INFORMATION-THEORETIC CLASSIFIER-FREE GUIDANCE WITH ADAPTIVE SCHEDULE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have achieved strong performance in image and video generation, where conditional generation is often controlled by classifier-free guidance (CFG). CFG improves condition consistency by increasing a guidance weight, but stronger guidance typically reduces distributional coverage. It remains unclear how this consistency-coverage trade-off should be controlled across the reverse diffusion trajectory, since the distribution induced by CFG is not simply the fixed-time tilted distribution given by the guided score field. To address this issue, we propose an information-theoretic framework for CFG schedule optimization. Our approach uses a clean endpoint reference to specify the desired consistency-coverage trade-off, while optimizing the actual distribution induced by the guided sampler toward this reference. We derive trajectory-level formulas to estimate the objective from samples and score evaluations, avoiding explicit density estimation. Across four diffusion backbones on ImageNet-512 and COCO, the learned schedules achieve improved consistency–FID trade-offs, with higher condition consistency at comparable FID, than constant and non-uniform guidance baselines.

1 INTRODUCTION

Diffusion models have emerged as a powerful paradigm for generative modeling. Building on denoising diffusion and score-based formulations (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2021b), they have achieved strong performance across many domains. In image generation, latent diffusion models enable high-resolution synthesis by moving the diffusion process to a learned latent space (Rombach et al., 2022), while recent large-scale systems further improve text-to-image and ultra-high-resolution image generation (Podell et al., 2024; Zhang et al., 2025). Beyond images, diffusion models have been extended to multimodal generation (Saharia et al., 2022; Ramesh et al., 2022; Ho et al., 2022b;a; Austin et al., 2021).

Much of the practical usefulness of diffusion models comes from conditional generation, where samples must follow class labels, text prompts, or other structured conditions. This control is central to modern image, text-to-image, and video generation systems, where success depends not only on quality but also on faithful alignment with the specified condition. Classifier-free guidance (CFG) has become one of the most widely used mechanisms for improving such conditional generation (Ho & Salimans, 2022), building on earlier classifier-guided sampling (Dhariwal & Nichol, 2021). Specifically, CFG combines conditional and unconditional score estimates through a guidance weight w. Larger w often improves condition consistency and perceptual quality, but this typically comes at the cost of reduced distributional coverage.

Such a nuanced trade-off has motivated methods beyond a single constant guidance weight. Some works show that applying guidance only over selected noise intervals can improve sample quality and distribution coverage over constant guidance throughout the reverse process (Kynkäänniemi et al., 2024). Other adaptive schedule designs vary guidance across denoising steps, including β-CFG and C^{2}FG (Malarz et al., 2025; Gao et al., 2026), while learned approaches can optimize an additional guidance function (Galashov et al., 2025). Prior stage-wise analyses suggest that CFG can have qualitatively different effects across noise levels, including early direction changes, intermediate mode separation, and late concentration (Jin et al., 2025). A weaker generative model can also be used as the guidance reference to improve generation (Karras et al., 2024).

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

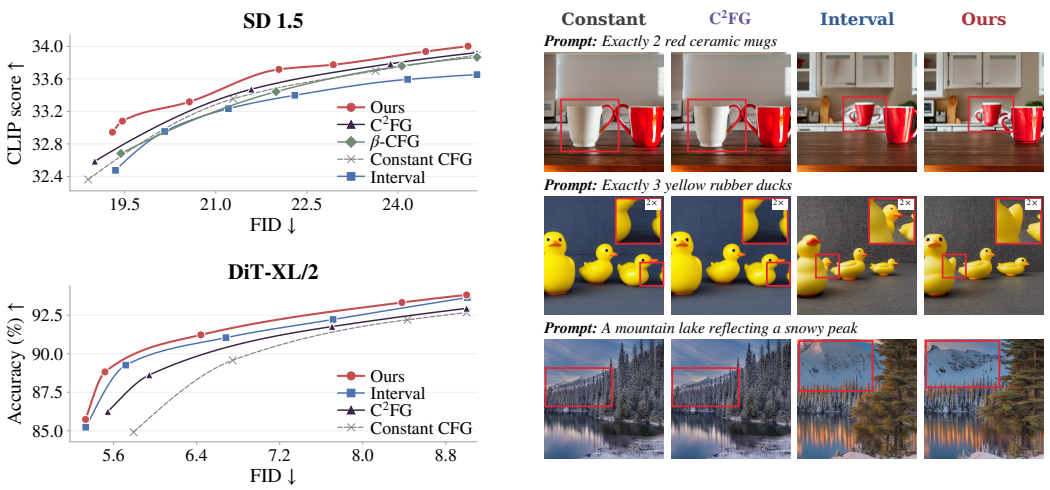


Figure 1: Overview of the empirical comparison. Left: consistency–FID trade-off curves for SD 1.5 on COCO (top) and DiT-XL/2 on ImageNet-512 (bottom); upper-left is preferred. Right: SD 1.5 qualitative comparisons among Constant CFG, C^{2}FG(Gao et al., 2026), Interval (Kynkäänniemi et al., 2024), and Ours at comparable dataset-level FID (approximately 21). Each row uses the same prompt and initial noise across methods; red boxes highlight prompt-relevant regions where the generated content differs. CLIP score and classification accuracy measure condition consistency, while FID measures distributional agreement with the data distribution; together, they characterize the evaluation metric trade-off.

However, a principled way for choosing the guidance schedule remains unclear. The main difficulty is characterizing the actual distribution induced by CFG. At a fixed time, CFG resembles a distributional tilt toward the condition, but its samples are produced by the entire reverse trajectory and need not match this fixed-time tilted interpretation (Bradley & Nakkiran, 2024; Chidambaram et al., 2024; Moufad et al., 2025). A further practical challenge is that the generated distribution is implicit: during generation, we typically have only samples and score evaluations, rather than an explicit density, making consistency-based objectives difficult to estimate directly.

This leads to the central question of this paper:

Can we find a principled approach that balances the trade-off between consistency and coverage in CFG schedules?

We address this question by using a clean endpoint reference to specify the desired consistency-coverage trade-off, and by optimizing the actual distribution induced by the guided sampler toward this reference. The resulting objective combines two information-theoretic quantities evaluated under the actual CFG distribution: a consistency term, which measures alignment between generated samples and the condition, and a coverage term, which penalizes deviation from the original data distribution. Optimizing this trajectory-level objective yields an adaptive guidance schedule. Figure 1 provides a preview of our empirical results, demonstrating improved condition and prompt consistency at comparable FID scores. The full evaluation is given in Section 5. Our main contributions include:

- We use a clean endpoint reference to specify the desired consistency-coverage trade-off, and formulate CFG schedule optimization as moving the actual distribution induced by the guided sampler toward this reference.
- We derive trajectory-level formulas for estimating the consistency and coverage terms in the objective from samples and score evaluations, without explicit density estimation.
- We directly optimize guidance weights across sampler steps using the actual CFG trajectories and frozen model scores, without prescribing a schedule shape or using evaluation metrics. Across four diffusion backbones on ImageNet-512 and COCO, the learned schedules improved evaluation metric trade-offs against constant and non-uniform guidance baselines.

2 PRELIMINARIES

2.1 DIFFUSION MODELS AND CLASSIFIER-FREE GUIDANCE (CFG)

We consider the variance exploding (VE) formulation of score-based diffusion models (Song & Ermon, 2019; Song et al., 2021b). Let $(X_0, Y) \sim p_0(x_0, y)$ denote the data-label distribution over $\mathbb{R}^D \times \mathcal{Y}$. The forward-noising process is

$$X_t = X_0 + \sigma_t Z, \qquad Z \sim \mathcal{N}(0, I_D), \qquad Z \perp (X_0, Y), \qquad t \in [0, T], \tag{1}$$

where $\sigma_0 = 0$ and σ_t is nondecreasing. Equivalently,

$$p_{t|0}(x_t \mid x_0) = \mathcal{N}(x_t; x_0, \sigma_t^2 I_D). \tag{2}$$

We denote by $p_t(x_t)$ and $p_t(x_t \mid y)$ the unconditional and conditional marginals induced by this forward process at time step t. The corresponding scores are given by

$$s_t^{\mathrm{un}}(x_t) := \nabla_{x_t} \log p_t(x_t), \qquad s_t^{\mathrm{con}}(x_t, y) := \nabla_{x_t} \log p_t(x_t \mid y). \tag{3}$$

Under VE parameterization, unconditional generation starts from $X_T \sim p_T$ and integrates the probability-flow ODE backward to $t = 0$ (Song et al., 2021b):

$$\frac{dX_t}{dt} = -\sigma_t \dot{\sigma}_t \, s_t^{\mathrm{un}}(X_t), \tag{4}$$

where $\dot{\sigma}_t = d\sigma_t/dt$. Similarly, the probability-flow ODE for conditional generation can be obtained by replacing s_t^{un} with s_t^{con}.

CFG (Ho & Salimans, 2022) strengthens conditional generation by using a guidance weight $w \geq 0$. Motivated by the classifier-guidance (Dhariwal & Nichol, 2021), at each fixed noise level t, it can be interpreted as the tilted density

$$q_{t,\mathrm{tilt}}^{w}(x_t \mid y) \propto p_t(x_t)\, p_t(y \mid x_t)^w. \tag{5}$$

For $w = 1$, this reduces to the standard conditional distribution $p_t(x_t \mid y)$. Its score field recovers the standard CFG field

$$s_t^w(x_t, y) := (1 - w) s_t^{\mathrm{un}}(x_t) + w s_t^{\mathrm{con}}(x_t, y). \tag{6}$$

Substituting this guided field into the VE probability-flow ODE gives the CFG sampling dynamics

$$\frac{dX_t}{dt} = -\sigma_t \dot{\sigma}_t \, s_t^w(X_t, y). \tag{7}$$

Let $\Phi_{T\to t}^{w,y}$ denote the flow map of (7) from time T to t. With the standard initialization $X_T \sim p_T$, we define the CFG-induced marginal by

$$q_t^w(\cdot \mid y) := \left(\Phi_{T\to t}^{w,y}\right)_{\#} p_T, \tag{8}$$

where $(\cdot)_{\#}$ denotes the pushforward of a distribution. For a time-dependent schedule $\mathbf{w} = \{w_t\}_{t\in[0,T]}$, we replace the constant guidance weight by w_t, giving

$$s_t^{\mathbf{w}}(x_t, y) = (1 - w_t) s_t^{\mathrm{un}}(x_t) + w_t s_t^{\mathrm{con}}(x_t, y). \tag{9}$$

Let $\Phi_{T\to t}^{\mathbf{w},y}$ be the corresponding flow map. The sampler-induced marginal is

$$q_t^{\mathbf{w}}(\cdot \mid y) := \left(\Phi_{T\to t}^{\mathbf{w},y}\right)_{\#} p_T. \tag{10}$$

Thus, $q_t^{\mathbf{w}}$ denotes the distribution of the random variable X_t produced by the full guided trajectory, whereas x_t denotes a realization. For a constant schedule $w_t \equiv w$, we have $q_t^{\mathbf{w}} = q_t^w$.

2.2 HOW CFG CHANGES THE GENERATED DISTRIBUTION

CFG induces a consistency-coverage trade-off in conditional generation: stronger guidance can improve alignment with the condition, but may reduce coverage by moving the generated distribution farther from the original data distribution (Ho & Salimans, 2022). In particular, changing w does not

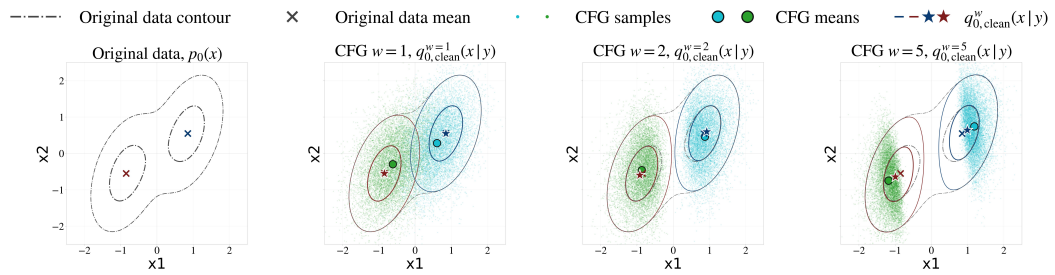


Figure 2: Mixture Gaussian visualization of CFG and clean endpoint tilting. The left panel shows the original data distribution $p_0(x)$. In the remaining panels, colored dots are CFG-generated samples, circles are empirical CFG means, and stars are the means of the clean references $q^w_{0,\text{clean}}(x_0 \mid y) \propto p_0(x_0)p_0(y \mid x_0)^w$. As w increases, the actual CFG samples separate and concentrate more aggressively than the clean references, illustrating that CFG does not simply implement direct endpoint reweighting.

simply make samples more condition-specific while keeping them near the original data distribution; it reshapes the generated distribution through the entire reverse trajectory.

To see this, we consider a two-dimensional Gaussian mixture example. Let $Y \in \{\pm 1\}$ with equal prior probability, and denote the original conditional distributions as

$$p_0(x \mid y) = \mathcal{N}(x; \mu_y, \Sigma), \qquad x \in \mathbb{R}^2, \qquad \mu_1 = (0.85, 0.55), \qquad \mu_{-1} = -(0.85, 0.55). \quad (11)$$

We apply the same forward noising process as in (1), then the noisy conditional distributions remain Gaussian $p_t(x_t \mid y) = \mathcal{N}(x_t; \mu_y, \Sigma + \sigma_t^2 I_2)$, and both s_t^{un} and s_t^{con} can be computed analytically.

