Opening the Black Box of Classifier-Free Guidance via Information Bottleneck

Jiayang Gao[1,2,*], **Tianyi Zheng**[2,*], **Jiayang Zou**[3],
Fengxiang Yang[2], **Luyao Fan**[1], **Lv Tang** [4], **Bo Li**[2,†], **Jia Wang**[1,†]
[1] Shanghai Jiao Tong University, Shanghai, China
[2]vivo BlueImage Lab, vivo Mobile Communication Co., Ltd., China
[3] Stanford University, California, USA
[4] Michael Smith Laboratories, University of British Columbia
{gjy0515, jiawang}@sjtu.edu.cn, {zhengtianyi, libra}@vivo.com, jyangzou@stanford.edu

Abstract

Classifier-free guidance (CFG) is a standard technique for improving conditional generation in diffusion models, yet its activation schedule is usually selected heuristically. In practice, most existing CFG-based methods rely on a pre-defined activation policy throughout inference, while recent dynamic variants often depend on hand-designed schedules without a principled theoretical explanation of when conditional information is most useful. In order to understand how the informational relevance between conditioning inputs and the score function evolves over the course of generation, we explore when guidance should be activated from an information-theoretic perspective. In this work, we formulate the diffusion trajectory as a time-dependent information bottleneck and show that the effectiveness of conditional information is not uniform throughout sampling, but becomes significantly stronger after a critical stage characterized by a Fisher-information balance condition. Motivated by this insight, we propose a training-free dynamic CFG strategy for inference. Rather than prescribing a fixed schedule in advance, our method constructs a stochastic process from conditional and unconditional score statistics and activates guidance when the estimated guidance gain crosses a prescribed threshold. This yields an adaptive activation rule that aligns condition injection with the underlying generation dynamics, while also recovering interval-based guidance as a limiting special case. Extensive experiments across diverse architectures and tasks, demonstrate that our method is effective, robust, and broadly applicable. Beyond empirical improvements, our framework provides an interpretation of dynamic guidance methods and a principled foundation for inference-time control in conditional diffusion models.

1 Introduction

Diffusion models [1, 2, 3, 4] have become a major paradigm for high-quality image generation [5], text-to-image synthesis [6] and 3D generation [7]. With the development of conditional diffusion models, the generation process can be effectively controlled by class labels [8, 5], textual descriptions [9, 10], or other forms of prior information. Among existing conditioning methods, Classifier-Free Guidance (CFG) [8] has become one of the most widely used mechanisms for conditional generation. Most existing CFG-based methods are applied with a fixed guidance strategy throughout inference, implicitly assuming that conditional information is equally important across all denoising stages. However, this assumption may be overly simplistic, as the relevance of conditional information is

∗ These authors contributed equally to this work.
† Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

likely to evolve over time, which is a phenomenon that remains insufficiently understood. Although several recent methods have attempted to improve this design through dynamic guidance strategies, such as time-dependent reweighting [11, 12, 13, 14], fixed interval activation [15], or modified guidance dynamics [16, 17], these approaches still rely largely on heuristic schedules. In particular, it remains unclear how the informational relevance between conditioning inputs and the score function evolves during generation, and hence when conditional guidance should be activated most effectively.

Diffusion models generate samples by learning a reverse denoising process that progressively reconstructs the target data distribution. For conditional generation, the key question is how to inject conditioning information into this process effectively. Early approaches such as classifier guidance [5], as well as more recent TFG-style methods [18, 19], use external classifiers to provide conditional gradients, but they often struggle to remain stable under heavy noise. By contrast, CFG [8] incorporates conditional information directly into training by jointly learning conditional and unconditional branches, making it more robust and practically appealing. Based on CFG, a series of variant methods replaces the unconditional output with alternative degraded references, such as the prediction of an under-trained conditional network [20] or that of a conditional network with perturbed conditioning inputs [21]. These approaches preserve the general guidance form of comparing a target conditional prediction against a weaker reference, while modifying how the reference branch is constructed.

Taken together, these conditioning and guidance strategies highlight that the effectiveness of guidance depends not only on how conditional information is injected, but also on when it becomes most relevant to the denoising dynamics. This raises a more fundamental question: **what are the benefits of properly scheduling the injection of conditional information, and how can this be understood from a theoretical perspective?** To answer this, we investigate how conditional information propagates through a conditional diffusion model, and identify when it becomes most influential. We approach this problem through the lens of the information bottleneck, enabling a principled analysis of how conditioning signals are preserved and utilized across the denoising trajectory. In particular, we study the relationship between conditioning inputs and the score function, and establish a connection between mutual information and score-related statistics. Our theoretical analysis reveals that the effectiveness of conditional information is not uniform throughout inference; instead, it becomes significantly stronger only after a critical stage. Correspondingly, the discrepancy between conditional and unconditional predictions, together with changes in score norms, provides observable signatures of this phenomenon. These theoretical findings are consistent with empirical observations, suggesting that uniformly applying a fixed CFG strategy across the entire inference process is suboptimal.

Building on these insights, we propose a dynamic CFG switching strategy for inference. Our key idea is to activate conditional guidance only when the generation process reaches the information bottleneck, so that the timing of condition injection is explicitly aligned with the theoretical analysis. The method is training-free and can be readily applied to a broad range of diffusion models and generation tasks, establishing a principled design criterion for CFG-based generation highlighting the differences and connections between various dynamic guidance strategies. In particular, it clarifies Interval Guidance [15] can be viewed through the same theoretical lens, offering insights into how we choose time to activate CFG. Extensive experiments on SiT [22, 23], DiT [24], FLUX [25], and Stable Diffusion [6, 26] show that our approach is both effective and broadly applicable. Our contributions are summarized as follows:

- **Information-Theoretic Perspective:** We introduce a time-dependent information-bottleneck framework to investigate the role of conditional information throughout the diffusion trajectory. By connecting mutual information evolution with Fisher information, we derive a balance condition that characterizes an information-theoretically motivated transition region for CFG activation.
- **Principled Activation Criterion:** Building on the theoretical analysis, we derive an observable guidance-gain statistic from the difference between conditional and unconditional score norms. We further develop a rescaled threshold criterion that improves numerical conditioning and provides an interpretable, model-dependent mechanism for identifying effective guidance activation regions.
- **Adaptive Guidance Strategy:** We propose a training-free and plug-and-play dynamic CFG strategy that determines the activation time online through a threshold-crossing rule. Our framework accommodates both batch-level and single-sample inference, while recovering

fixed-interval guidance in the large-batch concentration regime, thereby establishing a connection between adaptive and predetermined guidance schedules.

- **Extensive Empirical Validation:** We conduct comprehensive experiments across diverse diffusion and flow-based architectures, including SiT, DiT, SD3, and FLUX. The results demonstrate improvements in generation quality under various evaluation settings, supported by multi-seed stability analyses, alternative activation criteria, and high-resolution single-sample inference experiments.

