G_s(Y_s^w) \cdot dB_s$. It is known that stochastic exponent is a supermartingale (Karatzas and Shreve, 1991, 2.28) and Doob's inequality (Revuz and Yor, 2013, II.1.15) for supermartingales states that

$$\mathbb{P}\left(\sup_{t \in [0,T]} S_t \geq \lambda \right) \leq \lambda^{-1} \mathbb{E} \, S_0$$

Substituting S_t we get

$$\mathbb{P}\left(\sup_{t\in[0,T]} \frac{p(c)}{p(c|Y_t^w)} \exp\left(\int_0^t 2w_t \left\|\nabla G_s\right\|^2 ds\right) \geq \lambda\right) \leq \lambda^{-1}\, \mathbb{E}\, \frac{p(c)}{p(c|Y_0^w)} = \lambda^{-1},$$

which is equivalent to the first statement of Corollary 2.

Next, we show the second statement. In general, stochastic exponential is only a local martingale, however, it is a true martingale under Novikov's condition (Karatzas and Shreve, 1991, 5.12), which in our case, means that we need to check whether

$$\mathbb{E} \exp\left(\int_0^T \left\|\nabla G_t(Y_w^t)\right\|^2 dt\right) < \infty$$

As we showed in (17) for $t < T$

$$\left\|\nabla G_t(x)\right\|^2 \leq \frac{4e^{2(t-T)}}{(1-e^{2(t-T)})^2} R^2 < \infty,$$

so S_t is a true martingale on $[0, T)$. We note that so far we are initial condition agnostic, so the same argument works in a more general case when the reference process Y_t^w is substituted with $\bar{Y}_t^w = (Y_t^w | Y_{t_0}^w = x)$ satisfying

$$\begin{cases} d\bar{Y}_t^w = \left[Y_w^t + 2\nabla \log p_{T-t}(\bar{Y}_t^w|c) + 2w\nabla G_t(\bar{Y}_t^w)\right] ds + \sqrt{2} dB_t, \quad t \in (t_0, T) \\ \bar{Y}_{t_0}^w = x, \end{cases}$$

for $0 \leq t_0 < T$. So, the process

$$\exp\left(-G_t(\bar{Y}_t^w) + \int_{t_0}^t 2w_s \left\|\nabla G_s(\bar{Y}_t^w)\right\|^2 ds\right)$$

is still a true martingale on $[t_0, T)$ and as result, we get the second part stating that for $t \in [t_0, T)$

$$\mathbb{E}\left[p_{T-t}^{-1}(c|Y_t^w) \exp\left(\int_{t_0}^t 2w \left\|\nabla G_s\right\|^2 ds\right) \middle| Y_t^w = x\right] = p_{T-t_0}^{-1}(c|x).$$

□

A.3 Proof of Theorem 3 and Remark 4, Section 4

The SDE (5) governing process Y^w is only well defined on the semi-interval $[0, T)$, so to prove that $Y_T^w \in \operatorname{supp} p(\cdot|c)$ we need to show two statements: (i) Y^w can be continuously extended on $[0, T]$, i.e. $Y_T^w = \lim_{t\to T} Y_t^w$ exists a.s. (ii) The constructed Y_T^w belongs to $\operatorname{supp} p(\cdot|c)$. We address both problems by the application of Donsker and Varadhan's variational formula.

We recall that the marginal of the forward process is represented as

$$X_t \overset{\mathcal{D}}{=} e^{-t} X_0 + \sqrt{1 - e^{-2t}} Z,$$

where $Z \sim \mathcal{N}(0, \mathrm{Id}_d)$. Therefore, to be consistent in X_0 we consider the process $e^{T-t} Y_t^w$ instead and define $Y_T^w = \lim_{t\to T} e^{T-t} Y_t^w$. Note that this limit (if it exists) is equal to $\lim_{t\to T} Y_t^w$, since $\lim_{t\to T} e^{T-t} = 1$.

Let d be the dimension of $X_0 \in \mathbb{R}^d$. We use $A^c = \mathbb{R}^d \backslash A$ to denote the complement of set $A \subset \mathbb{R}^d$ and $cA = \{ca : a \in A\}$ for a scalar $c \in \mathbb{R}$.

Let $\Omega := \operatorname{supp} p_0(\cdot|c)$ denotes the support and $\Omega_\varepsilon := \left\{x \in \mathbb{R}^d : \inf_{\omega\in\Omega} \|x - \omega\| \leq \varepsilon\right\}$ denotes the ε-neighborhood of Ω.