The tilted distribution in (5) gives a useful reweighting interpretation of the CFG score field at each noisy level. Since samples are evaluated at the clean endpoint at $t = 0$, a natural reference for the desired consistency-coverage trade-off is the following clean endpoint tilted distribution,

$$q^w_{0,\text{clean}}(x \mid y) \propto p_0(x)p_0(y \mid x)^w. \quad (12)$$

We then examine whether the actual generated distribution $q^w_0(x_0 \mid y)$ matches either the original distribution $p_0(x_0 \mid y)$ or the clean reference $q^w_{0,\text{clean}}(x_0 \mid y)$. Figure 2 shows that the CFG distribution does not match $p_0(x_0 \mid y)$. As w increases, the CFG samples separate and concentrate more aggressively. With moderate guidance, this movement can bring the empirical CFG mean closer to the corresponding data component. With large guidance, such as $w = 5$, the sample clouds become too narrow, and some samples move into low-probability regions under the original data distribution.

The stars in Figure 2 show the means of distribution $q^w_{0,\text{clean}}(x_0 \mid y)$. This clean reference mainly suppresses mass near the overlap between the two components, so its mean moves relatively mildly and remains closer to the true mean. In contrast, CFG modifies the guided score field along the noisy reverse trajectory. Thus, although both clean endpoint tilting and CFG make the conditional samples more separated as w increases, $q^w_0(x_0 \mid y)$ does not match $q^w_{0,\text{clean}}(x_0 \mid y)$. This example motivates the distinction formalized next: fixed-time reweighting provides an interpretation of CFG, but it does not characterize the actual distribution induced by the guided reverse dynamics.

2.3 Why Tuning CFG is Difficult

The Gaussian example highlights the mismatch between the clean reference $q^w_{0,\text{clean}}(x_0 \mid y)$ and the actual CFG distribution $q^w_0(x_0 \mid y)$. As shown in Figure 2, in general,

$$q^w_0(x_0 \mid y) \neq q^w_{0,\text{clean}}(x_0 \mid y). \quad (13)$$

The source of this mismatch can be understood through two related facts: tilting before noising is different from tilting after noising, and the fixed-time tilted distribution should not be identified with the distribution induced by the guided reverse dynamics. This is consistent with recent analyses showing that guided diffusion does *not* generally sample from the tilted distribution suggested by the fixed-time guided score field (Bradley & Nakkiran, 2024; Chidambaram et al., 2024; Moufad et al., 2025). We provide the detailed derivation in Appendix D.

This mismatch makes tuning CFG difficult because a desired clean-time trade-off does not directly specify how the guidance schedule should be distributed over noise levels. The clean reference specifies what kind of output distribution we would like to obtain: samples should be compatible with the condition while remaining close to the original clean data distribution. However, CFG does not directly follow this reference distribution. It controls noisy-level score fields along the reverse trajectory, and the final distribution is the accumulated result of all guidance decisions along that trajectory. Therefore, changing w_t at one noise level can affect the later trajectory and the clean-time trade-off in a nonlocal way.

A second difficulty is that the actual CFG distribution is implicit. Even for a fixed schedule $\mathbf{w}$, the induced distributions $q_t^{\mathbf{w}}(x_t \mid y)$ and $q_0^{\mathbf{w}}(x_0 \mid y)$ are not available as explicit densities. Thus, the desired consistency-coverage trade-off cannot be evaluated by directly computing endpoint density ratios such as $\log q_0^{\mathbf{w}}(x_0 \mid y) - \log p_0(x_0)$. In practice, we only have samples generated along the reverse trajectory and score evaluations from the conditional and unconditional diffusion models.

3 An Information-Theoretic Analysis of Classifier-Free Guidance

The previous discussion suggests that a principled objective for guidance schedule optimization should satisfy two requirements. First, it should specify the desired trade-off between condition consistency and closeness to the original clean data distribution. Second, it should be evaluable under the actual CFG distribution induced by the sampler, even though this distribution is available only implicitly through generated trajectories and score functions.

In this section, we construct an objective for guidance schedule optimization. The objective uses a clean endpoint reference to specify the desired consistency-coverage trade-off and optimizes the actual sampler-induced distribution toward this reference. To make the objective estimable, we rewrite it as trajectory-level expressions along the reverse sampler. This yields two terms: a consistency term measuring compatibility with the condition, and a coverage term measuring deviation from the original clean data distribution.

3.1 Objective

We first address how to specify the desired consistency-coverage trade-off. We use the clean reference as in (12), now indexed by a reference parameter $\lambda \geq 0$:

$$q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y) \propto p_0(x_0) p_0(y \mid x_0)^{\lambda}. \tag{14}$$

Larger λ places more weight on samples that are consistent with the condition, while the factor $p_0(x_0)$ anchors the reference to the original data distribution. The parameter λ defines the clean endpoint reference; it does not affect the reverse sampler. In contrast, the guidance schedule $\mathbf{w}$ controls the sampling dynamics and induces the actual CFG distribution $q_0^{\mathbf{w}}(x_0 \mid y)$.

The next step is to evaluate whether the guided sampler realizes this desired trade-off. Since we can sample trajectories from the actual CFG distribution, we compare the sampler-induced distribution to the clean reference:

$$\mathbb{E}_{p(y)} \mathrm{KL}\big(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)\big)\,. \tag{15}$$

Note that we use the forward KL rather than the reverse KL because we do not have a sampler for the reference distribution $q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)$, which specifies the desired trade-off. In contrast, $q_0^{\mathbf{w}}(x_0 \mid y)$ is the distribution actually produced by the guided sampler, so expectations under it can be estimated from generated trajectories.

The KL in (15) also decomposes into the two quantities we want to control.

Lemma 1. *For fixed λ, minimizing $\mathbb{E}_{p(y)} \mathrm{KL}\Big(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)\Big)$ over guidance schedules w is equivalent to minimizing*

$$\mathcal{L}_0(\mathbf{w}; \lambda) := -\lambda \underbrace{\mathbb{E}_{p(y) q_0^{\mathbf{w}}(x_0 \mid y)} \left[\log p_0(y \mid x_0)\right]}_{\textit{consistency}} + \underbrace{\mathbb{E}_{p(y)} \mathrm{KL}(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, p_0(x_0))}_{\textit{coverage}}\,. \tag{16}$$

The proof of Lemma 1 is provided in Appendix E.Lemma 1 shows that matching the clean reference is equivalent to optimizing a consistency–KL trade-off. Moreover, $q_{0,\mathrm{clean}}^{\lambda}$ is the exact minimizer of

this objective when the endpoint distribution $q(x_0 \mid y)$ can be chosen freely. Figure 5 in Appendix C illustrates these two information quantities used in our objective for the Gaussian example discussed in Section 2.2. In particular, increasing w in CFG improves the consistency term but also increases the coverage KL, illustrating the trade-off induced by constant guidance. In contrast, varying λ in the clean reference yields a better trade-off: at a similar consistency level, the clean reference remains closer to the original data distribution than the actual CFG distribution.

Moreover, as shown in the following two subsections, an additional advantage of this KL formulation is that both terms can be estimated along the guided probability-flow trajectory from generated samples and score evaluations. Classical de Bruijn identities relate the evolution of information quantities to score functions (Cover & Thomas, 2006). Unlike the classical setting, however, our objective is evaluated under the sampler-induced distribution $q_t^{\mathbf{w}}$, whose density is only implicitly defined. To overcome this challenge, Lemmas 2 and 3 derive the corresponding trajectory formulas using the guided samples and model scores (Song et al., 2021a;b).

3.2 Consistency Term

The consistency term rewards samples that are compatible with the condition y under the original data distribution. Since it is difficult to evaluate directly at $t = 0$, the following lemma expresses this quantity along the guided probability-flow trajectory.

Lemma 2. *Under the VE probability-flow dynamics in* (7)*, for any* $t \in [0, T]$*,*

$$\mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[\log p_t(y \mid x_t)\right] = \mathbb{E}_{p(y)q_T^{\mathbf{w}}(x_T|y)}\left[\log p_T(y \mid x_T)\right] \tag{17}$$
$$- \int_t^T \sigma_s \dot{\sigma}_s \, \mathbb{E}_{p(y)q_s^{\mathbf{w}}(x_s|y)}\left[\nabla_x \cdot s_s^{\mathrm{diff}}(x_s, y) + \left\langle s_s^{\mathrm{diff}}(x_s, y), s_s^{\mathrm{con}}(x_s, y)\right\rangle - w_s \left\| s_s^{\mathrm{diff}}(x_s, y)\right\|^2\right] ds,$$

where $s_t^{\mathrm{diff}}(x_t, y) := s_t^{\mathrm{con}}(x_t, y) - s_t^{\mathrm{un}}(x_t)$.

Setting $t = 0$ in Lemma 2 gives the consistency term in (16). The identity has a de Bruijn-type (Cover & Thomas, 2006; Song et al., 2021a) interpretation: it tracks how the noisy consistency $\mathbb{E}[\log p_t(y \mid x_t)]$ changes along the probability-flow trajectory, instead of evaluating the quantity directly. The integral captures two effects: the divergence term measures local expansion or contraction of the condition-dependent score direction, while the inner-product term measures its interaction with the guided score field. A detailed derivation is given in Appendix F.

In practice, for a fixed schedule $\mathbf{w}$, we generate trajectories from the guided sampler and evaluate the integrand in Lemma 2. The divergence $\nabla_x \cdot s_t^{\mathrm{diff}}$ can be estimated with Hutchinson trace estimators (Hutchinson, 1990; Grathwohl et al., 2019), and the inner-product term only requires evaluations of s_t^{un}, s_t^{con}, and s_t^{diff}, which are available in CFG diffusion models.

3.3 Coverage Term

We use *coverage* to describe how well the guided distribution remains close to the original clean data distribution. The second term in (16), $\mathrm{KL}(q_0^{\mathbf{w}}(x_0 \mid y) \| p_0(x_0))$, measures this departure from $p_0(x_0)$, with larger values indicating greater deviation from the original data distribution. As with the consistency term, we avoid direct estimation by rewriting this quantity as an integral along the guided probability-flow trajectory. The following lemma allows us to estimate this KL term without directly estimating the clean densities $q_0^{\mathbf{w}}(x_0 \mid y)$ or $p_0(x_0)$.

Lemma 3. *Under the VE probability-flow dynamics in* (7)*, the KL term in* (16) *satisfies*

$$\begin{aligned}\mathbb{E}_{p(y)}\mathrm{KL}(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, p_0(x_0)) &= \mathbb{E}_{p(y)}\mathrm{KL}(q_T^{\mathbf{w}}(x_T \mid y) \,\|\, p_T(x_T)) \\ &- \int_0^T \sigma_t \dot{\sigma}_t \, \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)} w_t \left[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \left\langle s_t^{\mathrm{un}}(x_t), s_t^{\mathrm{diff}}(x_t, y)\right\rangle\right] dt.\end{aligned} \tag{18}$$

With the standard initialization $X_T \sim p_T$ in generation, the first KL term on the right-hand side is zero. We keep it in the identity to make the probability-flow relation explicit. Detailed derivations are given in Appendix G.

Remark 1 (Local effect of guidance). Holding the current sampler marginal $q_t^{\mathbf{w}}$ fixed, Lemma 2 shows that the local consistency response to increasing w_t is proportional to $\mathbb{E}_{p(y)q_t^{\mathbf{w}}}[\|s_t^{\mathrm{diff}}\|^2]$. Lemma 3

shows that the same increase in w_t also changes the KL term through $-\mathbb{E}_{p(y)q_t^{\mathbf{w}}}[\nabla_x \cdot s_t^{\text{diff}} + \langle s_t^{\text{un}}, s_t^{\text{diff}} \rangle]$, thereby affecting coverage. Therefore, the local consistency–coverage trade-off is governed by both effects, rather than solely by the magnitude of the score difference, as commonly considered in the literature (Gao et al., 2026; Malarz et al., 2025). This joint perspective motivates a new learning schedule that achieves improved performance. The exact characterization of the local response is provided in Corollary 1 in Appendix G.

4 Guidance Schedule Optimization Algorithm

We now use the trajectory objective to learn a time-dependent guidance schedule. The objective derived above is written in continuous noise time $t \in [0, T]$, while practical samplers run on a finite reverse-time grid $T = t_K > \cdots > t_1 \approx 0$. We therefore optimize one guidance value per sampling step. Let $\mathbf{w} = (w_1, \ldots, w_K)$ denote the discrete schedule, where $w_k := w(t_k)$, k indexes the discrete sampler step, and t_k is the corresponding time step.

Given a batch of conditions $\{y^{(b)}\}_{b=1}^{B}$, we generate trajectories $\{x_{t_k}^{(b)}\}_{k=1,b=1}^{K,B}$ using the current schedule. We approximate the trajectory objective by the Riemann sum

$$\widehat{\mathcal{L}}_0(\mathbf{w}; \lambda) = \sum_{k=1}^{K} \frac{a_k}{B} \sum_{b=1}^{B} \Bigg[(\lambda - w_k) \Big(\nabla_x \cdot s_{t_k}^{\text{diff}}(x_{t_k}^{(b)}, y^{(b)}) + \Big\langle s_{t_k}^{\text{diff}}(x_{t_k}^{(b)}, y^{(b)}), s_{t_k}^{\text{con}}(x_{t_k}^{(b)}, y^{(b)}) \Big\rangle \Big) + w_k(1 - \lambda) \Big\| s_{t_k}^{\text{diff}}(x_{t_k}^{(b)}, y^{(b)}) \Big\|^2 \Bigg]. \tag{19}$$

Here, a_k is the integration weight at sampling step k, accounting for the factor $\sigma_t \dot{\sigma}_t$ and the step size. We drop additive terms that do not depend on $\mathbf{w}$, since they do not affect its optimization.

We optimize this objective by iteratively updating the guidance schedule. At each iteration, we generate trajectories using the current schedule, hold them fixed, and use $\partial \widehat{\mathcal{L}}_0 / \partial w_k$ to propose updates to the guidance weights w_k. This calculation does not account for how the new weights change the sampler's later states. We therefore limit the update size and rerun the sampler with the candidate schedule to evaluate its effect. Following a trust-region approach (Schulman et al., 2015), we adjust this limit by comparing the objective decrease predicted from the fixed trajectories with the decrease measured after rerunning the sampler. Appendix H gives the derivation and the update rule.