2 Background

Diffusion Models and CFG. Diffusion models [1, 2, 3, 4] generate data by reversing a gradual noising process. In the continuous-time formulation, the forward diffusion process can be described by the following stochastic differential equation (SDE) [4]:

$$\mathrm{d}x_t = f(x_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}w_t, \tag{1}$$

where $f(x_t, t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient, and w_t is a standard Wiener process. The corresponding reverse-time SDE [27] is given by

$$\mathrm{d}x_t = \left[f(x_t, t) - \frac{1}{2}\left(g^2(t) + \sigma^2(t)\right)\nabla \log p_t(x_t)\right]\mathrm{d}t + \sigma(t)\,\mathrm{d}\bar{w}_t, \tag{2}$$

where $\bar{w}_t$ denotes a standard reverse-time Wiener process, and $\sigma(t)$ specifies the noise scale. The score function $\nabla \log p_t(x_t)$ is the target quantity that the network aims to predict [4]. For conditional generation, Classifier-Free Guidance (CFG) combines conditional and unconditional predictions to steer sampling:

$$\epsilon_\omega(x_t, t, c) = \omega\,\epsilon_\theta(x_t, t, c) - (\omega - 1)\epsilon_\theta(x_t, t, \varnothing), \tag{3}$$

where $\epsilon_\theta(x_t, t, \cdot)$ denotes the conditional noise predictor and $\omega \geq 1$ controls the guidance strength. We defer a more detailed discussion of representative guidance variants to Appendix A.

Mutual Information, Fisher Information, and Information Bottleneck (IB). To analyze the role of conditional information, we adopt an information-theoretic perspective. The mutual information [28] between random variables X and Y is defined as

$$I(X;Y) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right], \tag{4}$$

which measures their statistical dependence. We also consider Fisher information,

$$J(X) = \mathbb{E}_{p(x)}\left[\|\nabla_x \log p(x)\|^2\right], \quad J(X|Y) = \mathbb{E}_{p(x,y)}\left[\|\nabla_x \log p(x|y)\|^2\right] \tag{5}$$

which is directly related to score functions. Analogous to the decomposition of mutual information, the Fisher mutual information between X and Y can be defined as

$$J(X;Y) = J(X|Y) - J(X), \tag{6}$$

which quantifies the amount of Fisher information about X explained by conditioning on Y. At last we introduce information bottleneck principle [29], which formulates representation learning as a trade-off between compression and relevance. Given random variables X and C, IB seeks an intermediate representation T that minimizes

$$\mathcal{L}_{\text{IB}} = I(X;T) - \beta I(T;C), \quad \beta > 0, \tag{7}$$

where β is a Lagrange multiplier controlling the balance between compression of the input X and preservation of information relevant to the target variable C. Beyond its original formulation, the IB principle has also been used to interpret the training dynamics of neural networks [30]. In this view, a neural network induces a Markov chain $Y \to X \to \hat{X}$, where $\hat{X}$ denotes the network representation or output. The learning objective can then be cast as an information bottleneck problem of the form

$$\mathcal{L}\left[p(\hat{x}|x)\right] = I(X;\hat{X}) - \beta I(\hat{X};Y), \tag{8}$$

which trades off the complexity of the representation $\hat{X}$ against its predictive power with respect to the labels Y. In our setting, this perspective provides a useful lens for understanding when conditional information becomes most important during generation.

3 Theoretical Analysis

3.1 Information Bottleneck on Conditional Diffusion Model

To analyze classifier-free guidance (CFG) from an information-theoretic perspective, we view the diffusion trajectory as a Markov chain $C \to X_0 \to X_t$, where C represents the conditioning information (such as class labels in image classification tasks or textual prompts in text-to-image generation), and X_t is the noisy latent variable at diffusion time step t. Thus we can formulate CFG as a time-dependent information bottleneck problem. Specifically, we consider the objective

$$\inf_t L(t) = I(X_0; X_t) - \beta I(X_t; C), \quad \beta > 0, \tag{9}$$

where t denotes the diffusion time and β controls the relative emphasis on conditional information.

We first characterize the necessary condition satisfied by an interior minimizer of equation 9.

Theorem 1 (Information Bottleneck Optimality for Conditional Diffusion Models)**.** *Assume that X_t follows the forward SDE equation 1, and that the mutual information in equation 9 are differentiable with respect to t. Assume that by choosing an appropriate β, the minimizer t^* of equation 9 lies in the interior of the interval $(0, +\infty)$.Then t^* satisfies*

$$\beta J(X_{t^*}; C) = J(X_{t^*}; X_0), \tag{10}$$

It can also be written as

$$\beta \left(J(X_{t^*}|C) - J(X_{t^*}) \right) = J(X_{t^*}|X_0) - J(X_{t^*}). \tag{11}$$

Proof. See Appendix B.2. □

We now specialize the result to an n-dimensional Ornstein–Uhlenbeck (OU) diffusion process

$$\mathrm{d}X_t = f(t)X_t \mathrm{d}t + g(t)\mathrm{d}w_t, \tag{12}$$

whose solution is

$$X_t = e^{\int_0^t f(s)\mathrm{d}s} X_0 + \int_0^t e^{\int_u^t f(s)\mathrm{d}s} \boldsymbol{g}(u)\mathrm{d}w_u. \tag{13}$$

After reparameterize, X_t can be written as

$$X_t = e^{\int_0^t f(s)\mathrm{d}s} X_0 + \sqrt{m(t)}Z, \tag{14}$$

where $Z \sim \mathcal{N}(0, I)$, $m(t) = \int_0^t \exp\left(2\int_s^t f(r)\mathrm{d}r\right) g^2(s)\mathrm{d}s$. Furthermore, assuming a learned score function $s_\theta(x, t, c) \approx \nabla \log p_t(x|c)$, we approximate the Fisher information by

$$J(X_t|C) \approx \mathbb{E}_{x_t, c\sim p_t(x,c)} \left[\|s_\theta(x_t, t, c)\|^2 \right], \quad J(X_t) \approx \mathbb{E}_{x_t, c\sim p_t(x,c)} \left[\|s_\theta(x_t, t, \varnothing)\|^2 \right]. \tag{15}$$

For such processes, the conditional Fisher information admits a closed-form expression: $J(X_t|X_0) = \frac{n}{m(t)}$. Substituting these expressions into Theorem 1, the optimality condition at t^* becomes

$$\beta(\mathbb{E}[\|s_\theta(x_{t^*}, t^*, c)\|^2] - \mathbb{E}[\|s_\theta(x_{t^*}, t^*, \varnothing)\|^2]) = \frac{n}{m(t^*)} - \mathbb{E}[\|s_\theta(x_{t^*}, t^*, \varnothing)\|^2]. \tag{16}$$

Equation 16 characterizes the diffusion time at which the information bottleneck objective achieves a balance between conditional relevance and information compression. In principle, this condition provides a criterion for selecting an appropriate time scale at which CFG should be activated.