We prove in full generality, assuming that there are constants $\infty > C_{\min}, C_{\max} > 0$

$$-\frac{1}{2} + C_{\min} \leq w \leq C_{\max}$$

Then, using the same reasoning as in Section 4,

$$\mathsf{KL}(Y^w_{T-\varepsilon}\|Y^0_{T-\varepsilon}) \le \mathsf{KL}(Y^w_{[0,T-\varepsilon]}\|Y^0_{[0,T-\varepsilon]}) \le \max\left(C_{\max}, \frac{C^2_{\max}}{2C_{\min}}\right)\log p^{-1}(c) := C_{\mathsf{KL}} < \infty.$$

First we prove the following lemma

Lemma 5. *Under the same conditions as in Theorem 3, for any $\varepsilon > 0$*

$$\lim_{t\to T}\mathbb{P}\big(e^{T-t}Y^w_t \in \Omega_\varepsilon\big) = 1. \tag{27}$$

Proof. We recall (Boucheron et al., 2013, Corollary 4.15) the duality formula for KL stating that for two probability distributions P and Q

$$\mathsf{KL}(Q\|P) = \sup_{Z:\mathbb{E}_P e^Z<\infty} \mathbb{E}_Q Z - \log \mathbb{E}_P e^Z.$$

Substituting $Z = \alpha \mathbb{1}_{e^{t-T}\Omega^c_\varepsilon}$ for some $\alpha > 0$ and as Q and P laws of Y^w_t and Y^0_t we get

$$\mathsf{KL}\big(Y^w_t\|Y^0_t\big) + \log[\mathbb{P}\big(Y^0_t \in e^{t-T}\Omega_\varepsilon\big) + e^\alpha\,\mathbb{P}\big(Y^0_t \notin e^{t-T}\Omega_\varepsilon\big)] \ge \mathbb{P}\big(Y^w_t \notin e^{t-T}\Omega_\varepsilon\big)$$

So, for any $\alpha > 0$

$$\begin{aligned}\frac{C_{\mathsf{KL}} + \log[1 + e^\alpha\,\mathbb{P}\big(Y^0_t \notin e^{t-T}\Omega_\varepsilon\big)]}{\alpha} &\ge \frac{\mathsf{KL}\big(Y^w_t\|Y^0_t\big) + \log[\mathbb{P}\big(Y^0_t \in e^{t-T}\Omega_\varepsilon\big) + e^\alpha\,\mathbb{P}\big(Y^0_t \notin e^{t-T}\Omega_\varepsilon\big)]}{\alpha}\\ &\ge \mathbb{P}\big(Y^w_t \notin e^{t-T}\Omega_\varepsilon\big).\end{aligned} \tag{28}$$

Finally, we note that Y^0_t is a true backward process, so $Y^0_t \overset{\mathcal{D}}{=} e^{t-T}Y^0_T + \sqrt{1-e^{2(t-T)}}Z$, where $Z \sim \mathcal{N}(0, \mathrm{Id}_D)$ and $Y^0_T \sim p(\cdot|c)$, therefore

$$\mathbb{P}\big(Y^0_t \notin e^{t-T}\Omega_\varepsilon\big) = \mathbb{P}\Big(e^{t-T}Y^0_T + \sqrt{1-e^{2(t-T)}}Z \notin e^{t-T}\Omega_\varepsilon\Big) \le \mathbb{P}\Big(\sqrt{1-e^{2(t-T)}}\,\|Z\| \ge e^{t-T}\varepsilon\Big) \to_{t\to T} 0.$$

So, taking the limit $t \to T$ we get that for all $\alpha > 0$

$$\frac{C_{\mathsf{KL}}}{\alpha} \ge \lim_{t\to T}\mathbb{P}\big(Y^w_t \notin e^{t-T}\Omega_\varepsilon\big)$$

By taking $\alpha \to \infty$ we get (27). □

Lemma 6. *Under the same conditions as in Theorem 3, the process $e^{T-t}Y^w_t$ a.s. admits a continuous extension on $[0, T]$.*

Proof. We show this by checking Cauchy type condition. We fix $T_i = T - 2^{-i}$ and prove that a.s.

$$\sum_{i=1}^\infty \sup_{s,t\in[T_i,T_{i+1}]} \left\|e^{T-s}Y^w_s - e^{T-t}Y^w_t\right\| < \infty. \tag{29}$$

Then Y^w_T, for example, can be constructed as a telescopic limit

$$Y^w_T := e^{1/2}Y^w_{T-1/2} + \sum_{i=1}^\infty (e^{T-T_{i+1}}Y^w_{T_{i+1}} - e^{T-T_i}Y^w_{T_i}),$$

and (29) will guarantee that the sum converges and it does not depend on the choice of $\{T_i\}$.