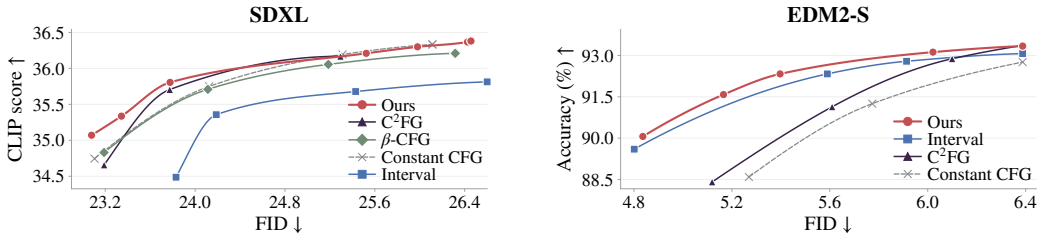
Rather than restricting the schedule to a predefined shape or a few parameters, we directly optimize the guidance weight at each sampler step where updates are allowed. This lets us change both the overall guidance strength and when guidance is stronger or weaker. Unlike *Learn to Guide Your Diffusion Model* (Galashov et al., 2025), we do not train an additional guidance function. The diffusion model stays frozen, and we learn the schedule from its CFG trajectories and conditional and unconditional scores, without real-image supervision or evaluation metrics. We examine the learned schedules and their qualitative effects in Sections 5.3 and 5.4.

5 Experiments

Our formulation uses λ to specify an endpoint trade-off and optimizes time-dependent guidance under the actual CFG dynamics. We first test whether this objective trade-off is reflected in standard evaluation metrics, then examine how the learned schedules allocate guidance along the reverse trajectory.

5.1 Experimental Setup

Models and baselines. We evaluate text-to-image generation with SD 1.5 and SDXL on COCO val2017, and class-conditional generation with DiT-XL/2 and EDM2-S on ImageNet-512. We compare against constant CFG, Interval Guidance (Kynkäänniemi et al., 2024), C^2FG (Gao et al., 2026), and β-CFG (Malarz et al., 2025), where applicable. For our method, each operating point uses a schedule optimized for a reference parameter λ. The schedules are learned from generated



(a) SDXL: CLIP–FID.

(b) EDM2-S: classification accuracy–FID.

Figure 3: Additional consistency–FID trade-off curves. Left: CLIP–FID on COCO with SDXL. Right: classification accuracy–FID on ImageNet-512 with EDM2-S. Baseline curves sweep guidance strength while keeping the schedule form fixed; our points use schedules optimized for different reference parameters λ. Upper-left is preferred.

trajectories and the conditional and unconditional scores of the frozen diffusion model; no evaluation metric is used during schedule optimization.

Evaluation protocol. For text-to-image generation, we evaluate SD 1.5 and SDXL on the same 5,000 COCO val2017 image–caption pairs using FID (Heusel et al., 2017) and CLIP score (Radford et al., 2021). Both models use 40-step deterministic sampling. For class-conditional generation, each operating point is evaluated using 10,000 ImageNet-512 samples, with 100-step deterministic DDIM sampling for DiT-XL/2 and 32-step deterministic Heun sampling for EDM2-S. We report FID, ImageNet classification accuracy, Inception Score (IS), precision, and recall. Within each backbone, competing methods use the same generation and evaluation protocol, with conditioning inputs and initial noise reused across methods where applicable. Detailed baseline configurations, optimization settings, evaluation protocols, metric definitions, and supplementary results are provided in Appendix I.

5.2 Evaluation Metric Trade-offs

The objective in Section 3 balances condition consistency against deviation from the original data distribution. We evaluate this trade-off using CLIP–FID for text-to-image generation and classification accuracy–FID for class-conditional generation. The curves in Figures 1 and 3 are obtained by varying guidance strength for the baselines and λ for our method. Because these parameters have different meanings, we compare the resulting metric trade-offs rather than matching their numerical values. Curves toward the upper-left are preferred: they offer higher condition consistency at comparable FID, or lower FID at comparable consistency.

Our learned schedules improve the measured trade-offs across all four backbones. On SD 1.5 and SDXL, they achieve higher CLIP scores at comparable FID, or lower FID at comparable CLIP. On DiT-XL/2 and EDM2-S, they achieve higher classification accuracy at comparable FID over a substantial portion of the evaluated range. These gains are measured by metrics that are not used during schedule optimization, providing external evidence that the trajectory objective yields improved evaluation metric trade-offs.

The curves compare methods across a range of operating points. To complement this comparison with concrete numerical results, Table 1 reports each method's lowest-FID operating point within its evaluated ImageNet parameter grid. Our method achieves the lowest FID on DiT-XL/2 and nearly matches Interval Guidance on EDM2-S. At these minimum-FID operating points, it also achieves the highest IS on both backbones and higher classification accuracy than Interval Guidance. Its precision is highest on both backbones, while Interval Guidance attains higher recall; full precision–recall curves are provided in Appendix I.4.

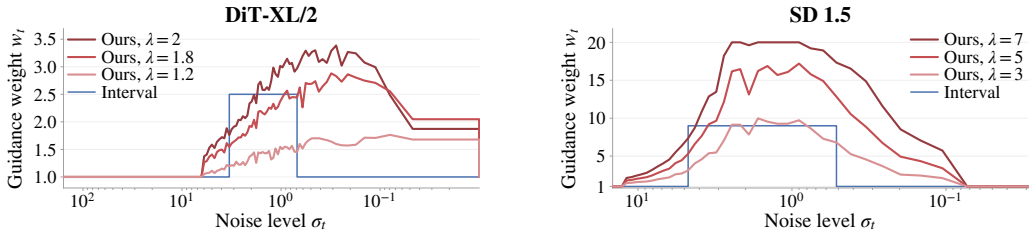
5.3 Learned Guidance Schedules

To interpret the gains in Section 5.2, we examine the learned schedules for DiT-XL/2 and SD 1.5 (Figure 4). As λ varies, these schedules change in shape as well as strength, rather than simply

Table 1: Lowest-FID operating point for each method within its evaluated ImageNet-512 parameter grid. All metrics use 10,000 generated samples.

Method	Setting	FID ↓	Acc. ↑	IS ↑	Prec. ↑	Rec. ↑
DiT-XL/2 *(DDIM, 100 steps)*						
Constant CFG	$w = 1.5$	5.79	84.91	224.18	0.799	0.704
C^2FG	$w_0 = 1.1$	5.55	**86.25**	238.99	0.822	0.695
Interval	$w = 2.5$	5.33	85.23	233.24	0.805	**0.715**
Ours	$\lambda = 1.8$	**5.29**	85.74	**240.32**	**0.828**	0.691
EDM2-S *(Heun, 32 steps)*						
Constant CFG	$w = 1.4$	5.27	88.59	209.66	0.782	0.717
C^2FG	$w_0 = 1.1$	5.12	88.41	209.39	0.780	0.722
Interval	$w = 2.1$	**4.81**	89.60	218.13	0.795	**0.747**
Ours	$\lambda = 1.45$	4.83	**90.06**	**222.97**	**0.797**	0.743

rescaling a fixed profile or reproducing the predefined window of Interval Guidance. Stage-wise ablations in Figure 9 further examine how guidance at different noise levels affects generation.



(a) DiT-XL/2 on ImageNet-512.

(b) SD 1.5 on COCO.

Figure 4: Learned guidance schedules across noise levels. Left: DiT-XL/2 on ImageNet-512. Right: SD 1.5 on COCO. Interval Guidance applies a fixed guidance strength over a prescribed noise range. The horizontal axis follows the reverse trajectory from high to low noise.

5.4 Additional Generation Visualizations

To better understand how guidance timing affects generation, we analyze both the denoising trajectory and final samples. Appendix I.6 shows that strong early guidance can introduce an incorrect structure that persists through later denoising, while our learned schedule uses weaker early guidance and avoids this error. Appendix J further shows that strong constant guidance improves condition consistency but leads to less diverse generations, whereas our learned schedules retain clear condition-specific content while preserving greater variation and composition. Together, these results illustrate the benefit of adapting guidance across noise levels.

6 Conclusion

This work formulates the desired consistency–coverage trade-off for classifier-free guidance using a clean endpoint reference and optimizes the sampler-induced distribution toward this target. We derive exact trajectory-level identities for the consistency and KL terms under the sampler-induced distribution and characterize how guidance locally affects both terms across noise levels. These theoretical results motivate a time-dependent guidance schedule rather than a single global guidance strength. Experiments across four diffusion backbones on ImageNet-512 and COCO demonstrate that the learned schedules improve consistency–FID trade-offs over constant and non-uniform guidance baselines. Further analyses of the learned schedules and generation visualizations illustrate how guidance timing influences intermediate denoising dynamics and final sample quality.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, formatting checks, code verification, literature search, and reference verification. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector, 2024.

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.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo J. Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11472–11481, 2022.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2006.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Alexandre Galashov, Valentin De Bortoli, J. Swaroop Guntupalli, Guan-Horng Zhou, Kevin Murphy, Arthur Gretton, and Arnaud Doucet. Learn to guide your diffusion model. *arXiv preprint arXiv:2510.00815*, 2025.

Jiayang Gao, Tianyi Zheng, Jiayang Zou, Fengxiang Yang, Shice Liu, Luyao Fan, Zheyu Zhang, Hao Zhang, Jinwei Chen, Peng-Tao Jiang, Bo Li, and Jia Wang. C^2FG: Control classifier-free guidance via score discrepancy analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34398–34407, 2026.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

Florian Handke, Félix Koulischer, Gabriel Raya, and Luca Ambrogioni. Measuring semantic information production in generative diffusion models. In *ICLR 2025 Workshop on Deep Generative Models in Machine Learning: Theory, Principle and Efficacy*, 2025.

Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

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

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8633–8646, 2022b.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990.

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

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

Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. *Advances in Neural Information Processing Systems*, 37:52996–53021, 2024.

Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. Revelio: Interpreting and leveraging semantic information in diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Diederik P. Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21696–21707, 2021.

Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *International Conference on Learning Representations*, 2023.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

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. *arXiv preprint arXiv:2404.07724*, 2024.

Marvin Li and Sitan Chen. Critical windows: Non-asymptotic theory for feature emergence in diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 27474–27498. PMLR, 2024.

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

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

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494, 2022.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2256–2265, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, 2021a.

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.

Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

Jinjin Zhang, Qiuyu Huang, Junjie Liu, Xiefan Guo, and Di Huang. Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23464–23473, 2025.

A Limitations and Broader Impacts

Limitations. Our method is developed under the VE probability-flow formulation and relies on score evaluations from pretrained diffusion models. The sampler-induced law $q_t^{\mathbf{w}}$ is available only through generated trajectories, so the optimizer uses a fixed-trajectory descent direction and limits each proposed schedule change before checking it on the resulting trajectory. This remains an approximation to full trajectory-level optimization. The objective is also estimated with finite samples and Hutchinson trace estimators and can therefore depend on estimator variance, sampler discretization, and the chosen noise grid. Moreover, λ specifies an endpoint preference in the objective; its mapping to evaluation metrics such as FID, CLIP score, or classifier accuracy can be model dependent and nonlinear. We evaluate four diffusion backbones on ImageNet-512 and COCO, but broader evaluation across architectures, datasets, and modalities remains future work.

Broader impacts. This work studies how to improve conditional diffusion generation by learning where to apply classifier-free guidance along the reverse trajectory. Better guidance schedules may improve the balance between condition consistency, sample quality, and distributional coverage. At the same time, improvements to image and text-to-image generation can make synthetic content more realistic or more closely aligned with specified conditions, which may increase risks such as misleading generated media or other misuse. We do not release a new pretrained model or dataset, and our experiments are conducted on standard academic benchmarks using existing models.

B Additional Related Work on Diffusion Time

A growing body of work suggests that diffusion time and noise level are important axes for understanding and controlling generative diffusion models. Variational diffusion models relate the continuous-time objective to the signal-to-noise ratio and study noise-schedule optimization (Kingma et al., 2021). Prior work on timestep weighting also suggests that diffusion time is not uniform. Changing the weights assigned to different noise levels can substantially affect training and sample quality (Choi et al., 2022; Hang et al., 2023). Related design choices in EDM also emphasize the role of noise parameterization and sampling discretization (Karras et al., 2022). These works focus on training-time weighting, while our work studies a sampling-time question: how to allocate classifier-free guidance across the reverse trajectory.

Several recent works also study how semantic and structural information emerges along the reverse trajectory. Critical-window analyses show that certain features can emerge over narrow time intervals of the reverse process (Li & Chen, 2024). Statistical-physics analyses identify distinct dynamical regimes in diffusion sampling (Raya & Ambrogioni, 2023; Biroli et al., 2024). Other works measure when semantic information is produced during generation or analyze how interpretable concepts and semantic features evolve across denoising timesteps (Handke et al., 2025; Tinaz et al., 2025; Kim et al., 2025). These studies motivate treating guidance location as a meaningful design choice rather than using a single constant guidance strength at all steps.

Our objective is also related to information-theoretic views of diffusion. Classical de Bruijn-type identities connect Gaussian smoothing, entropy, and Fisher information (Cover & Thomas, 2006), and probability-flow likelihood formulas use related trajectory-level identities for score-based diffusion models (Song et al., 2021b;a). Information-Theoretic Diffusion further connects diffusion objectives to I-MMSE relations and information quantities along the noising process (Kong et al., 2023). In contrast to these works, which primarily analyze diffusion trajectories or training objectives, we use trajectory-level information quantities to define and optimize a sampling-time CFG schedule under the actual sampler-induced distribution.

C Additional Discussion of the Gaussian Toy Example

This appendix gives a qualitative discussion of the Gaussian toy example used in the main text. The purpose is not to derive a closed-form solution for the CFG-induced endpoint distribution, but to illustrate the terminal mismatch and the resulting consistency–coverage trade-off.