3.2 Extension and limitation on Flow Models

Our information bottleneck analysis can be extended to a class of stochastic flow/interpolation models. In particular, consider

$$X_t = \alpha_t X_0 + \sigma_t X_1, \qquad t \in [0, 1], \tag{17}$$

where X_0 denotes the target distribution, X_1 denotes the prior distribution, and $\alpha_0 = \sigma_1 = 1,\ \alpha_1 = \sigma_0 = 0$. Assume that $X_1 \sim \mathcal{N}(0, I)$ is independent of X_0.

Theorem 2 (Information Bottleneck Optimality for Flow Models)**.** *Assume that X_t follows the forward SDE equation 17, and that the mutual information in equation 9 are differentiable with respect to t. Assume that by choosing an appropriate β, the minimizer t^* of equation 9 lies in the interior of the interval $(0, 1)$.Then t^* satisfies*

$$\beta J(X_{t^*}; C) = J(X_{t^*}; X_0). \tag{18}$$

Equivalently,

$$\beta \left(J(X_{t^*} \mid C) - J(X_{t^*}) \right) = J(X_{t^*} \mid X_0) - J(X_{t^*}). \tag{19}$$

Proof. See Appendix B.3. □

Since the conditional distribution is $X_t|X_0 \sim \mathcal{N}(\alpha_t X_0, \sigma_t^2 I)$, we have $J(X_t|X_0) = \frac{n}{\sigma_t^2}$. Therefore, the optimality condition at t^* becomes

$$\beta\Big(\mathbb{E}[\|s_\theta(x_{t^*}, t^*, c)\|^2] - \mathbb{E}[\|s_\theta(x_{t^*}, t^*, \varnothing)\|^2]\Big) = \frac{n}{\sigma_{t^*}^2} - \mathbb{E}[\|s_\theta(x_{t^*}, t^*, \varnothing)\|^2]. \tag{20}$$

Limitation. Theorem 2 applies only to *stochastic* interpolation paths with an independent Gaussian endpoint. It does not directly cover deterministic couplings, such as n-rectified flow [31] for $n \geq 2$, where X_t is a deterministic function of X_0 and the standard continuous information bottleneck may degenerate. Nevertheless, equation 20 can still be used as a practical heuristic in such models, although without the same rigorous information-theoretic justification. As for how to estimate score on Rectified Flow, a detailed discussion is provided in Appendix C.

3.3 Justification via Hyperparameter Rescale

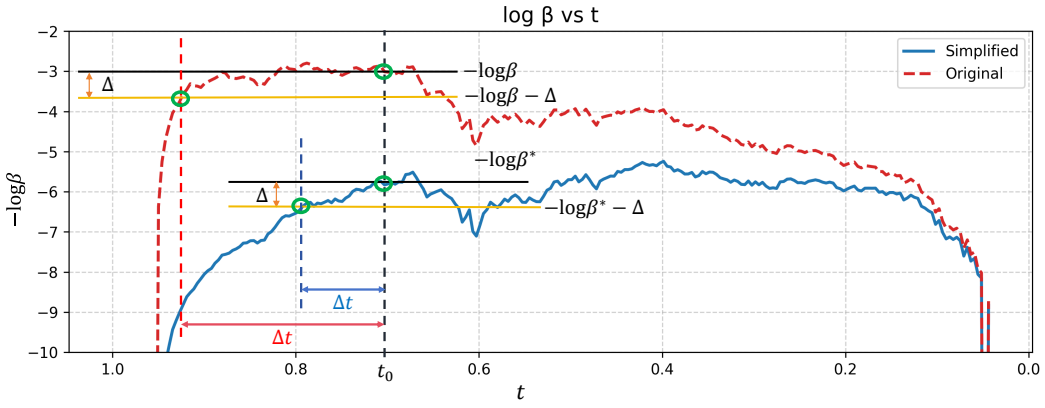


Figure 1: Sensitivity of the solution t^* to the hyperparameter in the original and simplified formulations. The original parameterization (red Δt) exhibits a much flatter $\log \beta$–t relationship than the rescaled version (blue Δt), indicating higher sensitivity to perturbations. Notably, for a fixed perturbation in $\log \beta$, the resulting deviation in t (Δt) is significantly larger in the original formulation. Experiments are conducted on SiT-XL/2.

Despite its conceptual appeal, directly solving equation 16 in practice is challenging. The expectations in equation 16 are estimated using finite mini-batches during sampling, and involve squared score magnitudes, which are known to exhibit high variance. As a result, the empirical estimates of the Fisher information terms can be noisy and biased, especially at early diffusion times where score norms vary rapidly. To mitigate this issue, we propose a parameter renormalization perspective to solve the problem. Consider the original bottleneck objective (equation 21) and the simplified objective (equation 22) defined with a surrogate scale β^*:

$$\text{Original:} \quad \beta \cdot \Delta\mathcal{J}(t) = \frac{n}{m(t)} - \mathbb{E}[\|s_\theta(x_t, t, \varnothing)\|^2], \tag{21}$$

$$\text{Simplified:} \quad \beta^* \cdot \Delta\mathcal{J}(t) = \frac{n}{m(t)}, \tag{22}$$

where $\Delta\mathcal{J}(t) = \mathbb{E}[\|s_\theta(x_t, t, c)\|^2] - \mathbb{E}[\|s_\theta(x_t, t, \varnothing)\|^2]$ denotes the conditional guidance gain.

We define the ratio of the marginal Fisher information to the noise kernel Fisher information as $\eta(t)$:

$$\eta(t) := \frac{\mathbb{E}[\|s_\theta(x_t, t, \varnothing)\|^2]}{n/m(t)} = \frac{J(X_t)}{J(X_t|X_0)}. \tag{23}$$

By substituting $\eta(t)$ into equation 21, we can rewrite the original equilibrium condition as:

$$\beta \cdot \Delta\mathcal{J}(t) = \frac{n}{m(t)}(1 - \eta(t)) \iff \left(\frac{\beta}{1-\eta(t)}\right) \cdot \Delta\mathcal{J}(t) = \frac{n}{m(t)}. \tag{24}$$

Comparing this with equation 22, we establish a direct mapping between the original hyperparameter β and the simplified hyperparameter β^*:

$$\beta^* = \frac{\beta}{1-\eta(t)}. \tag{25}$$

To further analyze the stability of the two formulations, we examine the sensitivity of the solution t^* with respect to perturbations in the hyperparameter. Figure 1 shows the relationship between $\log \beta$ and the corresponding solution t for the original objective, as well as $\log \beta^*$ for the simplified objective. We observe that the original parameterization exhibits a much flatter slope, which suggests that the inverse mapping from β to t^* is poorly conditioned, making the root-finding procedure unstable in practice. In contrast, the simplified formulation yields a significantly steeper curve, implying improved conditioning and substantially reduced sensitivity to hyperparameter perturbations. As a result, it enables more robust and reliable tuning in practice.

Thus, our objective shifts from identifying the exact physical β to determining the *effective* scale β^* that satisfies the simplified bottleneck condition. Solving the simplified problem with β^* yields a solution t^* that is asymptotically consistent with the original problem under β. Since β is already an empirical hyperparameter tuned for sample quality, absorbing the scaling factor $(1-\eta(t))^{-1}$ into the selection of β incurs no loss of generality. This allows us to use the computationally accurate equation 22 as a proxy for the exact information bottleneck.