The forward OU process X_t can be represented as a rescaled Wiener process, more precisely

$$X_t = e^{-t}(X_0 + W_{e^{2t}-1}).$$

So, for $0 \le A < B < \infty$

$$\sup_{s,t\in[A,B]} \left\|e^tX_t - e^sX_s\right\| = \sup_{s,t\in[A,B]} \|W_{e^{2t}-1} - W_{e^{2s}-1}\| \le 2\sup_t \|W_{e^{2t}-1} - W_{e^{2A}-1}\|,$$

and by the reflection principle for $\delta < 1/4$

$$\mathbb{P}\left(\sup_{s,t\in[A,B]} \left\|e^t X_t - e^s X_s\right\| \geq 2\sqrt{2d(\log 8d)(e^{2B} - e^{2A})\log\delta^{-1}}\right)$$
$$\leq \mathbb{P}\left(2\sup_t \|W_{e^{2t}-1} - W_{e^{2A}-1}\| \geq 2\sqrt{2d((e^{2B}-1)-(e^{2A}-1))\log(2d\delta^{-1})}\right) \leq \delta,$$

where we additionally used $\log(2d\delta^{-1}) < (\log 8d)\log\delta^{-1}$. So, for $A, B \leq 1/2$

$$\mathbb{P}\left(\sup_{s,t\in[A,B]} \left\|e^t X_t - e^s X_s\right\| \geq 8\sqrt{d(\log 8d)(B-A)\log\delta^{-1}}\right) \leq \delta.$$

As result, by (28)

$$\frac{C_{\mathsf{KL}} + \log[1 + e^{\alpha_n}\delta_n]}{\alpha_n} \geq \mathbb{P}\left(\sup_{s,t\in[T_i,T_{i+1}]} \left\|e^{T-s}Y_s^w - e^{T-t}Y_t^w\right\| \geq 8\sqrt{d(\log 8d)(T_i - T_{i+1})\log\delta_n^{-1}}\right). \tag{30}$$

We take $\alpha_n = \gamma^{-1}n^2$ and $\delta_n = \exp\left(-\gamma^{-1}n^2\right)$, so

$$\gamma\frac{C_{\mathsf{KL}} + \log 2}{n^2} \geq \mathbb{P}\left(\sup_{s,t\in[T_i,T_{i+1}]} \left\|e^{T-s}Y_s^w - e^{T-t}Y_t^w\right\| \geq 8\sqrt{d(\log 8d)(T_i - T_{i+1})\gamma^{-1}n^2}\right)$$

Summing over all T_i, since $T_{i+1} - T_i = 2^{-(i+1)}$ and

$$\sum \frac{n}{2^{n/2}} = 4 + 3\sqrt{2} < 10, \quad \sum \frac{1}{n^2} = \frac{\pi^2}{6},$$

we get the union bound

$$\gamma(C_{\mathsf{KL}} + \log 2)\frac{\pi^2}{6} > \mathbb{P}\left(\sum_i \sup_{s,t\in[T_i,T_{i+1}]} \left\|e^{T-s}Y_s^w - e^{T-t}Y_t^w\right\| \geq 80\sqrt{\gamma^{-1}}\sqrt{d\log 8d}\right).$$

By taking $\gamma \to 0$ we show (29). □

Since almost sure convergence implies weak convergence and Ω_ε is closed

$$1 = \lim_{t\to T}\mathbb{P}(Y_t^w \in \Omega_\varepsilon) = \limsup_{t\to T}\mathbb{P}(Y_t^w \in \Omega_\varepsilon) \leq \mathbb{P}(Y_T^w \in \Omega_\varepsilon).$$

We finish the proof of Theorem 3 and Remark 4 by taking $\varepsilon \to 0$.

A.4 Omitted Calculations in Section 4

In this Section we present calculations ommited in Section 4.