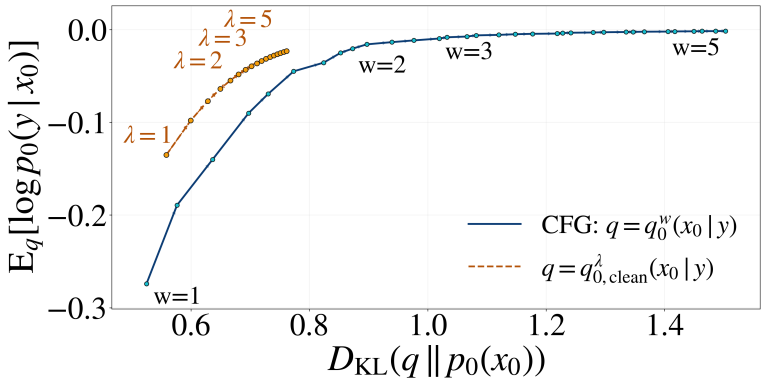


Figure 5: Gaussian trade-off between condition consistency and deviation from the original clean data distribution. The tilted family provides a natural endpoint reference: increasing λ improves condition consistency while controlling the deviation from the original clean distribution $p_0(x_0)$.

The Gaussian toy example also illustrates why $w = 1$ need not be the best empirical choice for the actual sampler. The choice $w = 1$ gives the standard conditional score field,

$$s_t^w(x_t, y) = s_t^{\mathrm{con}}(x_t, y). \tag{20}$$

Therefore, if the reverse process were initialized from the exact terminal conditional distribution $p_T(x_T \mid y)$, the conditional probability-flow ODE would transport this distribution back to the clean conditional distribution $p_0(x_0 \mid y)$.

However, practical sampling initializes the reverse process from the high-noise Gaussian approximation

$$X_T \sim \mathcal{N}(0, \sigma_T^2 I), \tag{21}$$

rather than from the exact class-conditional terminal distribution. In the Gaussian example, if

$$X_0 \mid Y = y \sim \mathcal{N}(\mu_y, \Sigma), \tag{22}$$

then the exact noised conditional distribution is

$$p_T(x_T \mid y) = \mathcal{N}(\mu_y, \Sigma + \sigma_T^2 I). \tag{23}$$

When σ_T is large, this distribution is close to the condition-independent Gaussian initialization in (21), but it is not identical to it. The mismatch includes both a mean mismatch, through μ_y, and a covariance mismatch, through $\Sigma + \sigma_T^2 I$ versus $\sigma_T^2 I$.

Thus, in this toy example, the reason $w = 1$ may fail to be empirically optimal is not that the conditional score field is incorrect. Rather, the full sampler is not exact because the reverse process starts from an approximate, condition-independent terminal distribution instead of the exact $p_T(x_T \mid y)$. Moderate guidance can empirically compensate for this terminal mismatch by pulling samples toward the corresponding data component. However, excessively large guidance over-guides the trajectory, increases the coverage-side KL, and may move samples into low-density regions of the clean data distribution.

Figure 5 connects the Gaussian example to the information quantities used in our objective. The horizontal axis is the coverage-side KL

$$\mathrm{KL}(q \,\|\, p_0(x_0)), \tag{24}$$

and the vertical axis is the consistency

$$\mathbb{E}_q[\log p_0(y \mid x_0)]. \tag{25}$$

The clean endpoint tilted family $q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)$ traces the desired endpoint trade-off: increasing λ improves consistency while increasing the associated KL divergence.

In contrast, the actual CFG-induced distributions $q_0^w(x_0 \mid y)$ move along a different curve as w changes. For the same consistency level, CFG can incur a larger coverage-side KL, and for large w it can move much farther from the original clean data distribution. This gap shows that increasing the sampler guidance weight w is not equivalent to increasing the endpoint trade-off parameter λ. Therefore, we use $q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)$ to define the desired consistency–coverage trade-off, and then tune the actual CFG schedule $\mathbf{w}$ to approach this target using the sampler-induced distribution.

D Details for the Mismatches

D.1 Clean Endpoint Tilt Versus Noisy Level Tilt

We first explain why the clean endpoint tilt and the noisy level tilt are generally different. The clean endpoint tilt is

$$q^w_{0,\mathrm{clean}}(x_0 \mid y) = \frac{1}{Z_0(y;w)} p_0(x_0) p_0(y \mid x_0)^w. \tag{26}$$

After applying the forward noising kernel $p_{t|0}(x_t \mid x_0)$, its time t marginal is

$$\begin{aligned} q^w_{t,\mathrm{clean}}(x_t \mid y) &= \int p_{t|0}(x_t \mid x_0) q^w_{0,\mathrm{clean}}(x_0 \mid y)\, dx_0 \\ &= \frac{1}{Z_0(y;w)} \int p_{t|0}(x_t \mid x_0) p_0(x_0) p_0(y \mid x_0)^w\, dx_0. \end{aligned} \tag{27}$$

Using Bayes' rule under the original forward process,

$$p_0(x_0 \mid x_t) = \frac{p_{t|0}(x_t \mid x_0) p_0(x_0)}{p_t(x_t)}, \tag{28}$$

we can rewrite (27) as

$$q^w_{t,\mathrm{clean}}(x_t \mid y) = \frac{p_t(x_t)}{Z_0(y;w)} \mathbb{E}_{p_0(x_0|x_t)} \left[p_0(y \mid x_0)^w \right]. \tag{29}$$

On the other hand, the noisy level tilt is

$$q^w_{t,\mathrm{tilt}}(x_t \mid y) = \frac{1}{Z_t(y;w)} p_t(x_t) p_t(y \mid x_t)^w. \tag{30}$$

Under the original joint distribution, $Y \to X_0 \to X_t$ forms a Markov chain, therefore

$$p_t(y \mid x_t) = \mathbb{E}_{p_0(x_0|x_t)} \left[p_0(y \mid x_0) \right], \tag{31}$$

we have

$$q^w_{t,\mathrm{tilt}}(x_t \mid y) = \frac{p_t(x_t)}{Z_t(y;w)} \left(\mathbb{E}_{p_0(x_0|x_t)} \left[p_0(y \mid x_0) \right] \right)^w. \tag{32}$$

Comparing (29) and (32), the two constructions differ because

$$\mathbb{E}_{p_0(x_0|x_t)} \left[p_0(y \mid x_0)^w \right] \neq \left(\mathbb{E}_{p_0(x_0|x_t)} \left[p_0(y \mid x_0) \right] \right)^w \qquad \text{in general.} \tag{33}$$

Thus,

$$q^w_{t,\mathrm{clean}}(x_t \mid y) \neq q^w_{t,\mathrm{tilt}}(x_t \mid y) \qquad \text{in general.} \tag{34}$$

Equality can hold in special cases, for example when $w = 0$, when $w = 1$ or when $p_0(y \mid X_0)$ is almost surely constant under $p_0(x_0 \mid x_t)$, but it does not hold in general.

D.2 Noisy Level Tilt Versus Actual CFG Distribution

The noisy level tilt

$$q^w_{t,\mathrm{tilt}}(x_t \mid y) \propto p_t(x_t) p_t(y \mid x_t)^w \tag{35}$$

is useful because its score is exactly the CFG field:

$$\nabla_{x_t} \log q^w_{t,\mathrm{tilt}}(x_t \mid y) = s^{\mathrm{un}}_t(x_t) + w \left(s^{\mathrm{con}}_t(x_t, y) - s^{\mathrm{un}}_t(x_t) \right) = s^w_t(x_t, y). \tag{36}$$

However, this identity is a statement about the score field at a single noise level. It does not imply that the family $\{q^w_{t,\mathrm{tilt}}(\cdot \mid y)\}_{t \in [0,T]}$ is the distribution path generated by the CFG probability-flow ODE.

The actual CFG distribution $q^w_t(x_t \mid y)$ is defined by evolving samples through the guided ODE, integrated backward from $t = T$ to $t = 0$:

$$\frac{dX_t}{dt} = -\sigma_t \dot{\sigma}_t s^w_t(X_t, y). \tag{37}$$

Therefore $q_t^w(x_t \mid y)$ is determined by the full transport dynamics and by the sampling distribution at the T end. A collection of densities defined separately at each noise level, such as $q_{t,\mathrm{tilt}}^w$, need not satisfy the continuity equation induced by (37). Consequently,

$$q_t^w(x_t \mid y) \neq q_{t,\mathrm{tilt}}^w(x_t \mid y) \qquad \text{in general.} \tag{38}$$

Equivalently, even though s_t^w is the score of the noisy level tilt, it is not generally the score of the actual CFG law:

$$s_t^w(x_t, y) \neq \nabla_{x_t} \log q_t^w(x_t \mid y). \tag{39}$$

This is why our information-theoretic quantities are evaluated under the actual CFG law $q_t^w(x_t \mid y)$, rather than under the noisy level tilt.

E Proof of Lemma 1

Assume the relevant densities are positive on the support of $q_0^{\mathbf{w}}(x_0 \mid y)$ and the expectations below are finite.

Proof. By the definition of the clean endpoint reference,

$$q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y) = \frac{1}{Z_\lambda(y)} p_0(x_0) p_0(y \mid x_0)^\lambda, \tag{40}$$

and therefore

$$\log q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y) = \log p_0(x_0) + \lambda \log p_0(y \mid x_0) - \log Z_\lambda(y). \tag{41}$$

Substituting this identity into the endpoint KL gives

$$\begin{aligned}
&\mathbb{E}_{p(y)} \mathrm{KL}\big(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)\big) \\
&= \mathbb{E}_{p(y)} \mathbb{E}_{q_0^{\mathbf{w}}(x_0|y)} \left[\log q_0^{\mathbf{w}}(x_0 \mid y) - \log q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)\right] \\
&= \mathbb{E}_{p(y)} \mathrm{KL}(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, p_0(x_0)) - \lambda \mathbb{E}_{p(y) q_0^{\mathbf{w}}(x_0|y)} \left[\log p_0(y \mid x_0)\right] + \mathbb{E}_{p(y)} \left[\log Z_\lambda(y)\right].
\end{aligned} \tag{42}$$

The last term is a scalar constant for fixed λ, because it depends on λ and the marginal $p(y)$, but not on the guidance schedule $\mathbf{w}$. Therefore, for fixed λ, minimizing the endpoint KL over $\mathbf{w}$ is equivalent to minimizing

$$\mathcal{L}_0(\mathbf{w}; \lambda) = -\lambda \mathbb{E}_{p(y) q_0^{\mathbf{w}}(x_0|y)} \left[\log p_0(y \mid x_0)\right] + \mathbb{E}_{p(y)} \mathrm{KL}(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, p_0(x_0)). \tag{43}$$

This proves the claim.

To derive the trade-off interpretation of λ, We next derive the slope interpretation in (51)–(52). For each condition y, define

$$Z_y(\lambda) := \int p_0(x_0) p_0(y \mid x_0)^\lambda \, dx_0, \qquad A_y(\lambda) := \log Z_y(\lambda). \tag{44}$$

Then

$$A_y'(\lambda) = \mathbb{E}_{q_{0,\mathrm{clean}}^{\lambda}(x_0|y)} \left[\log p_0(y \mid x_0)\right], \tag{45}$$

$$A_y''(\lambda) = \mathrm{Var}_{q_{0,\mathrm{clean}}^{\lambda}(x_0|y)} \left[\log p_0(y \mid x_0)\right] \geq 0. \tag{46}$$

Averaging (46) over $p(y)$ gives

$$\frac{d}{d\lambda} \mathbb{E}_{p(y) q_{0,\mathrm{clean}}^{\lambda}(x_0|y)} \left[\log p_0(y \mid x_0)\right] = \mathbb{E}_{p(y)} \mathrm{Var}_{q_{0,\mathrm{clean}}^{\lambda}(x_0|y)} \left[\log p_0(y \mid x_0)\right]. \tag{47}$$

Moreover,

$$\log \frac{q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y)}{p_0(x_0)} = \lambda \log p_0(y \mid x_0) - A_y(\lambda), \tag{48}$$

so

$$\mathrm{KL}\big(q_{0,\mathrm{clean}}^{\lambda}(x_0 \mid y) \,\|\, p_0(x_0)\big) = \lambda A_y'(\lambda) - A_y(\lambda). \tag{49}$$

Differentiating and using (46) yields

$$\frac{d}{d\lambda}\mathbb{E}_{p(y)}\mathrm{KL}\big(q^{\lambda}_{0,\mathrm{clean}}(x_0 \mid y)\,\|\,p_0(x_0)\big) = \lambda\,\mathbb{E}_{p(y)}\,\mathrm{Var}_{q^{\lambda}_{0,\mathrm{clean}}(x_0|y)}\left[\log p_0(y \mid x_0)\right]. \tag{50}$$

Combining the last two identities gives

$$\frac{d}{d\lambda}\mathbb{E}_{p(y)}\mathrm{KL}\big(q^{\lambda}_{0,\mathrm{clean}}(x_0 \mid y)\,\|\,p_0(x_0)\big) = \lambda\frac{d}{d\lambda}\mathbb{E}_{p(y)q^{\lambda}_{0,\mathrm{clean}}(x_0|y)}\left[\log p_0(y \mid x_0)\right]. \tag{51}$$

Whenever the consistency term changes with λ, dividing the two derivatives gives

$$\frac{d\,\mathbb{E}_{p(y)}\mathrm{KL}\Big(q^{\lambda}_{0,\mathrm{clean}}(x_0 \mid y)\,\|\,p_0(x_0)\Big)}{d\,\mathbb{E}_{p(y)q^{\lambda}_{0,\mathrm{clean}}(x_0|y)}\left[\log p_0(y \mid x_0)\right]} = \lambda. \tag{52}$$

Since the variance is nonnegative, both consistency and distributional deviation are **nondecreasing** in λ for $\lambda \geq 0$.