4 Method

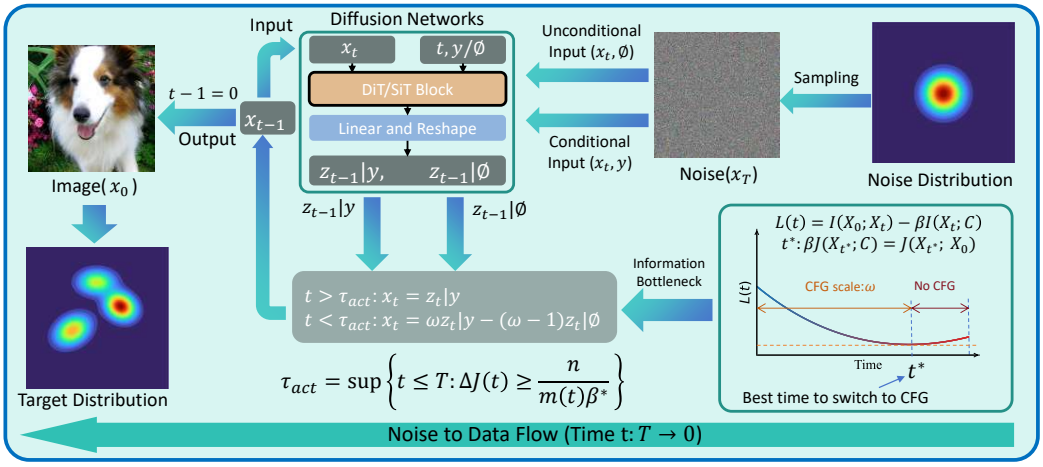


Figure 2: Overview of our adaptive CFG activation pipeline. Starting from noise, the diffusion model jointly evaluates conditional and unconditional scores at each reverse step, computes a guidance-gain statistic $\Delta\mathcal{J}(t)$, and activates classifier-free guidance only after the stopping criterion is met. The illustration uses a DiT/SiT-style diffusion backbone.

Motivated by the theoretical and empirical analysis in Section 3, we propose a practical procedure to determine when classifier-free guidance should be activated. Unlike fixed schedules, our method operates dynamically on a per-batch basis. Rather than explicitly solving the equilibrium condition in

continuous time, we construct a stochastic process from mini-batch statistics of the learned score function. We then activate guidance at the first time this process crosses a prescribed threshold. An overview of the proposed sampling procedure is shown in Figure 2. The figure illustrates how our method augments the standard reverse diffusion process with a batch-wise activation rule: at each step, the model jointly evaluates conditional and unconditional scores, estimates the guidance gain from their batch statistics, and turns on CFG only after the estimated gain exceeds the threshold implied by our information-bottleneck analysis.

At each diffusion step t, we perform a single forward pass by concatenating conditional and unconditional inputs within the same mini-batch, yielding score estimates $s_\theta(x_t, t, c)$ and $s_\theta(x_t, t, \varnothing)$. This allows us to define the batchwise estimators

$$\widehat{\mathcal{J}}_{\mathrm{cond}}(t) = \frac{1}{B}\sum_{i=1}^{B} \|s_\theta(x_t^{(i)}, t, c^{(i)})\|^2, \quad \widehat{\mathcal{J}}_{\mathrm{uncond}}(t) = \frac{1}{B}\sum_{i=1}^{B} \|s_\theta(x_t^{(i)}, t, \varnothing)\|^2, \tag{26}$$

which serve as Monte Carlo approximations to the conditional and marginal Fisher information, respectively. We then consider the stochastic process

$$\Delta\widehat{\mathcal{J}}(t) = \widehat{\mathcal{J}}_{\mathrm{cond}}(t) - \widehat{\mathcal{J}}_{\mathrm{uncond}}(t), \tag{27}$$

which acts as a proxy for the instantaneous guidance gain predicted by the information bottleneck condition. Since the generative process of diffusion models proceeds *backward in time* from the noise prior at T to the data distribution at 0, it is natural to perform the threshold check in the same reverse temporal order. Specifically, we perform a backward scan from the final diffusion step T to 0 and select the *latest* time step (i.e., the first encountered in the reverse loop) where the guidance gain exceeds the threshold. Formally, this defines a *backward stopping time*

$$\tau_{\mathrm{activate}} = \sup\left\{ t \le T : \ \Delta\widehat{\mathcal{J}}(t) \ge \frac{n}{m(t)\beta^*} \right\}.$$

Guidance is then activated for all subsequent steps in the forward progression of the generative process (i.e., for all $t \le \tau_{\mathrm{activate}}$). Algorithm 1 outlines our adaptive guidance activation procedure.

Discussion: Why activate CFG near the information bottleneck?

Activating classifier-free guidance (CFG) near the information bottleneck balances data fidelity and conditional relevance. The information bottleneck objective

$$L(t) = I(X_0;\, X_t) - \beta\, I(X_t;\, C)$$

captures the balance between retaining information from the data distribution ($I(X_0; X_t)$) and incorporating information from the conditioning variable ($I(X_t; C)$). The optimal time for guidance activation corresponds to the point where this balance reaches a critical threshold.

In the early stages of the reverse diffusion process ($t \approx T$), the sample is dominated by noise, and both $I(X_0; X_t)$ and $I(X_t; C)$ are small. At this point, applying CFG would amplify noise rather than useful structure. As the process progresses, $I(X_0; X_t)$ increases, while $I(X_t; C)$ remains weak. This causes the guidance signal to be less effective. However, once $I(X_t; C)$ starts to dominate, $L(t)$ reaches a minimum at a critical point t^*. Here, data fidelity and conditional information are optimally balanced for guidance.

Consequently, the guidance term $\omega\,(\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t, \varnothing))$ in equation 3 acts as a control signal that is most effective at t^*. Activating CFG at the minimum of $L(t)$ ensures the model leverages the conditional signal reliably without compromising the underlying data structure.

Algorithm 1 Reverse Diffusion with Adaptive Guidance Activation

Require: $\{\boldsymbol{x}_T^{(i)}\}_{i=1}^B \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I}_n)$, $\omega, \beta^* > 0$, $\tau_{\text{back}} = \text{undefined}$.

for $t = T$ **to** 1 **do**
  Calculate noise predictions $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t^{(i)})$, $\hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t^{(i)})$.
  $\frac{m(t)}{n}\Delta\widehat{\mathcal{J}}(t) \leftarrow \frac{1}{Bn}\sum_{i=1}^B \left(\|\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(x_t^{(i)})\|^2 - \|\hat{\boldsymbol{\epsilon}}_{\varnothing}(x_t^{(i)})\|^2\right)$ ▷ Compute for current batch
  if τ_{back} is undefined **then**
    if $\Delta\widehat{\mathcal{J}}(t) \geq \frac{n}{\beta^* m(t)}$ **then**
      $\tau_{\text{back}} \leftarrow t$
    end if
  end if
  if τ_{back} is defined **then**
    $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) \leftarrow \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t) + \omega(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t) - \hat{\boldsymbol{\epsilon}}_{\varnothing}(\boldsymbol{x}_t))$, update x_{t-1}
  else
    $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}^{\omega}(\boldsymbol{x}_t) \leftarrow \hat{\boldsymbol{\epsilon}}_{\boldsymbol{c}}(\boldsymbol{x}_t)$, update x_{t-1}
  end if
end for
return x_0