We recall that the sensitivity process $Z_t(\tau)$ is defined as

$$Z_t(\tau) = \frac{\partial Y_t^w}{\partial w_\tau}.$$

Applying the chain rule to the reward $R(w)$ we obtain

$$\begin{aligned}
\frac{\partial R(w)}{\partial w_\tau} &= \frac{\partial}{\partial w_\tau}\mathbb{E}\int_0^T (1 + 2w_t - \alpha w_t^2)\left\|\nabla G_t(Y_t^w)\right\|^2 dt \\
&= \mathbb{E}\int_0^T 2\cdot\delta(\tau - t)(1 - \alpha w_t)\left\|\nabla G_t(Y_t^w)\right\|^2 + 2(1 + 2w_t - \alpha w_t^2)\nabla^T G_t(Y_t^w)\nabla^2 G_t(Y_t^w)\frac{\partial Y_t^w}{\partial w_\tau}dt \\
&= 2\,\mathbb{E}(1 - \alpha w_\tau)\left\|\nabla G_t(Y_t^w)\right\|^2 + \mathbb{E}\int_\tau^T 2(1 + 2w_t - \alpha w_t^2)\nabla^T G_t(Y_t^w)\nabla^2 G_t(Y_t^w)Z_t(\tau)dt
\end{aligned} \tag{31}$$

Next, using the chain rule, we write the differential equation on the sensitivity process

$$\begin{aligned} dZ_t(\tau) = d\frac{\partial Y_t^w}{\partial w_\tau} &= \frac{\partial}{\partial w_\tau}\Big(\left[Y_w^t + 2\nabla \log p_{T-t}(Y_t^w|c) + 2w_t \nabla G_t(Y_t^w)\right] dt + \sqrt{2} dB_t\Big) \\ &= \left[Z_t(\tau) + 2\nabla^2 \log p_{T-t}(Y_t^w|c) Z_t(\tau) + 2\delta(\tau - t)\nabla G_\tau(Y_\tau^w) + 2w_t \nabla^2 G_t(Y_t^w) Z_t(\tau)\right] dt \end{aligned}$$

implying that it satisfies ODE

$$\begin{cases} dZ_t(\tau) = \left[\mathrm{Id} + 2\nabla^2 \log p_{T-t}(Y_t^w|c) + 2w_t \nabla^2 G_t(Y_t^w)\right] Z_t(\tau) dt, \\ Z_\tau = 2\nabla G_\tau(Y_\tau^w). \end{cases} \tag{32}$$

We then consider an adjoint backward process given by

$$\begin{cases} d\lambda_t = -\underbrace{\left[\mathrm{Id} + 2\nabla^2 \log p_{T-t}(Y_t^w|c) + 2w_t \nabla^2 G_t(Y_t^w)\right]}_{:=A_t}{}^T \lambda_t dt - \underbrace{2(1 + 2w_t - \alpha w_t^2)\nabla^T G_t(Y_t^w)\nabla^2 G_t(Y_t^w)}_{:=B_t} dt \\ \lambda_T = 0 \end{cases} \tag{33}$$

Then, integrating by parts, and noting that $\langle x, Wy\rangle = \langle W^T x, y\rangle$, we see that

$$\begin{aligned} \langle \lambda_\tau, 2\nabla G_\tau(Y_\tau^w)\rangle = \langle \lambda_\tau, Z_\tau\rangle = \langle \lambda_\tau, Z_\tau\rangle - \langle \lambda_T, Z_T\rangle &= -\int_\tau^T \left(\left\langle \frac{d}{dt}\lambda_t, Z_t\right\rangle + \left\langle \lambda_t, \frac{d}{dt} Z_t\right\rangle\right) dt \\ = -\int_\tau^T \left(-\left\langle {A_t}^T \lambda_t - B_t^T, Z_t\right\rangle + \langle \lambda_t, A_t Z_t\rangle\right) dt &= \int_\tau^T 2(1 + 2w_t - \alpha w_t^2)\nabla^T G_t(Y_t^w)\nabla^2 G_t(Y_t^w) Z_t dt \end{aligned}$$

that simplifies (31) and results in

$$\frac{\partial R(w)}{\partial w_\tau} = 2\,\mathbb{E}(1 - \alpha w_\tau)\,\|\nabla G_t(Y_t^w)\|^2 + 2\langle \lambda_\tau, \nabla G_\tau(Y_\tau^w)\rangle$$

B ADDITIONAL EXPERIMENTS & DETAILS

B.1 Additional Experimental Details

Model specifications. We consider two pretrained diffusion model families: Stable Diffusion v1.5 and EDM2. Stable Diffusion is used in its standard latent-diffusion form, with denoising performed in a $4 \times 64 \times 64$ latent space corresponding to 512×512 images, using the model's pretrained VAE encoder and decoder without modification. EDM2 is evaluated on pretrained ImageNet checkpoints; the 512×512 models are likewise represented in a $4 \times 64 \times 64$ latent space, whereas the 64×64 models operate directly in pixel space and therefore do not use a VAE. In all cases, the underlying generative model is frozen and only the scalar guidance schedule is trained.