□

F Proofs for the Consistency Quantities

Standard information-theoretic diffusion along the original VE family. Information theory provides a natural lens for analyzing the generation process: mutual information measures dependence on the condition, while entropy measures the physical spread of the sample distribution. Under additive Gaussian smoothing, the relevant entropy and mutual-information identities follow from the heat-flow form of de Bruijn's identity and its mutual-information counterpart; see Cover & Thomas (2006).

Let

$$\tau(t) := \sigma_t^2, \qquad \dot{\tau}(t) := \frac{d}{dt}\sigma_t^2. \tag{53}$$

Then the conditional entropy $H(X_t \mid Y)$ and unconditional entropy $H(X_t)$ evolve according to

$$H(X_t \mid Y) = H(X_T \mid Y) + \frac{1}{2}\int_t^T \dot{\tau}(s)\,\mathbb{E}_{p_s(x_s,y)}[\nabla_x \cdot s_s^{\mathrm{con}}(x_s,y)]\,ds, \tag{54}$$

$$H(X_t) = H(X_T) + \frac{1}{2}\int_t^T \dot{\tau}(s)\,\mathbb{E}_{p_s(x_s)}[\nabla_x \cdot s_s^{\mathrm{un}}(x_s)]\,ds. \tag{55}$$

Consequently, the mutual information $I(X_t;Y) = H(X_t) - H(X_t \mid Y)$ admits the exact representation

$$\begin{aligned} I(X_t;Y) = I(X_T;Y) &- \frac{1}{2}\int_t^T \dot{\tau}(s)\,\mathbb{E}_{p_s(x_s,y)}[\nabla_x \cdot s_s^{\mathrm{con}}(x_s,y)]\,ds \\ &+ \frac{1}{2}\int_t^T \dot{\tau}(s)\,\mathbb{E}_{p_s(x_s,y)}[\nabla_x \cdot s_s^{\mathrm{un}}(x_s)]\,ds. \end{aligned} \tag{56}$$

Remark. For the original additive Gaussian smoothing family induced by p_0, integration by parts yields

$$\mathbb{E}_{p_t}[\nabla_x \cdot s_t^{\mathrm{un}}(X_t)] = -\mathbb{E}_{p_t}\left[\|s_t^{\mathrm{un}}(X_t)\|^2\right], \tag{57}$$

and analogously, for each fixed y,

$$\mathbb{E}_{p_t(\cdot|y)}[\nabla_x \cdot s_t^{\mathrm{con}}(X_t,y)] = -\mathbb{E}_{p_t(\cdot|y)}\left[\|s_t^{\mathrm{con}}(X_t,y)\|^2\right]. \tag{58}$$

In this appendix, we derive the consistency identity used in Lemma 2. Throughout, we use the VE/additive Gaussian smoothing parameterization

$$X_t = X_0 + \sigma_t Z, \qquad Z \sim \mathcal{N}(0, I_D), \qquad Z \perp (X_0, Y), \tag{59}$$

and write

$$\tau(t) := \sigma_t^2, \qquad \dot{\tau}(t) = \frac{d}{dt}\sigma_t^2 = 2\sigma_t\dot{\sigma}_t. \tag{60}$$

The original unconditional and conditional scores are

$$s_t^{\mathrm{un}}(x_t) := \nabla_{x_t} \log p_t(x_t), \qquad s_t^{\mathrm{con}}(x_t, y) := \nabla_{x_t} \log p_t(x_t \mid y). \tag{61}$$

We also define

$$s_t^{\mathrm{diff}}(x_t, y) := s_t^{\mathrm{con}}(x_t, y) - s_t^{\mathrm{un}}(x_t). \tag{62}$$

By Bayes' rule,

$$s_t^{\mathrm{diff}}(x_t, y) = \nabla_{x_t} \log p_t(y \mid x_t). \tag{63}$$

The CFG score field is

$$s_t^{\mathbf{w}}(x_t, y) = (1 - w_t) s_t^{\mathrm{un}}(x_t) + w_t s_t^{\mathrm{con}}(x_t, y) = s_t^{\mathrm{un}}(x_t) + w_t s_t^{\mathrm{diff}}(x_t, y), \tag{64}$$

and the induced probability-flow dynamics are

$$\frac{d}{dt} x_t = -\frac{1}{2} \dot{\tau}(t) s_t^{\mathbf{w}}(x_t, y) = -\sigma_t \dot{\sigma}_t s_t^{\mathbf{w}}(x_t, y). \tag{65}$$

F.1 VE HEAT-FLOW IDENTITIES

For each fixed y, the conditional density is obtained by Gaussian smoothing:

$$p_t(x_t \mid y) = \int p_0(x_0 \mid y) \mathcal{N}(x_t; x_0, \sigma_t^2 I_D)\, dx_0. \tag{66}$$

Similarly,

$$p_t(x_t) = \int p_0(x_0) \mathcal{N}(x_t; x_0, \sigma_t^2 I_D)\, dx_0. \tag{67}$$

Therefore both densities satisfy the heat equation in the variance parameter $\tau = \sigma_t^2$:

$$\partial_t p_t(x_t \mid y) = \frac{1}{2} \dot{\tau}(t) \Delta_x p_t(x_t \mid y), \tag{68}$$

$$\partial_t p_t(x_t) = \frac{1}{2} \dot{\tau}(t) \Delta_x p_t(x_t). \tag{69}$$

Using

$$\Delta p = \nabla_x \cdot (p \nabla_x \log p) = p \left(\nabla_x \cdot \nabla_x \log p + \|\nabla_x \log p\|^2 \right),$$

we obtain

$$\partial_t \log p_t(x_t \mid y) = \frac{1}{2} \dot{\tau}(t) \left[\nabla_x \cdot s_t^{\mathrm{con}}(x_t, y) + \|s_t^{\mathrm{con}}(x_t, y)\|^2 \right], \tag{70}$$

$$\partial_t \log p_t(x_t) = \frac{1}{2} \dot{\tau}(t) \left[\nabla_x \cdot s_t^{\mathrm{un}}(x_t) + \|s_t^{\mathrm{un}}(x_t)\|^2 \right]. \tag{71}$$

F.2 CONSISTENCY IDENTITY

Define

$$C(t) := \mathbb{E}_{p(y) q_t^{\mathbf{w}}(x_t \mid y)} \left[\log p_t(y \mid x_t) \right]. \tag{72}$$

Let

$$F_t(x_t, y) := \log p_t(y \mid x_t).$$

By Bayes' rule,

$$F_t(x_t, y) = \log p_t(x_t \mid y) - \log p_t(x_t) + \log p(y), \tag{73}$$

and therefore

$$\nabla_x F_t(x_t, y) = s_t^{\mathrm{con}}(x_t, y) - s_t^{\mathrm{un}}(x_t) = s_t^{\mathrm{diff}}(x_t, y). \tag{74}$$

Using (70) and (71), we also have

$$\begin{aligned} \partial_t F_t(x_t, y) &= \frac{1}{2} \dot{\tau}(t) \Big[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \|s_t^{\mathrm{con}}(x_t, y)\|^2 - \|s_t^{\mathrm{un}}(x_t)\|^2 \Big] \\ &= \frac{1}{2} \dot{\tau}(t) \Big[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \big\langle s_t^{\mathrm{diff}}(x_t, y), s_t^{\mathrm{con}}(x_t, y) + s_t^{\mathrm{un}}(x_t) \big\rangle \Big]. \end{aligned} \tag{75}$$

The distribution $q_t^{\mathbf{w}}(x_t \mid y)$ is transported by the velocity

$$b_t^{\mathbf{w}}(x_t, y) = -\frac{1}{2}\dot{\tau}(t) s_t^{\mathbf{w}}(x_t, y).$$

Equivalently, $q_t^{\mathbf{w}}$ satisfies the continuity equation

$$\partial_t q_t^{\mathbf{w}}(x_t \mid y) + \nabla_x \cdot (q_t^{\mathbf{w}}(x_t \mid y) b_t^{\mathbf{w}}(x_t, y)) = 0.$$

Therefore, applying the standard transport identity to $F_t(x_t, y) = \log p_t(y \mid x_t)$, we have

$$\frac{d}{dt}\mathbb{E}_{q_t^{\mathbf{w}}(x_t|y)}\left[F_t(x_t, y)\right] = \mathbb{E}_{q_t^{\mathbf{w}}(x_t|y)}\left[\partial_t F_t(x_t, y) + \langle \nabla_x F_t(x_t, y), b_t^{\mathbf{w}}(x_t, y)\rangle\right].$$

Hence the derivative of $C(t)$ along the transported distribution is

$$\begin{aligned}\frac{d}{dt}C(t) &= \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[\partial_t F_t(x_t, y) + \langle \nabla_x F_t(x_t, y), b_t^{\mathbf{w}}(x_t, y)\rangle\right] \\ &= \frac{1}{2}\dot{\tau}(t)\mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\Big[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \left\langle s_t^{\mathrm{diff}}(x_t, y), s_t^{\mathrm{con}}(x_t, y) + s_t^{\mathrm{un}}(x_t) - s_t^{\mathbf{w}}(x_t, y)\right\rangle\Big].\end{aligned} \tag{76}$$

Using

$$s_t^{\mathbf{w}}(x_t, y) = s_t^{\mathrm{un}}(x_t) + w_t s_t^{\mathrm{diff}}(x_t, y),$$

this becomes

$$\frac{d}{dt}C(t) = \frac{1}{2}\dot{\tau}(t)\mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\Big[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \left\langle s_t^{\mathrm{diff}}(x_t, y), s_t^{\mathrm{con}}(x_t, y)\right\rangle - w_t\|s_t^{\mathrm{diff}}(x_t, y)\|^2\Big]. \tag{77}$$

Since $\frac{1}{2}\dot{\tau}(t) = \sigma_t\dot{\sigma}_t$, integrating (77) from t to T gives

$$\begin{aligned}C(t) = C(T) - \int_t^T \sigma_s\dot{\sigma}_s\mathbb{E}_{p(y)q_s^{\mathbf{w}}(x_s|y)}\Big[&\nabla_x \cdot s_s^{\mathrm{diff}}(x_s, y) + \left\langle s_s^{\mathrm{diff}}(x_s, y), s_s^{\mathrm{con}}(x_s, y)\right\rangle \\ &- w_s\|s_s^{\mathrm{diff}}(x_s, y)\|^2\Big]ds,\end{aligned} \tag{78}$$

which proves Lemma 2.

G Proofs for the Coverage Quantity

We now prove Lemma 3. Let

$$D(t) := \mathbb{E}_{p(y)}\mathrm{KL}(q_t^{\mathbf{w}}(x_t \mid y) \,\|\, p_t(x_t)). \tag{79}$$

The guided distribution $q_t^{\mathbf{w}}(x_t \mid y)$ is transported by

$$b_t^{\mathbf{w}}(x_t, y) = -\frac{1}{2}\dot{\tau}(t) s_t^{\mathbf{w}}(x_t, y), \tag{80}$$

while the original unconditional marginal $p_t(x_t)$ is transported by

$$b_t^{\mathrm{un}}(x_t) = -\frac{1}{2}\dot{\tau}(t) s_t^{\mathrm{un}}(x_t). \tag{81}$$

The corresponding continuity equations are

$$\partial_t q_t^{\mathbf{w}} + \nabla_x \cdot (q_t^{\mathbf{w}} b_t^{\mathbf{w}}) = 0, \tag{82}$$

$$\partial_t p_t + \nabla_x \cdot (p_t b_t^{\mathrm{un}}) = 0. \tag{83}$$

Using the transport identity for

$$D(t) = \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[\log q_t^{\mathbf{w}}(x_t \mid y) - \log p_t(x_t)\right],$$

we get

$$\frac{d}{dt}D(t) = \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[-\nabla_x \cdot b_t^{\mathbf{w}} - \partial_t \log p_t - \langle b_t^{\mathbf{w}}, \nabla_x \log p_t\rangle\right]. \tag{84}$$

From the continuity equation for p_t,

$$\partial_t \log p_t(x_t) = -\nabla_x \cdot b_t^{\mathrm{un}}(x_t) - \langle b_t^{\mathrm{un}}(x_t), s_t^{\mathrm{un}}(x_t)\rangle \,. \tag{85}$$

Substituting this identity gives

$$\begin{aligned}\frac{d}{dt}D(t) &= \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[-\nabla_x \cdot b_t^{\mathbf{w}} + \nabla_x \cdot b_t^{\mathrm{un}} + \langle b_t^{\mathrm{un}} - b_t^{\mathbf{w}}, s_t^{\mathrm{un}}\rangle\right] \\ &= \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[-\nabla_x \cdot (b_t^{\mathbf{w}} - b_t^{\mathrm{un}}) - \langle b_t^{\mathbf{w}} - b_t^{\mathrm{un}}, s_t^{\mathrm{un}}\rangle\right]. \end{aligned} \tag{86}$$

Here w_t is a time-dependent scalar schedule, so it is independent of x_t. Since

$$b_t^{\mathbf{w}}(x_t, y) - b_t^{\mathrm{un}}(x_t) = -\frac{1}{2}\dot{\tau}(t) w_t s_t^{\mathrm{diff}}(x_t, y), \tag{87}$$

we obtain

$$\frac{d}{dt}D(t) = \frac{1}{2}\dot{\tau}(t)\mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)} w_t \left[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \left\langle s_t^{\mathrm{un}}(x_t), s_t^{\mathrm{diff}}(x_t, y)\right\rangle\right]. \tag{88}$$

Integrating from 0 to T and using $\frac{1}{2}\dot{\tau}(t) = \sigma_t\dot{\sigma}_t$ yields

$$D(0) = D(T) - \int_0^T \sigma_t\dot{\sigma}_t \, \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)} w_t \left[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \left\langle s_t^{\mathrm{un}}(x_t), s_t^{\mathrm{diff}}(x_t, y)\right\rangle\right] dt. \tag{89}$$

This is exactly Lemma 3.