Relationship to Interval Guidance [15]. Our framework recovers Interval Guidance in the large-batch regime. As the batch size grows to ∞, the mini-batch score statistics in our thresholding rule concentrate, causing the random activation time τ_{activate} to approach a deterministic boundary common to all samples. This fixed boundary can be identified with the activation endpoint t_h used in Interval Guidance. Compared with Interval Guidance, however, our method does not require manually prescribing this boundary in finite-batch settings; instead, it infers the activation time adaptively from mini-batch score statistics, and the resulting rule is grounded in our information-bottleneck analysis. Furthermore, we explore the statistics properties of τ_{activate} in Appendix E.

5 Results

5.1 Experimental Setup

Datasets and Models. We evaluate our method on class-conditional image generation and text-to-image synthesis. We conduct experiments on ImageNet [32] and MS-COCO [33]. All models are based on advanced diffusion formula, including Flow Matching [34] (SiT [22, 23]), DDPM [35] (DiT [24]) and Rectified Flow [31] (Stable Diffusion [6], FLUX [25]), using official pre-trained weights.

Evaluation Metrics and Implementation Details. We use FID [36], sFID [37] and IS [38] to assess both fidelity and conditional alignment. We use FID [36], Clip score [39], HPSv2 [40] and Aesthetic score [41] for text-to-image tasks. All experiments are run on NVIDIA GPUs.

5.2 Experimental Results

Results on Class-Conditional Tasks. In Table 1, we evaluate our method on both DiT-XL/2 and SiT-XL/2 models, tested on the ImageNet benchmarks (256 and 512 resolutions) for class-conditional image generation. Our method demonstrates consistent improvements across all metrics when applied to both DiT and SiT models, especially in FID and IS scores.

For DiT-XL/2, our method outperforms the baseline across both resolutions, with a notable FID score of **2.06** on ImageNet-256 and **6.49** on ImageNet-512, showing strong performance. Similarly, for SiT-XL/2, our method yields the best FID score of **1.41** for SDE sampler and **1.47** for ODE sampler, surpassing the baseline and achieving a significant boost in IS scores. These results highlight the versatility and effectiveness of our approach, which consistently enhances the performance of both DiT and SiT models across different resolutions and datasets.

Table 1: **Quantitative comparison on Class-Conditional tasks.** Comparison of different evaluation metrics on Class-Conditional ImageNet datasets with different diffusion architectures. **Bold** indicates the best result, and underlined indicates the second best.

ImageNet (Default $B = 32$)			
Model-256, 50k samples, 250 inference timesteps	FID↓	IS↑	sFID↓
DiT-XL/2 ($\omega = 1.5$, SDE sampler)	2.29	276.8	4.6
DiT-XL/2 ($\omega = 1.6$)	2.71	**303.2**	4.5
DiT-XL/2 + Ours($\omega = 1.6, \log \beta^* = 8.0$)	**2.06**	280.9	**4.5**
SiT-XL/2 (REPA)($\omega = 1.35$, SDE)	1.80	284.0	**4.5**
SiT-XL/2 (REPA, Interval) ($\omega = 1.8, t_l = 0, t_h = 0.7$)	1.42	305.7	4.7
SiT-XL/2 (REPA) ($\omega = 1.9$)	4.67	**388.3**	5.5
SiT-XL/2 (REPA) + Ours ($\omega = 1.9, \log \beta^* = 5.9, B = 48$)	**1.41**	306.6	4.7
SiT-XL/2 (REPA)($\omega = 1.8, t_h = 0.7$, ODE)	1.56	283.0	4.6
SiT-XL/2 (REPA) + Ours ($\omega = 1.8, \log \beta^* = 6.5$, ODE)	**1.47**	**291.2**	**4.4**
Model-512, 10k samples, 100 inference timesteps			
DiT-XL/2 ($\omega = 1.5$, SDE)	6.81	229.5	20.0
DiT-XL/2 ($\omega = 1.7$)	7.08	**275.5**	19.2
DiT-XL/2 + Ours($\omega = 1.7, \log \beta^* = 8.5, B = 16$)	**6.49**	241.4	20.0

Table 2: **Ablation comparison.** Comparison of different evaluation metrics on Class-Conditional ImageNet datasets with different batch size B. Unless otherwise specified, all other settings are identical to those in Table 1.

ImageNet-256 10k samples, 50 inference timesteps		
SiT-XL/2	FID↓	IS↑
Baseline (SDE)	5.09	287.4
Ours($B = 32$)	4.92	299.7
Ours($B = 16$)	4.89	296.1
Ours($B = 8$)	4.96	294.6
Ours($B = 4$)	4.95	294.9

Table 3: **Quantitative comparison on text-to-image tasks.** Quantitative results on SD3 and FLUX.1 under different guidance configurations. We report FID, HPSv2.1, CLIP Score, and Aesthetic Score where available. For FLUX.1 at 1024×1024, we additionally evaluate single-sample inference ($B = 1$). "–" denotes unavailable results.

Model	Approach	FID↓	HPS v2.1					CLIP↑	Aesthetic↑
			paintings↑	photo↑	concept-art↑	anime↑	average↑		
SD 3 (1024, 28 steps)	CFG($\omega = 7.5, B = 4$)	25.98	30.99	28.39	30.63	31.31	30.33	31.83	5.44
	Interval($\omega = 7.5,\ t_h = 0.7T$)	–	27.66	24.76	27.57	28.55	27.13	31.41	5.21
	Ours($\omega = 7.5,\ \log \beta^* = 10.0$)	**24.88**	**31.12**	**28.57**	**30.98**	**31.51**	**30.55**	**31.89**	**5.45**
FLUX.1 (256, 50 steps)	CFG($\omega = 1.5, B = 64$)	30.93	29.77	27.18	29.21	30.80	29.24	**31.10**	5.46
	Ours($\omega = 1.5,\ \log \beta^* = 8.0$)	**30.36**	**29.87**	**27.38**	**29.26**	**30.94**	**29.36**	30.93	**5.48**
FLUX.1 (1024, 28 steps)	CFG($\omega = 7.5$)	–	20.41	18.80	20.71	20.60	20.13	–	–
	Ours($\omega = 7.5,\ \log \beta^* = 8.0,\ B = 16$)	–	**30.84**	**29.49**	**30.55**	**32.05**	**30.73**	–	–
	Ours($\omega = 7.5,\ \log \beta^* = 8.0,\ B = 1$)	–	30.36	29.08	30.21	31.50	30.29	–	–

Results on T2I Tasks. As shown in Table 3, our method achieves consistent improvements over standard CFG on SD3 in terms of FID, HPSv2.1, CLIP Score, and Aesthetic Score. On FLUX.1 at 256×256 resolution, our method improves FID and HPSv2.1, with a slight reduction in CLIP Score.