Data For Stable Diffusion experiments on the ImageNet dataset, conditioning prompts are constructed from the class names using the template "a photo of {class name}", with underscores replaced by spaces. Training was performed on all 1000 ImageNet classes, with 20 holdout prompts fixed to track performance during training. For EDM2, conditioning is performed directly using ImageNet class labels rather than text prompts.

Sampling and guidance schedule parameterization. In all experiments, reverse diffusion uses 64 steps during training and 32 steps during evaluation. Sampling uses Heun discretization, with an early-stopping cutoff of $t = 0.01$ during training and 2×10^{-6} during evaluation. For Stable Diffusion, the full diffusion process is discretized into 1000 timesteps, which index the noise schedule of the model; the sampled timesteps are chosen by taking evenly spaced indices from this ordered sequence after applying the early-stopping cutoff. For EDM2, we first construct a dense timestep sequence from the VE noise schedule used in the EDM2 setup, with $\sigma_{\max} = 80$, $\sigma_{\min} = 0.002$, $\rho = 7$, and 1025 base noise levels, convert it to VP time by SNR matching, and then select the 64 or 32 sampling steps by the same evenly spaced indexing procedure.

Figure 4: Images sampled using Stable Diffusion with guidance trained on the COCO dataset. Captions are taken from the holdout set. All images sampled using 100 points. The list of corresponding prompts from left to right: "The polar bear is looking at his latest treat.", "A plate of fruit such as bananas, lemons, apples, and oranges.", "A man stands between two railroad tracks as a train approaches", "Open book with a magazine cover of motorcycle", "a couple of horses grazing on some green grass".

Guidance parameterization. The guidance schedule is parameterized by one scalar parameter for each of the 64 training timesteps, so the learned schedule consists of 64 parameters in total, each associated with a specific point on the training-time sampling grid. To ensure that the guidance scale remains positive, each parameter is passed through a softplus transformation before being used at runtime. During evaluation, the sampling grid contains 32 timesteps rather than 64; when an evaluation timestep does not exactly coincide with a training-grid point, it is mapped to the nearest training timestep, and the corresponding learned scalar is used.

Training details. For Stable Diffusion, the default training configuration uses 20 epochs, batch size 20, micro-batch size 5, Adam, learning rate 0.1, and gradient clipping at 10.0. For EDM2, the default training configuration uses 20 epochs for 512×512 model and 25 epochs for 64×64 model, batch size 100, micro-batch size 25, SGD, learning rate 0.01, and gradient clipping at 10.0. In both cases, antithetic sampling is enabled during training. Mixed precision is enabled automatically on CUDA devices, using `bfloat16` when available and otherwise `float16`.

Evaluation details. Evaluation follows the same large-scale class-conditional protocol used in the EDM2 setup: 50,000 generated samples over 1000 ImageNet classes, with 50 images per class and random seed 0. We report FID, FD-DINOv2, Inception accuracy, and Inception confidence. For each setting, the learned timestep-dependent schedule corresponding to regularization parameter α is compared against a constant-guidance baseline with guidance strength α^{-1}.

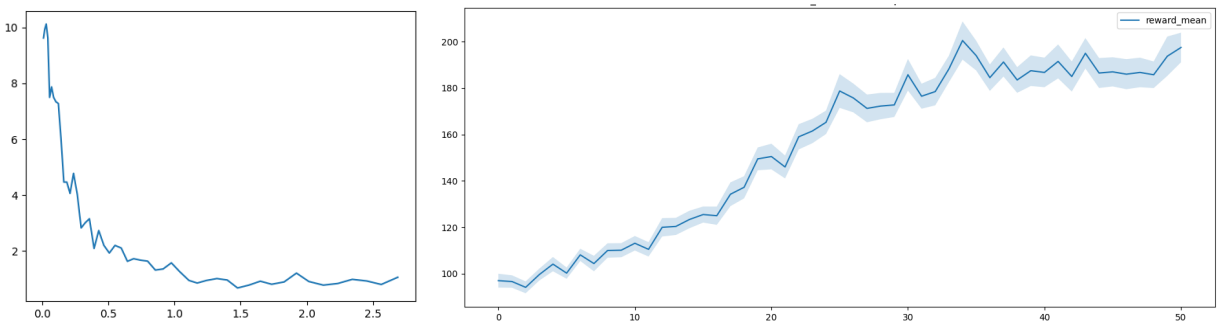


Figure 5: Guidance optimization for COCO dataset. Left: Guidance learned in 50 training steps. Right: Change of the mean reward $R(w_\theta)$ over training steps together with ± 1 standard deviation.