Corollary 1 (Local consistency–coverage trade-off of CFG)**.** *Under the assumptions of Lemmas 2 and 3, holding the current sampler marginal $q_t^{\mathbf{w}}$ fixed, and whenever the denominator is nonzero,*

$$\frac{\dfrac{\partial}{\partial w_t}\left[-\dfrac{d}{dt}\mathbb{E}_{p(y)}\mathrm{KL}(q_t^{\mathbf{w}}(x_t \mid y) \,\|\, p_t(x_t))\right]}{\dfrac{\partial}{\partial w_t}\left[-\dfrac{d}{dt}\mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)} \log p_t(y \mid x_t)\right]} = -\frac{\mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[\nabla_x \cdot s_t^{\mathrm{diff}}(x_t, y) + \left\langle s_t^{\mathrm{un}}(x_t), s_t^{\mathrm{diff}}(x_t, y)\right\rangle\right]}{\mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[\left\|s_t^{\mathrm{diff}}(x_t, y)\right\|^2\right]}. \tag{90}$$

Proof of Corollary 1. From (77) and $s_t^{\mathrm{con}} = s_t^{\mathrm{un}} + s_t^{\mathrm{diff}}$, the reverse-time consistency rate satisfies

$$-\frac{d}{dt}C(t) = -\sigma_t\dot{\sigma}_t \, \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[\nabla_x \cdot s_t^{\mathrm{diff}} + \left\langle s_t^{\mathrm{un}}, s_t^{\mathrm{diff}}\right\rangle + (1 - w_t)\|s_t^{\mathrm{diff}}\|^2\right]. \tag{91}$$

Holding the current marginal $q_t^{\mathbf{w}}$ fixed therefore gives

$$\frac{\partial}{\partial w_t}\left[-\frac{d}{dt}C(t)\right] = \sigma_t\dot{\sigma}_t \, \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[\|s_t^{\mathrm{diff}}\|^2\right]. \tag{92}$$

Similarly, (88) gives

$$\frac{\partial}{\partial w_t}\left[-\frac{d}{dt}D(t)\right] = -\sigma_t\dot{\sigma}_t \, \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[\nabla_x \cdot s_t^{\mathrm{diff}} + \left\langle s_t^{\mathrm{un}}, s_t^{\mathrm{diff}}\right\rangle\right]. \tag{93}$$

Taking the ratio of (93) and (92) proves (90).

Under the same boundary conditions used for integration by parts,

$$\begin{aligned}&-\mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[\nabla_x \cdot s_t^{\mathrm{diff}} + \left\langle s_t^{\mathrm{un}}, s_t^{\mathrm{diff}}\right\rangle\right] \\ &\quad = \mathbb{E}_{p(y)q_t^{\mathbf{w}}}\left[\left\langle \nabla_x \log q_t^{\mathbf{w}} - s_t^{\mathrm{un}}, s_t^{\mathrm{diff}}\right\rangle\right], \end{aligned} \tag{94}$$

which gives the alignment interpretation stated in the main text.

Entropy–cross-entropy decomposition. The KL term also admits the standard decomposition

$$\begin{aligned}&\mathbb{E}_{p(y)}\mathrm{KL}(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, p_0(x_0)) \\ &\quad = -H_{q_0^{\mathbf{w}}}(X_0 \mid Y) - \mathbb{E}_{p(y)q_0^{\mathbf{w}}(x_0|y)}\left[\log p_0(x_0)\right]. \end{aligned} \tag{95}$$

The first term is the negative conditional entropy of the generated distribution, while the second is its cross-entropy with the original clean data distribution. This identity is provided only as an interpretation of the KL term; the proof of Lemma 3 above does not rely on it.

H Loss Implementation

H.1 Combining the consistency and coverage identities

Recall that the endpoint objective is

$$\mathcal{L}_0(\mathbf{w};\lambda) = -\lambda C(0) + D(0), \tag{96}$$

where

$$C(0) := \mathbb{E}_{p(y)q_0^{\mathbf{w}}(x_0|y)}\left[\log p_0(y \mid x_0)\right], \tag{97}$$

and

$$D(0) := \mathbb{E}_{p(y)}\mathrm{KL}(q_0^{\mathbf{w}}(x_0 \mid y) \,\|\, p_0(x_0)). \tag{98}$$

From Lemma 2, we have

$$C(0) = C(T) - \int_0^T \sigma_t\dot{\sigma}_t\, \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[A_t(x_t,y) - w_t R_t(x_t,y)\right] dt, \tag{99}$$

where

$$A_t(x_t,y) := \nabla_x \cdot s_t^{\mathrm{diff}}(x_t,y) + \left\langle s_t^{\mathrm{diff}}(x_t,y), s_t^{\mathrm{con}}(x_t,y)\right\rangle, \tag{100}$$

$$R_t(x_t,y) := \left\| s_t^{\mathrm{diff}}(x_t,y)\right\|^2. \tag{101}$$

From Lemma 3, we also have

$$D(0) = D(T) - \int_0^T \sigma_t\dot{\sigma}_t\, \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[w_t B_t(x_t,y)\right] dt, \tag{102}$$

where

$$B_t(x_t,y) := \nabla_x \cdot s_t^{\mathrm{diff}}(x_t,y) + \left\langle s_t^{\mathrm{un}}(x_t), s_t^{\mathrm{diff}}(x_t,y)\right\rangle. \tag{103}$$

Since

$$s_t^{\mathrm{con}}(x_t,y) = s_t^{\mathrm{un}}(x_t) + s_t^{\mathrm{diff}}(x_t,y), \tag{104}$$

we have

$$A_t(x_t,y) = B_t(x_t,y) + R_t(x_t,y), \qquad \text{or equivalently} \qquad B_t(x_t,y) = A_t(x_t,y) - R_t(x_t,y). \tag{105}$$

Substituting (99) and (102) into $\mathcal{L}_0(\mathbf{w};\lambda) = -\lambda C(0) + D(0)$, and dropping the terminal terms $-\lambda C(T) + D(T)$ that are shared by schedules using the same terminal sampling prior, gives the schedule-dependent trajectory objective

$$\begin{aligned}
\mathcal{L}_{0,\mathrm{traj}}(\mathbf{w};\lambda) &= \int_0^T \sigma_t\dot{\sigma}_t\, \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[\lambda\left(A_t(x_t,y) - w_t R_t(x_t,y)\right) - w_t B_t(x_t,y)\right] dt \\
&= \int_0^T \sigma_t\dot{\sigma}_t\, \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[\lambda A_t - \lambda w_t R_t - w_t(A_t - R_t)\right] dt \\
&= \int_0^T \sigma_t\dot{\sigma}_t\, \mathbb{E}_{p(y)q_t^{\mathbf{w}}(x_t|y)}\left[(\lambda - w_t)A_t + w_t(1-\lambda)R_t\right] dt.
\end{aligned} \tag{106}$$

This is the continuous trajectory loss that is discretized in the implementation.

H.2 Fixed-trajectory proposal with adaptive update size

We now describe the adaptive step-limit update used to optimize our trajectory objective. The objective itself is always (109); the adaptive rule only determines how a fixed-trajectory proposal is accepted and how its update radius is adjusted after rerunning the sampler. We use this trust-region-style update for all reported SD 1.5, SDXL, and DiT-XL/2 experiments. EDM2-S optimizes the same trajectory objective without the adaptive-radius mechanism, as described below.

For a sampler with reverse-time grid

$$T = t_K > \cdots > t_1 \approx 0,$$

we write the stepwise schedule as

$$\mathbf{w} = (w_1, \dots, w_K), \qquad w_k = w(t_k).$$

At optimization iteration m, trajectories

$$\{x_{t_k}^{(b,m)}\}_{k=1,b=1}^{K,B}$$

are generated under the current schedule $\mathbf{w}^{(m)}$. Holding these trajectories fixed, define

$$A_k^{(m)} := \frac{1}{B}\sum_{b=1}^{B}\left[\nabla_x \cdot s_{t_k}^{\mathrm{diff}}(x_{t_k}^{(b,m)}, y^{(b)}) + \left\langle s_{t_k}^{\mathrm{diff}}(x_{t_k}^{(b,m)}, y^{(b)}), s_{t_k}^{\mathrm{con}}(x_{t_k}^{(b,m)}, y^{(b)})\right\rangle\right], \tag{107}$$

$$R_k^{(m)} := \frac{1}{B}\sum_{b=1}^{B}\left\| s_{t_k}^{\mathrm{diff}}(x_{t_k}^{(b,m)}, y^{(b)})\right\|^2. \tag{108}$$

The implementation applies the same sample averaging and dimension normalization used by the corresponding loss estimator to these quantities. The finite-grid objective is

$$\widehat{\mathcal{L}}_0(\mathbf{w};\lambda) = \sum_{k=1}^{K} a_k\left[(\lambda - w_k)A_k + w_k(1-\lambda)R_k\right], \tag{109}$$

where a_k is the integration weight defined in Section 4.

The fixed-trajectory derivative with respect to w_k is therefore

$$G_k^{(m)} = a_k\left[-A_k^{(m)} + (1-\lambda)R_k^{(m)}\right]. \tag{110}$$

Thus, the integration weight is already included in $G_k^{(m)}$; it is not multiplied again when forming the schedule update. This derivative captures only the explicit dependence of the objective on w_k while the sampled trajectory is held fixed. It does not differentiate through the dependence of the sampler-induced trajectory on the schedule.

To form the next candidate, we choose an update $\boldsymbol{\delta} = (\delta_1, \dots, \delta_K)$ that minimizes the fixed-trajectory linear change

$$\sum_{k=1}^{K} G_k^{(m)}\delta_k \tag{111}$$

subject to

$$\|\boldsymbol{\delta}\|_2 \le R_{\mathrm{TR}}^{(m)}, \qquad w_{\min} \le w_k^{(m)} + \delta_k \le w_{\max}, \qquad \delta_k = 0 \text{ for frozen steps.} \tag{112}$$

Here $R_{\mathrm{TR}}^{(m)}$ limits the total change of the schedule in one iteration. In the implementation, the constrained update can be written as

$$\delta_k = \mathrm{clip}\left(-\frac{G_k^{(m)}}{\mu},\, w_{\min} - w_k^{(m)},\, w_{\max} - w_k^{(m)}\right) \tag{113}$$

for updatable steps, with $\delta_k = 0$ on frozen steps. The scalar $\mu > 0$ is found by bisection so that the update satisfies the ℓ_2 constraint.

Let

$$\Delta_{\mathrm{pred}} = -\sum_{k=1}^{K} G_k^{(m)}\delta_k \tag{114}$$

be the decrease predicted from the current trajectories. We then rerun the sampler under the candidate schedule $\widetilde{\mathbf{w}} = \mathbf{w}^{(m)} + \boldsymbol{\delta}$ and compute

$$\Delta_{\mathrm{actual}} = \widehat{\mathcal{L}}_0(\mathbf{w}^{(m)};\lambda) - \widehat{\mathcal{L}}_0(\widetilde{\mathbf{w}};\lambda). \tag{115}$$

The current and candidate evaluations use the same conditions, initial noise, and Hutchinson random vectors, while all trajectory-dependent quantities are recomputed under $\widetilde{\mathbf{w}}$. When

$$\Delta_{\mathrm{pred}} \le 10^{-12}\max\left(1, \left|\widehat{\mathcal{L}}_0(\mathbf{w}^{(m)};\lambda)\right|\right), \tag{116}$$

we do not form a ratio; the current schedule is retained and the allowed update size is reduced. Otherwise, we compute

$$r = \frac{\Delta_{\text{actual}}}{\Delta_{\text{pred}}}. \tag{117}$$

The candidate schedule is used for the next iteration when $\Delta_{\text{actual}} > 0$ and $r \geq 0.1$; otherwise the current schedule is retained. Independently, the allowed update size for the next iteration is

$$R_{\text{TR}}^{(m+1)} = \text{clip}_{[R_{\min}, R_{\max}]} \begin{cases} 0.5R_{\text{TR}}^{(m)}, & r < 0.25, \\ 1.5R_{\text{TR}}^{(m)}, & r \geq 0.75 \text{ and } \|\boldsymbol{\delta}\|_2 \geq 0.9R_{\text{TR}}^{(m)}, \\ R_{\text{TR}}^{(m)}, & \text{otherwise}, \end{cases} \tag{118}$$

with

$$R_{\min} = R_0/128, \qquad R_{\max} = 8R_0. \tag{119}$$

The initial radius R_0 is model dependent. In particular, $0.1 \leq r < 0.25$ can still update the schedule while reducing the radius for the next iteration. Each optimization iteration evaluates only one candidate, so the number of iterations is not the number of schedule changes actually taken.

Algorithm 1 Guidance Schedule Update with Adaptive Step Limit

Require: λ, initial schedule $\mathbf{w}^{(0)}$, bounds $[w_{\min}, w_{\max}]$, initial radius R_0, and frozen sampler steps

Set $R_{\text{TR}}^{(0)} \leftarrow R_0$, $R_{\min} \leftarrow R_0/128$, and $R_{\max} \leftarrow 8R_0$
for $m = 0, \ldots, M-1$ **do**
  Generate current trajectories and estimate $\widehat{\mathcal{L}}_0(\mathbf{w}^{(m)}; \lambda)$
  Compute $G_k^{(m)}$ from (110)
  Solve (111)–(112) for $\boldsymbol{\delta}$
  Set $\widetilde{\mathbf{w}} \leftarrow \mathbf{w}^{(m)} + \boldsymbol{\delta}$
  Rerun the sampler with $\widetilde{\mathbf{w}}$ using the same conditions, initial noise, and Hutchinson random vectors
  Compute Δ_{pred} and Δ_{actual}
  if Δ_{pred} satisfies (116) **then**
    $\mathbf{w}^{(m+1)} \leftarrow \mathbf{w}^{(m)}$
    $R_{\text{TR}}^{(m+1)} \leftarrow \max(R_{\min}, 0.5R_{\text{TR}}^{(m)})$
  else
    Compute $r \leftarrow \Delta_{\text{actual}}/\Delta_{\text{pred}}$
    if $\Delta_{\text{actual}} > 0$ and $r \geq 0.1$ **then**
      $\mathbf{w}^{(m+1)} \leftarrow \widetilde{\mathbf{w}}$
    else
      $\mathbf{w}^{(m+1)} \leftarrow \mathbf{w}^{(m)}$
    end if
    Update $R_{\text{TR}}^{(m+1)}$ using (118)
  end if
end for
return $\mathbf{w}^{(M)}$

I Experiment Details

This appendix provides the model-specific generation settings, baseline configurations, optimization settings, evaluation protocols, and supplementary results for the experiments in Section 5. Unless stated otherwise, all methods compared within the same backbone use the same conditioning inputs, initial noise, sampler, and evaluation pipeline.