We further evaluate FLUX.1 at 1024×1024 resolution with $\omega = 7.5$. Our method achieves an average HPSv2.1 score of 30.73 with $B = 16$, compared with 20.13 for standard CFG. Importantly, it also achieves 30.29 under single-sample inference ($B = 1$), demonstrating its applicability without requiring multiple generated samples.

Ablation Study. As shown in Table 2, we investigate the impact of batch size on generation performance. Our method consistently outperforms the baseline across all tested batch sizes ($B \in \{4, 8, 16, 32\}$), achieving lower FID scores in every setting. To further assess the reproducibility of these improvements, we conduct multi-seed experiments. As reported in Table 4, our method achieves an FID of 4.98 ± 0.06 and an IS of 298.1 ± 2.6, consistently outperforming Interval Guidance across three random seeds. We further analyze the distribution of the batch-wise activation time τ under different batch sizes. As shown in Table 5, the mean activation time remains stable at approximately 0.68, while its standard deviation decreases as the batch size increases. This observation is consistent with the concentration of batch-level statistics and supports the connection between our adaptive criterion and fixed-interval guidance in the large-batch regime. Finally, we compare our Fisher-derived score-norm difference with an alternative activation statistic based on the conditional–unconditional score-discrepancy MSE. As shown in Table 6, our criterion achieves better FID and IS under the tested configurations, suggesting its practical effectiveness in identifying a suitable guidance activation point. Additional results are provided in Appendix E.

Table 4: **Multi-seed stability.** FID and IS over three seeds on SiT-XL/2.

Method	Metric	Seed 0	Seed 1	Seed 2	Mean ± Std.
Interval	FID↓	5.09	5.25	5.19	5.18 ± 0.08
	IS↑	287.4	280.9	284.1	284.1 ± 3.3
Ours	FID↓	4.92	5.05	4.98	$\mathbf{4.98 \pm 0.06}$
	IS↑	299.7	295.1	299.5	$\mathbf{298.1 \pm 2.6}$

Table 5: **Stopping-time statistics.** Activation time across batch sizes.

Batch	Mean (τ)	Std. (τ)
$B = 8$	0.68	0.107
$B = 16$	0.68	0.086
$B = 32$	0.68	0.054

Table 6: **Activation statistic ablation.** Comparison of switching criteria.

Activation Statistic	FID↓	IS↑
Baseline	5.09	287.4
Score-discrepancy MSE	5.14	287.5
Fisher norm difference (Ours)	**4.92**	**299.7**

Qualitative Comparison. We present visual comparisons of text-to-image generation results using Stable Diffusion 3 and Flux.1 in Figure 3. The red text in the prompts indicates specific semantic details that the baseline model fails to generate correctly. The regions highlighted by boxes illustrate the differences between the methods. As shown, our approach successfully aligns with these complex semantic requirements, generating details that are missing or incorrect in the baseline results.



Figure 3: **Qualitative Comparison.** Visual results on text-to-image tasks using Stable Diffusion 3 and Flux.1. The red text in the prompts highlights specific details that the baseline fails to generate.

6 Conclusion and Future Work

In this work, we study Classifier-Free Guidance (CFG) from Information Bottleneck perspective. We theoretically established that conditional information is not uniformly effective throughout the generation process but becomes critical only after a specific stage characterized by a Fisher-information balance. Based on this insight, we proposed a training-free, dynamic CFG strategy that adaptively activates guidance by monitoring score statistics. Extensive experiments demonstrate that our method consistently improves sample quality and robustness across diverse architectures.

There remain several promising directions for future research. **Theoretically**, as mentioned in Section 3, while our Information Bottleneck analysis extends to stochastic flow models, extending the framework to fully characterize deterministic couplings would be an important next step toward a more complete understanding. **Practically**, while our method supports single-sample inference, batch averaging can improve the stability of the activation statistic. Developing more robust sample-specific criteria and extending the framework to heterogeneous conditioning scenarios remain promising directions for future research.

Acknowledgment. This work is supported by National Nature Science Foundation of China under grant 62671375, U24A20220, and by the Shanghai Municipal Commission of Economy and Informatization, under Grant No. 2024-GZL-RGZN-01008.

References

[1] Jascha Narain Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015.

[2] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Neural Information Processing Systems*, 2019.

[3] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ArXiv*, abs/2010.02502, 2020.

[4] Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020.

[5] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.

[6] 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*, pages 10684–10695, 2022.

[7] Sangmin Woo, Byeongjun Park, Hyojun Go, Jin-Young Kim, and Changick Kim. Harmonyview: Harmonizing consistency and diversity in one-image-to-3d. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10574–10584, 2024.

[8] Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022.

[9] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022.

[10] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

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

[12] Mateusz Poleski, Jacek Tabor, and Przemysław Spurek. Geoguide: Geometric guidance of diffusion models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 297–305. IEEE, 2025.

[13] 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 (CVPR)*, pages 34398–34407, June 2026.

[14] Tianyi Zheng, Jiayang Zou, Peng-Tao Jiang, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Bidirectional beta-tuned diffusion model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):359–373, 2026.

[15] 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. *Advances in Neural Information Processing Systems*, 37:122458–122483, 2024.

[16] Hanyang Wang, Yiyang Liu, Jiawei Chi, Fangfu Liu, Ran Xue, and Yueqi Duan. Cfg-ctrl: Control-based classifier-free diffusion guidance. *arXiv preprint arXiv:2603.03281*, 2026.

[17] Seyedmorteza Sadat, Tobias Vontobel, Farnood Salehi, and Romann M. Weber. Guidance in the frequency domain enables high-fidelity sampling at low cfg scales. *ArXiv*, abs/2506.19713, 2025.

[18] Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. 2024.

[19] Yifei Shen, Xinyang Jiang, Yezhen Wang, Yifan Yang, Dongqi Han, and Dongsheng Li. Understanding and improving training-free loss-based diffusion guidance. *Advances in Neural Information Processing Systems 37*, 2024.

[20] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Proc. NeurIPS*, 2024.

[21] Javad Rajabi, Soroush Mehraban, Seyedmorteza Sadat, and Babak Taati. Token perturbation guidance for diffusion models. *arXiv preprint arXiv:2506.10036*, 2025.

[22] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024.

[23] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *International Conference on Learning Representations*, 2025.

[24] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

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

[26] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.

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

[28] Thomas M. Cover and Joy A. Thomas. Elements of information theory. 2005.

[29] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[30] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. Ieee, 2015.

[31] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *ArXiv*, abs/2209.03003, 2022.

[32] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[34] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ArXiv*, abs/2210.02747, 2022.

[35] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

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

[37] Xin Ding, Yongwei Wang, Zuheng Xu, William J Welch, and Z Jane Wang. Continuous conditional generative adversarial networks: Novel empirical losses and label input mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):8143–8158, 2022.

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

[39] 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.

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

[41] Christoph Schuhmann. Improved aesthetic predictor, 2023.