B.2 COCO experiment

We additionally conducted a small-scale experiment on the COCO-captions (Chen and Zitnick, 2015)(CC BY 4.0) dataset. We trained time-dependent guidance for Stable Diffusion 1.5 (Rombach et al., 2022)(CreativeML-OpenRail-M license). We set the regularization parameter to be equal to $\alpha = 0.1$ and used the first 576 prompts from the COCO-captions dataset to fine-tune guidance, while the next 32 prompts were used as a holdout set to evaluate the performance.

We trained the time-dependent guidance strength $w_t(\theta)$ for 50 gradient steps on the time interval $[0.01, 2.68]$, discretized into 50 points, by parameterizing θ with a small MLP with exp applied to the output. The batch size was chosen to be equal to 72, and the number of trajectories was 4 per prompt.

To improve stability, we have used an early stopping time $\delta = 0.01$ and applied quantile clipping while computing dR/dw_t. Finally, following Domingo-Enrich et al. (2025), we dropped the $\nabla^2 G_t \lambda_t$ term during simulation of the adjoint process.

Fig. 5 contains the learned guidance strength and the reward curve, and Fig. 4 are images sampled from the trained guided diffusion.

C PSEUDOCODE

Algorithm 1 Vector Jacobian Product (VJP)

Require: Differentiable vector field f, point y, vector v

$\phi(y) \leftarrow \langle f(y), v \rangle$

return $\nabla_y \phi(y) = \big(\nabla_y f(y)\big)^\top v$

Algorithm 2 Discrete SOC Optimization via Adjoint Method

Require: Parameters θ, time grid $0 = t_0 < t_1 < \cdots < t_N = T$, learning rate η

repeat

  Sample a minibatch of conditions $\{c\}$ and initialize the reverse process.

  Forward Pass: Simulate the controlled trajectory $\{Y_k\}_{k=0}^N$ conditioned on c on the grid using w_θ.

  Evaluate and cache discrete guidance values $w_k \leftarrow w_\theta(t_k, Y_k, c)$ and score terms $\nabla G_k \leftarrow \nabla G_{t_k}(Y_k)$.

  Backward Pass: Initialize the terminal adjoint state $\lambda_N \leftarrow 0$.

  for $k = N-1, N-2, \ldots, 0$ **do**

    $\Delta t_k \leftarrow t_{k+1} - t_k$

    Define the drift function $f_k(y) \leftarrow y + 2\nabla \log p_{T-t_k}(y \mid c) + 2w_k \nabla G_{t_k}(y)$

    Compute $A_k^\top \lambda_{k+1}$ using Algorithm 1:

      $A_k^\top \lambda_{k+1} \leftarrow \mathrm{VJP}(f_k, Y_k, \lambda_{k+1})$

    Compute the source term B_k of the adjoint ODE using AutoDiff:

      $B_k \leftarrow (1 + 2w_k - \alpha w_k^2)\nabla \|\nabla G_{t_k}(Y_{t_k}\|^2$

    Update the adjoint state via Euler integration:

      $\lambda_k \leftarrow \lambda_{k+1} + \Delta t_k (B_k + A_k^\top \lambda_{k+1})$

  end for

  Gradient Computation:

  for $k = 0, 1, \ldots, N$ **do**

    Compute the gradient of the objective w.r.t. the discrete guidance w_k:

      $g_k \leftarrow 2(1 - \alpha w_k)\|\nabla G_k\|^2 + 2\langle \lambda_k, \nabla G_k \rangle$

  end for

  Compute the full parameter gradient via the chain rule:

    $\widehat{\nabla_\theta R} \leftarrow \sum_{k=0}^N g_k \frac{\partial w_\theta(t_k, Y_k, c)}{\partial \theta} \Delta t_k$

  Perform a gradient ascent step:

    $\theta \leftarrow \theta + \eta \widehat{\nabla_\theta R}$

until convergence