I.1 Experimental Configurations

Baseline settings and comparison protocol. Unless otherwise noted, we use the settings from the original papers, adjusting them for the model or sampler when needed. The Interval Guidance

paper does not report settings for SD 1.5, so we use a middle-stage guidance window based on prior stage-wise analyses of CFG at different noise levels (Jin et al., 2025). For each baseline, we keep the schedule shape and other hyperparameters fixed and vary only the overall guidance strength to obtain a trade-off curve. Within each backbone, all methods use the same sampler, conditioning inputs, number of samples, and evaluation procedure. We also reuse the same initial noise where applicable.

For the initial schedule $\mathbf{w}^{(0)}$ required by Algorithm 1, we use the constant schedule $w = 1$ as the default initialization.

For our learned schedules, we exclude selected steps near the two ends of the noise grid from optimization and keep them at $w = 1$. Under a finite sampler discretization, guidance at the highest-noise end is weakly responsive, whereas the lowest-noise end can be overly sensitive to discretization and score estimation error. Optimizing these endpoint steps can therefore add computation without a reliable benefit and can amplify numerical error. The exact excluded indices depend on the model and sampler discretization.

Stable Diffusion 1.5. We evaluate SD 1.5 on 5,000 COCO val2017 image–caption pairs, using the first caption associated with each image in a fixed ordering. Generation uses 40 deterministic Euler steps with a Karras noise schedule ($\rho = 3$) at 512×512 resolution. Constant CFG uses $w \in \{1, 3, 5, 7, 9\}$. Interval Guidance sets $w = 1$ outside $0.2 \le t/999 \le 0.8$ (generation steps 14–29) and sweeps the active guidance strength over $\{5, 7, 9, 11, 16, 21\}$. C^{2}FG uses rate 0.2 and sweeps $w_0 \in \{3, 5, 7, 9\}$. For β-CFG, we fix $a = b = 2$ and $\gamma = 0$, so the schedule shape is $f(u) = 6u(1 - u)$, and sweep its amplitude over $\{3, 5, 7, 9\}$.

For our method, optimization uses 64 fixed prompt contexts; each iteration uses 8 contexts and 4 Rademacher Hutchinson probes. All reported SD 1.5 schedules use the adaptive step-limit update in Appendix H, with $\lambda \in \{3, 5, 7, 9, \ldots\}$, where the ellipsis denotes additional settings used for the trade-off curve, and 15 optimization iterations. On a single NVIDIA B200 GPU, these 15 iterations take approximately 6.2 minutes. The optimized weights are constrained to $[1, 20]$.

SDXL. We evaluate SDXL on the same 5,000 COCO val2017 image–caption pairs. Generation uses 40 deterministic Euler/EDM steps with a Karras noise schedule ($\rho = 3$) at 1024×1024 resolution. Constant CFG uses $w \in \{1, 3, 5, 7, 9\}$. The Interval baseline applies guidance over $\sigma \in (0.28, 5.42]$ (generation steps 13–32) and sweeps the active guidance strength over $\{5, 9, 13, 17\}$; this interval follows the published example setting rather than a COCO-specific re-optimization. C^{2}FG uses rate 0.2 with target mean guidance in $\{3, 5, 7\}$. For β-CFG, we again fix $a = b = 2$ and $\gamma = 0$ and use target mean guidance in $\{3, 5, 7, 9\}$.

For our method, optimization uses 64 fixed prompt contexts, 8 contexts per iteration, and 4 Rademacher Hutchinson probes. All reported SDXL schedules use the adaptive step-limit update in Appendix H, with $\lambda \in \{2, 3, 5, 6, 7, 8, 9\}$ and 15 optimization iterations. On a single NVIDIA B200 GPU, these 15 iterations take approximately 62.4–66.4 minutes. The optimized weights are constrained to $[1, 20]$.

DiT-XL/2. We use the class-conditional DiT-XL/2 512×512 checkpoint and evaluate 10,000 generated images per operating point. Generation uses 100 deterministic DDIM steps with $\eta = 0$ and `clip_denoised=False`. Target labels are sampled with a fixed random seed and are reused across methods together with the initial noise; they are not explicitly class-balanced. Constant CFG uses $w \in \{1, 1.25, 1.5, 1.75, 2, 3\}$. Interval Guidance sets $w = 1$ outside the mapped model-time interval $201 < t \le 502$, corresponding to zero-based DDIM steps 50–79, and sweeps the active guidance strength over $\{1.5, 2.5, 3.5, 4.5, 5.5\}$. This interval is mapped from the 250-step configuration in the original method rather than copied unchanged. C^{2}FG uses rate $\ln 2$ and sweeps $w_0 \in \{0.8, 1, 1.1, 1.2, 1.4, 2, 3\}$.

Our DiT schedules use the TR update in Appendix H with $\lambda \in \{1.2, 1.5, 1.6, 1.8, 1.9, 2, 3, 4\}$. Each optimization iteration uses 8 fresh trajectories and 8 Gaussian Hutchinson probes. We use 10 iterations for $\lambda \in \{1.5, 3\}$ and 7 iterations for the remaining values. A 7-iteration run takes approximately 132.3–132.4 minutes on a single NVIDIA B200 GPU. The schedule is represented on the original 250-step model-time grid and interpolated by model time to the 100-step DDIM generation grid. Optimized weights are constrained to $[1, 5]$.

Table 2: EDM2-S validation with 50,000 generated samples for FID, classification accuracy, and IS. Precision, recall, and F-score use the fixed 10K-generated versus 10K-real protocol.

Method	Setting	FID ↓	Acc. (%) ↑	IS ↑	Prec. ↑	Rec. ↑	F-score ↑
Constant CFG	$w = 1.4$	2.29	88.470	268.600	0.782	0.717	0.748
C^2FG	$w_0 = 1.1$	2.10	88.140	267.720	0.780	0.722	0.750
Interval	$w = 2.1$	**1.68**	89.400	282.970	0.795	**0.747**	0.770
Ours	$\lambda = 1.45$	1.70	**89.940**	**289.880**	**0.797**	0.743	0.769

EDM2-S. We use the conditional EDM2-S 512×512 checkpoint together with the EDM2-XS unconditional checkpoint and apply ordinary CFG (not autoguidance). Each operating point is evaluated with 10,000 generated images. Generation uses the deterministic 32-step Heun sampler with $S_{\text{churn}} = 0$, $\sigma_{\min} = 0.002$, $\sigma_{\max} = 80$, and $\rho = 7$; the same guidance value is used in the predictor and corrector evaluations. Target labels and initial noise are fixed within this backbone and reused across methods. Constant CFG uses $w \in \{1, 1.2, 1.4, 1.6, 1.8, 2, 3, 4\}$. Interval Guidance sets $w = 1$ outside steps 15–21, approximately $\sigma \in (0.28, 2.90]$, and sweeps the active guidance strength over $\{1.5, 2.1, 3.5, 4, 5, 6\}$. C^2FG uses rate 0.5 and sweeps $w_0 \in \{0.9, 1, 1.1, 1.4, 1.7, 2, 3, 4\}$; this rate is adapted from the published EDM2 autoguidance setting to ordinary CFG and is therefore not an exact reproduction of that configuration.

Our EDM2-S runs use $\lambda \in \{1.2, 1.45, 1.8, 2, 2.5, 3\}$. Optimization uses a fixed pool of 512 candidate trajectories, 8 trajectories per iteration, 8 Gaussian Hutchinson probes, and 60 iterations with update step size $\eta = 0.012$. On a single NVIDIA B200 GPU, these 60 iterations take approximately 13.5–15.6 minutes. These runs optimize the same trajectory objective but use the fixed-trajectory update without the adaptive-radius mechanism. EDM2-S has only 32 sampling steps, and we observed consistently high proposal acceptance in this setting, so the additional radius adaptation was unnecessary in practice.

The wall-clock cost depends on both model size and the number of sampling steps. For example, generating 10,000 samples on a single NVIDIA B200 GPU takes approximately 12 hours for DiT-XL/2 but about 1 hour for EDM2-S.

I.2 50K-Sample ImageNet Validation

The main paper uses 10,000 generated samples per operating point for the dense ImageNet trade-off sweeps. This keeps the comparison computationally tractable when many operating points are required for every method. The cost is particularly high for DiT-XL/2, which uses 100-step DDIM sampling: generating 10,000 samples for a single setting takes approximately 12 hours on one NVIDIA B200 GPU. We therefore use 10K samples for the main trade-off curves and perform an additional 50K evaluation on selected EDM2-S settings to check whether increasing the evaluation sample size materially changes the relative comparison.

Table 2 reports the EDM2-S 50K evaluation at the same operating points used in the main comparison. FID, top-1 classification accuracy, and IS are computed from 50,000 generated samples. Precision, recall, and F-score continue to use the fixed 10,000-generated versus 10,000-real protocol. The completed 50K results preserve the main comparison: Interval Guidance has slightly lower FID than Ours, while Ours has higher classification accuracy and IS. Both substantially improve over constant CFG and C^2FG in FID at these operating points. This supports using 10K samples for the dense trade-off sweeps in the main paper while reserving 50K evaluation for selected robustness checks.

For the EDM2-S adaptive schedule in Table 2, optimization uses the same setting as the corresponding main-paper operating point: a fixed pool of 512 trajectories, 60 optimization iterations, update step size $\eta = 0.012$, no trust-region update, and the same endpoint-step exclusion described above.

I.3 Evaluation Protocols and Metrics

Evaluation protocol. We evaluate all competing guidance schedules under a matched protocol within each diffusion backbone. In particular, methods compared on the same backbone use the

same sampler, conditioning inputs, number of sampling steps, and evaluation pipeline, with the same target conditions and initial noise reused across methods where applicable. Baseline methods sweep their corresponding guidance-strength parameter while keeping the schedule form fixed, whereas our operating points are obtained from schedules optimized with different values of λ. Since these parameters have different meanings across methods, we compare the resulting evaluation metric trade-offs rather than matching their numerical parameter values.

For text-to-image generation, we evaluate SD 1.5 and SDXL on COCO using FID and CLIP similarity. For class-conditional generation, we evaluate DiT-XL/2 and EDM2-S on ImageNet-512 using FID, top-1 classification accuracy, Inception Score (IS), precision, and recall. These evaluation metrics are used only for evaluation and are not included in the schedule-optimization objective.

The evaluation implementation is kept fixed across methods within each backbone. However, DiT-XL/2 and EDM2-S follow their respective Inception evaluation pipelines rather than a single shared implementation. We therefore use the ImageNet metrics primarily to compare guidance methods within a fixed backbone and do not use absolute cross-backbone differences in IS as evidence for one method over another.

Fréchet Inception Distance (FID). FID measures the discrepancy between the generated and reference image distributions in Inception feature space (Heusel et al., 2017). Lower FID indicates closer agreement with the reference distribution and reflects both sample fidelity and distributional coverage. For the ImageNet experiments, DiT-XL/2 uses the ADM TensorFlow Inception evaluation pipeline, whereas EDM2-S uses the NVIDIA PyTorch Inception detector `inception-2015-12-05.pkl`. For the COCO experiments, both SD 1.5 and SDXL use the NVIDIA/StyleGAN Inception-v3 detector `inception-2015-12-05.pkl` and its 2048-dimensional features. Generated and reference images are resized while preserving aspect ratio and center-cropped to 512×512 before feature extraction; FID is then computed from the empirical feature means and covariances using the standard matrix-square-root formulation. All methods within a given backbone are evaluated using the same FID implementation.

Classification accuracy. For both DiT-XL/2 and EDM2-S, we measure condition consistency using the same torchvision ResNet-50 classifier with `ResNet50_Weights.IMAGENET1K_V2` (`resnet50-11ad3fa6.pth`). Images are resized so that the shorter side is 232 pixels, center-cropped to 224×224, and normalized using the ImageNet preprocessing associated with these weights. We compare the top-1 predicted class with the target class used for generation and report the mean top-1 accuracy in percent. Higher accuracy therefore indicates stronger agreement between the generated sample and its conditioning class.

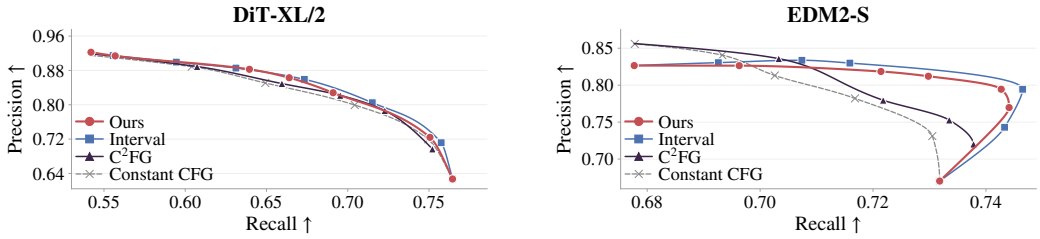
CLIP similarity. For both SD 1.5 and SDXL, we compute text–image alignment using OpenCLIP with the ViT-H/14 model (`ViT-H-14`) pretrained with the `laion2b_s32b_b79k` weights. Image and text embeddings are independently ℓ_2-normalized, and we average their cosine similarity over generated image–caption pairs. The reported CLIP value is 100 times this mean cosine similarity; no learned logit scale is applied. Higher values therefore indicate stronger alignment with the conditioning caption.