[42] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5404–5411, 2024.

[43] Runlong Liao, Jian Yu, Baiyu Su, Chi Zhang, Lizhang Chen, and Qiang Liu. Momentum guidance: Plug-and-play guidance for flow models. *arXiv preprint arXiv:2602.20360*, 2026.

[44] Tiancheng Li, Weijian Luo, Zhiyang Chen, Liyuan Ma, and Guo-Jun Qi. Self-guidance: Boosting flow and diffusion generation on their own. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[45] Shayan Mohajer Hamidi and En-Hui Yang. Conditional mutual information based diffusion posterior sampling for solving inverse problems. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pages 1–6, 2025.

[46] Haonan Yuan, Qingyun Sun, Xingcheng Fu, Cheng Ji, and Jianxin Li. Dynamic graph information bottleneck. In *The Web Conference 2024*, 2024.

[47] Yirong Shen, Lu GAN, and Cong Ling. Information theoretic learning for diffusion models with warm start. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[48] Tianyi Zheng, Jiayang Gao, Peng-Tao Jiang, Fengxiang Yang, Ben Wan, Hao Zhang, Jinwei Chen, Jia Wang, and Bo Li. Bidirectional noise injection: Enhancing diffusion models via coordinated input-output perturbation. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 28866–28874. AAAI Press, 2026.

[49] Yirong Shen, Lu Gan, and Cong Ling. Rényi diffusion models. In *Forty-third International Conference on Machine Learning*, 2026.

Appendix

Contents

A Extended Related Work **15**

B Proof of Theorems **16**

B.1 Proof of Lemmas 16

B.2 Proof of Theorem 1 17

B.3 Proof of Theorem 2 18

C Estimation of Score on Rectified Flow **18**

C.1 Asymptotic Consistency Near the Gaussian Endpoint 18

C.2 Consistency of Score Differences via Local PDE Dynamics 19

D Implementation Details and Code **20**

E Additional Analysis and Experiments **22**

E.1 Properties of Stopping Time 22

E.2 Ablation on Parameter $\log \beta^*$ 23

E.3 Additional visual results 24

A Extended Related Work

Conditional control in diffusion models is commonly achieved through classifier guidance (CG, [5]) and classifier-free guidance (CFG, [8]). CG introduces an external classifier to steer generation toward the target condition:

$$\hat{\epsilon}(x_t, t, y) = \epsilon_\theta(x_t, t, \varnothing) + \omega\, \Sigma_\theta(x_t, t) \nabla \log p_t(y \mid x_t), \tag{28}$$

while CFG replaces the classifier with the difference between conditional and unconditional score estimates:

$$\hat{\epsilon}(x_t, t, y) = \omega\, [\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t, \varnothing)] + \epsilon_\theta(x_t, t, \varnothing). \tag{29}$$

In both cases, the guidance scale ω controls the trade-off between fidelity and diversity. Owing to its simplicity and strong empirical performance, CFG has become the dominant guidance strategy in modern diffusion models [4]. However, most existing CFG-based methods apply a fixed guidance rule uniformly across all reverse diffusion steps, and recent studies have shown that this design is often suboptimal.

A broad series of work therefore investigates dynamic or adaptive guidance *strength*. For example, [11] proposes a β-adaptive strategy that enforces weak guidance in the early and late stages while strengthening it in the middle phase of denoising. [17] introduces Frequency-Decoupled Guidance (FDG), which applies different guidance strengths to low- and high-frequency components. [42] rescales CFG to alleviate over-exposure effects, while [12] proposes a geometry-aware normalization strategy to maintain stable guidance magnitude during late denoising. These methods suggest that guidance should vary with time or signal structure, and mainly focus on designing better weighting functions $w(t)$ or modified guidance parameterizations.

To mitigate the computational cost of training distinct conditional and unconditional models, several approaches explore the idea of *degrading* the conditional network to mimic unconditionality. [20] achieves this by utilizing an under-trained state of the network, arguing that early-stopped weights contain less semantic bias and thus behave more unconditionally. In a complementary vein, [21] introduces stochasticity by adding noise to the conditioning tokens directly. By feeding corrupted inputs into the conditional network, the output becomes a degraded approximation of the true unconditional distribution, effectively replacing the need for a separate network.

As for when to activate CFG, a representative example is Interval Guidance [15], which shows that applying CFG over all denoising steps is suboptimal: guidance tends to harm diversity in the high-noise regime, has limited effect in the very low-noise regime, and is most beneficial only over an intermediate interval. Their method therefore restricts guidance to a fixed range of timesteps, improving both sample quality and distributional fidelity. While this line of work highlights the importance of *when* guidance is applied, existing methods still largely rely on hand-designed schedules or fixed activation intervals.

Beyond schedule-based reweighting, another series of work revisits the guidance mechanism itself. CFG-Ctrl [16] formulates the denoising process as a dynamical system and interprets most existing CFG-based methods from a control-theoretic perspective, arguing that purely linear guidance may become unstable at large guidance scales and proposing a sliding-mode-control-based variant to improve robustness. In a different direction, Momentum Guidance [43] modifies the sampling dynamics by incorporating momentum into the CFG-adjusted flow update, providing a training-free and plug-and-play way to improve sampling robustness without retraining the model. Self-Guidance [44] instead derives an additional training-free guidance signal from temporal prediction discrepancies within the model itself, comparing predictions at the current and shifted timesteps and combining the resulting correction with CFG.

On a more theoretical front, several recent works have begun to analyze the role of conditioning in diffusion models through the lens of information theory. [45] proposes an information-theoretic approach that maximizes the conditional mutual information to ensure the generation process retains maximum information about the measurements. Concurrently, [46] proposes the Dynamic Graph Information Bottleneck (DGIB) to learn robust and discriminative representations, defending against attacks on link prediction tasks. These studies align closely with our goal of adaptive activation, as both seek to identify the most informative moments for conditioning.

Recent studies have explored information-theoretic principles to better understand and improve diffusion-based generative modeling. [47, 48] generalize the connection between KL divergence

,Fisher divergence and Wasserstein distance to non-Gaussian isotropic noise perturbations, introducing a warm-started, variance-aware framework for tighter likelihood estimation and more stable diffusion training. More recently, Shen et al. [49] propose Rényi Diffusion Models, which formulate diffusion training through Rényi divergence and derive a generalized score-matching objective to explicitly control the trade-off between sample quality and distribution coverage. While these studies primarily investigate information-theoretic objectives for learning generative distributions, our work focuses on the role of conditional information during inference. Specifically, we formulate the diffusion trajectory as a time-dependent information bottleneck and derive a Fisher-information balance condition to characterize an effective region for CFG activation. This leads to a training-free guidance strategy that dynamically determines when conditional guidance should be applied.

Overall, these studies demonstrate that diffusion guidance can be improved by adjusting its strength, modifying its direction, or altering how guided updates are integrated over time. In contrast, our work focuses on *adaptive guidance activation*: rather than prescribing a fixed schedule or interval in advance, we study when conditional guidance should be activated based on the generation dynamics themselves. Notably, our formulation also provides a unified perspective on interval-based guidance, which can be interpreted as a special case under a limiting setting of our adaptive activation rule.