Inception Score (IS). For DiT-XL/2, IS is computed using the ADM evaluator `Evaluator.compute_inception_score` with the TensorFlow `classify_image_graph_def.pb` Inception-v3 model. The computation reuses the Inception outputs produced during the corresponding FID evaluation, so no additional image generation is required.

For EDM2-S, we use NVIDIA's PyTorch `inception-2015-12-05.pkl` detector, which is also used by our EDM2 FID evaluation. We convert the detector outputs to softmax probabilities without adding the output-layer bias and compute the standard Inception Score,

$$\mathrm{IS} = \exp(\mathbb{E}_x \left[D_{\mathrm{KL}}(p(y \mid x) \,\|\, p(y)) \right]) . \tag{120}$$

Thus, both evaluations use Inception-v3 classifiers without an output-layer bias, but they rely on different model files and software implementations (TensorFlow for DiT-XL/2 and PyTorch for EDM2-S). We have not assumed numerical equivalence between the two implementations and therefore interpret IS comparisons within, rather than across, backbones.



(a) DiT-XL/2. (b) EDM2-S.

Figure 6: Precision–recall trade-offs on ImageNet-512. Each curve is obtained by varying the corresponding method-specific guidance parameter.

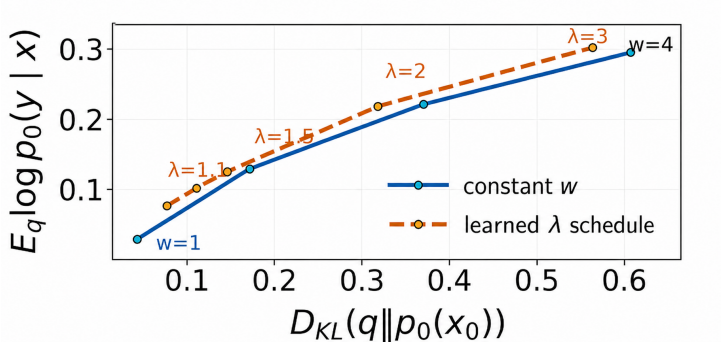


Figure 7: Objective trade-off between consistency and the KL term for learned schedules and constant guidance on the EDM experiment.

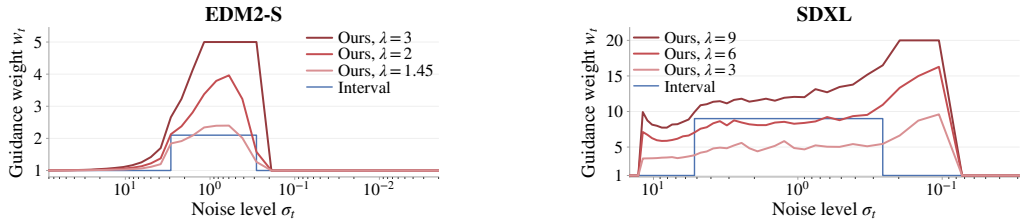
Precision and recall. For both DiT-XL/2 and EDM2-S, we compute improved precision and recall (Kynkäänniemi et al., 2019) in the same DINOv2 feature space. We use DINOv2 ViT-L/14 (`dinov2_vitl14`, `dinov2_vitl14_pretrain.pth`) to extract 1024-dimensional features from images resized to 224×224 and normalized with ImageNet statistics. Precision and recall are then computed with our project-local torch-fidelity implementation of `prc_features_to_metric`, using Euclidean distances and $k = 3$ nearest-neighbor manifolds. Precision measures the fraction of generated samples supported by the real-data manifold and primarily reflects sample fidelity, whereas recall measures the fraction of the real-data manifold covered by generated samples and primarily reflects distributional coverage.

Metric roles. In our analysis, classification accuracy and CLIP similarity measure *condition consistency*; recall measures *distributional coverage*; precision measures *sample fidelity*; and FID summarizes overall distributional agreement between generated and reference images. IS provides an additional class-conditional generation metric.

I.4 Additional Quantitative Results

Precision–recall trade-offs. Figure 6 reports the full precision–recall trade-offs for DiT-XL/2 and EDM2-S. These curves provide a complementary view of the minimum-FID operating points reported in Table 1; the methods are less clearly separated in precision–recall space than in the classification accuracy–FID comparisons in the main text.

Objective trade-off. Figure 7 evaluates the learned schedules using the consistency and KL quantities that define our trajectory objective. The learned schedules improve the measured objective trade-off over constant guidance, showing that the schedule optimization changes the criterion it is designed to optimize. We use the CLIP–FID and classification accuracy–FID curves in the main text as the primary evaluation because those metrics are not used during schedule optimization.



(a) EDM2-S on ImageNet-512.

(b) SDXL on COCO.

Figure 8: Additional learned guidance schedules across noise levels. Left: EDM2-S with $\lambda \in \{1.45, 2, 3\}$. Right: SDXL with $\lambda \in \{3, 6, 9\}$. Interval Guidance is shown for comparison. The horizontal axis follows the reverse trajectory from high to low noise.



(a) ImageNet class 281, tabby cat.

(b) Prompt: A woman sitting on a bench and a woman standing waiting for the bus.

Figure 9: Stage-wise ablation of guidance location on ImageNet and COCO. Guidance is activated only in one noise range and set to $w_t = 1$ elsewhere. Columns compare $w = 1$, high-noise guidance, middle-noise guidance, low-noise guidance, and the learned adaptive schedule.

I.5 Additional Schedule Analysis

Figure 8 complements Figure 4 with the learned schedules for the other two backbones.

I.6 Guidance Timing and Trajectory Analysis

Stage-wise guidance ablation. To isolate the effect of guidance location, we activate guidance only within one noise range (high, middle, or low noise) and set $w_t = 1$ elsewhere. Larger σ-indices correspond to higher noise levels. For EDM with 32 sampling steps, we use interval guidance $w = 3$ on the high-noise interval σ_{32}–σ_{20}, the middle-noise interval σ_{16}–σ_{10}, or the low-noise interval σ_9–σ_4. For SDXL with 40 sampling steps, we use interval guidance $w = 9$ on the high-noise interval σ_{40}–σ_{28}, the middle-noise interval σ_{27}–σ_8, or the low-noise interval σ_7–σ_1. Figure 9 shows that changing the location of guidance can alter both global structure and local image details, providing additional qualitative evidence that guidance timing matters.

Denoising trajectory visualization. To illustrate how guidance timing affects later denoising, we visualize the clean prediction at several noise levels in Figure 10. Under constant $w = 9$, an ambiguous dark region is pushed toward a fifth cup at an early stage and persists through subsequent denoising, yielding five cups in the final image. Our learned schedule with $\lambda = 9$ applies weaker early guidance; the same region remains less structured and is later suppressed, producing four cups. This example illustrates how early guidance can affect later denoising, motivating joint optimization of guidance across the trajectory.

J Additional Generation

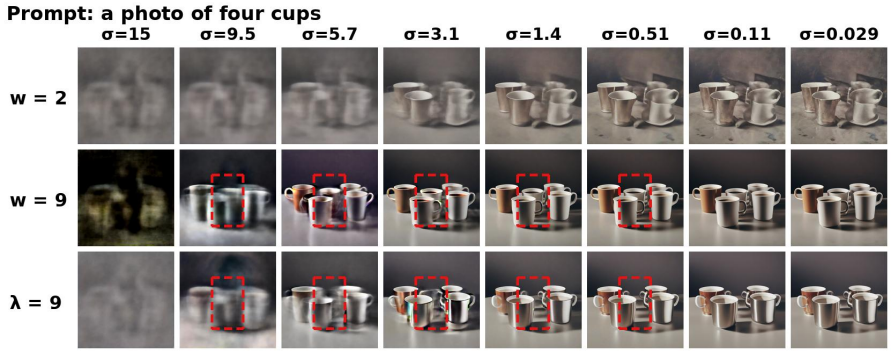


Figure 10: Denoising trajectory visualization for the prompt "a photo of four cups." At each noise level σ_t, we visualize the clean prediction $\hat{x}_0^{\mathbf{w}}(x_t, y) = x_t + \sigma_t^2 s_t^{\mathbf{w}}(x_t, y)$, using the VE score-to-denoiser transformation. From left to right, columns show the clean predictions as the noise level decreases.

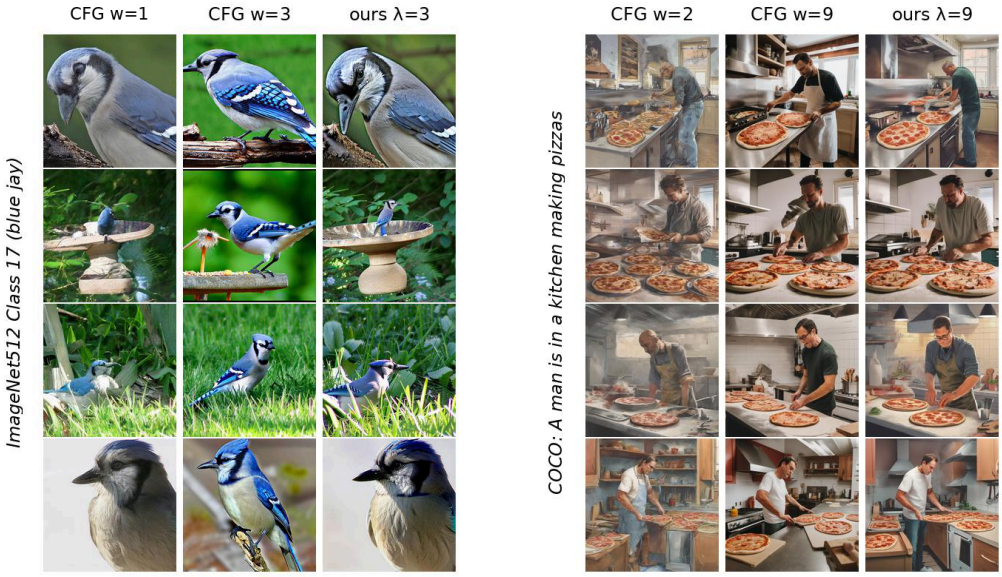


Figure 11: Qualitative comparison of constant and learned guidance schedules. Left: ImageNet class-conditional samples under $w = 1$, constant guidance $w = 3$, and our learned schedule with $\lambda = 3$. Right: COCO text-to-image samples for the prompt "A man is in a kitchen making pizzas," comparing constant guidance $w = 2$, constant guidance $w = 9$, and our learned schedule with $\lambda = 9$.

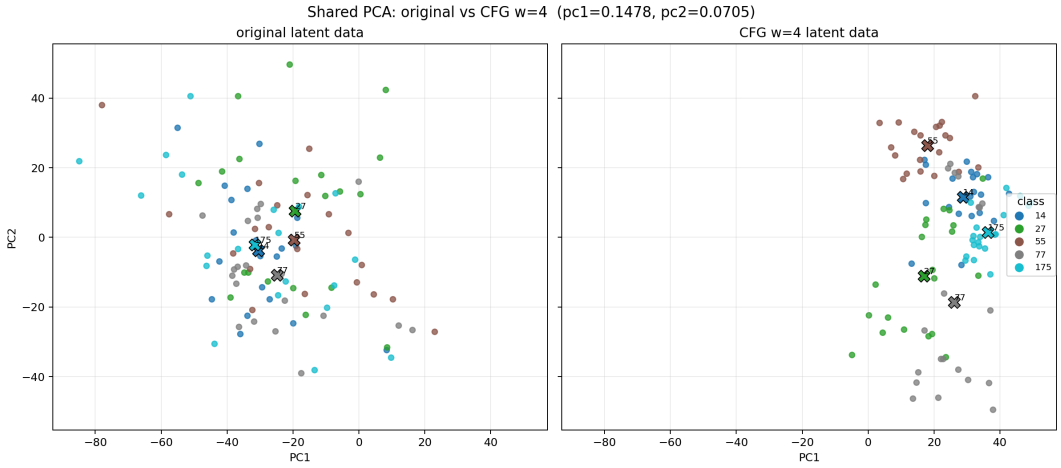


Figure 12: Shared PCA visualization of the original latent data and the latent data generated with CFG weight $w = 4$. Both panels are projected onto the same PCA basis. The left panel shows the original latent data, while the right panel shows the CFG-generated latent data. Colored markers indicate class membership, and large crossed markers indicate class means. The CFG distribution exhibits stronger class separation and concentration in the shared PCA space.

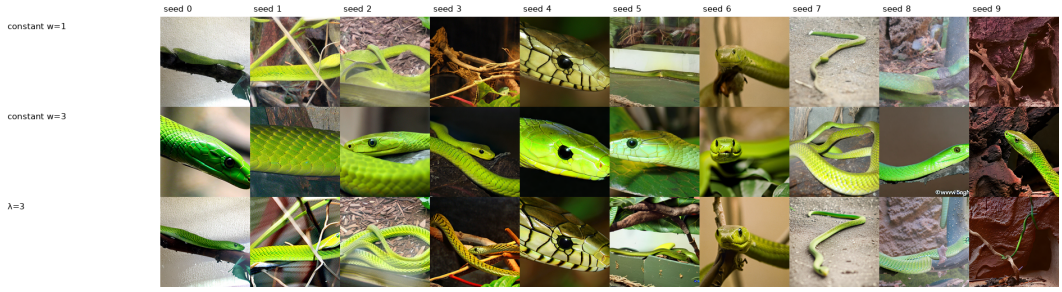


Figure 13: Qualitative comparison on ImageNet class 64, green mamba, using the same ten random seeds. The first row shows samples generated with constant guidance weight $w = 1$. The second row shows samples generated with constant guidance weight $w = 3$. The third row shows samples generated with the proposed step-wise optimized guidance schedule with tradeoff parameter $\lambda = 3$. Compared with $w = 1$, stronger guidance improves class consistency and visual sharpness. Compared with constant $w = 3$, the optimized schedule preserves similar semantic consistency while maintaining more seed-level variation in pose, background, and composition.