B Proof of Theorems

B.1 Proof of Lemmas

In this section, we present several auxiliary results that serve as preparatory steps for the proof of Theorem 1 and 2. First, we establish a general entropy evolution formula for diffusion processes. Second, we specialize this result to mutual information under conditioning.

Lemma 1. *Let $X_t \in \mathbb{R}^n$ be the solution of the SDE*

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + g(t)\mathrm{d}w_t. \tag{30}$$

Then

$$\frac{\mathrm{d}}{\mathrm{d}t}H(X_t) = \frac{1}{2}g^2(t)J(X_t) + \mathbb{E}_{p_t}[\nabla \cdot f_t], \tag{31}$$

where p_t is the marginal distribution of X_t. In particular, for the Ornstein–Uhlenbeck (OU) process

$$\mathrm{d}X_t = f(t)X_t\mathrm{d}t + g(t)\mathrm{d}w_t,$$

we have the form:

$$\frac{\mathrm{d}}{\mathrm{d}t}H(X_t) = \frac{1}{2}g^2(t)J(X_t) + \mathbb{E}[\nabla \cdot (f(t)X_t)] \tag{32}$$

$$= \frac{1}{2}g^2(t)J(X_t) + nf(t) \tag{33}$$

Proof. FPE of equation 30 is

$$\partial_t p_t = -\nabla \cdot (f_t p_t) + \frac{1}{2}g^2(t)\Delta p_t,$$

as for differential entropy $H(X_t) = -\int p_t \log p_t \mathrm{d}x$, we obtain

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}H(X_t) &= -\int \partial_t p_t \log p_t \mathrm{d}x - \int \partial_t p_t \mathrm{d}x \\
&= -\int \partial_t p_t \log p_t \mathrm{d}x \\
&= \int \nabla \cdot (f_t p_t) \log p_t \mathrm{d}x - \int \frac{1}{2}g^2(t)\Delta p_t \log p_t \mathrm{d}x,
\end{aligned}$$

then we calculate the terms in the above equation,

$$\begin{aligned}
\int \nabla \cdot (f_t p_t) \log p_t \mathrm{d}x &= (-1)\int \left\langle f_t p_t, \frac{\nabla p_t}{p_t} \right\rangle \\
&= -\int \langle f_t, \nabla p_t \rangle \\
&= \mathbb{E}_{p_t}[\nabla \cdot f_t],
\end{aligned}$$

using $\Delta \log p = \Delta p/p - (\nabla \log p)^2$,

$$\begin{aligned}\int \frac{1}{2} g^2(t) \Delta p_t \log p_t \mathrm{d}x &= \frac{1}{2} g^2(t) \int p_t \Delta \log p_t \\ &= \frac{1}{2} g^2(t) \int p_t (\Delta p_t / p_t - (\nabla \log p_t)^2) \\ &= -\frac{1}{2} g^2(t) J(X_t),\end{aligned}$$

so we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} H(X_t) = \frac{1}{2} g^2(t) J(X_t) + \mathbb{E}_{p_t}[\nabla \cdot f_t].$$

□

Lemma 1 characterizes how differential entropy evolves along a diffusion trajectory. We next leverage this result to study the time evolution of mutual information under conditioning.

Lemma 2 (Information evolution for diffusion models)**.** *We treat the conditioning variable Y as a random variable, which may correspond to C or X_0 in our setting. The output X_t of the SDE equation 30 with condition Y satisfies*

$$\frac{\mathrm{d}}{\mathrm{d}t} I(X_t; Y) = -\frac{1}{2} g^2(t) J(X_t; Y). \tag{34}$$

Proof. Lemma 1 holds for any initial distribution. In particular, if y is the given condition of conditional diffusion models, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} H(X_t|Y=y) = \frac{1}{2} g^2(t) J(X_t|Y=y) + \mathbb{E}_{p_t(\cdot|y)}[\nabla \cdot f_t], \tag{35}$$

where $p_t(\cdot|y)$ is the marginal distribution of X_t given $Y = y$. For any condition Y, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} H(X_t|Y) = \frac{\mathrm{d}}{\mathrm{d}t} \mathbb{E}_{p(y)} H(X_t|Y=y) = \frac{1}{2} g^2(t) J(X_t|Y) + \mathbb{E}_{p_t}[\nabla \cdot f_t], \tag{36}$$

where $p_t(\cdot)$ is the marginal distribution of X_t. Then we conclude that

$$\frac{\mathrm{d}}{\mathrm{d}t} I(X_t; Y) = \frac{\mathrm{d}}{\mathrm{d}t} (H(X_t) - H(X_t|Y)) \tag{37}$$

$$= \frac{1}{2} g^2(t) (J(X_t) - J(X_t|Y)) \tag{38}$$

$$= -\frac{1}{2} g^2(t) J(X_t; Y), \tag{39}$$

as desired. □

We are now ready to complete the proof of Theorem 1 and 2 by applying the above mutual information dynamics to the information bottleneck objective.

B.2 Proof of Theorem 1

In this section, we provide a complete proof of Theorem 1. The proof proceeds in three steps. We apply the lemmas proved in Appendix B.1 to the information bottleneck objective and derive the optimality condition.

Proof of Theorem 1. Let $L(t)$ denote the information bottleneck objective defined in equation 16. If t^* is an interior minimizer of $L(t)$, it must satisfy the first-order optimality condition

$$\frac{\mathrm{d}L(t^*)}{\mathrm{d}t^*} = 0. \tag{40}$$

Applying Lemma 2 to both mutual information terms appearing in $L(t)$, we obtain

$$\beta J(X_{t^*}; C) = J(X_{t^*}; X_0), \tag{41}$$

which completes the proof. □

B.3 Proof of Theorem 2

In this section, we provide a complete proof of Theorem 2.

Lemma 3 (Information evolution for flow models)**.** *Let the flow model be defined by*

$$X_t = \alpha_t X_0 + \sigma_t X_1, \tag{42}$$

where $\alpha_t > 0$ and $\sigma_t \geq 0$ are differentiable functions of t, and $X_1 \sim \mathcal{N}(0, I)$ is independent of (X_0, Y). Consequently, for any conditioning variable Y,

$$\frac{\mathrm{d}}{\mathrm{d}t} I(X_t; Y) = -\frac{1}{2} g^2(t) J(X_t; Y) = -\left(\sigma_t \dot{\sigma}_t - \frac{\dot{\alpha}_t}{\alpha_t}\sigma_t^2\right) J(X_t; Y). \tag{43}$$

Proof. The reparameterized form of OU-SDE $\mathrm{d}X_t = f(t)X_t \mathrm{d}t + g(t)\mathrm{d}w_t$ is

$$X_t = e^{\int_0^t f(s)\mathrm{d}s} X_0 + \sqrt{\int_0^t \exp\left(2\int_s^t f(r)\mathrm{d}r\right) g^2(s)\mathrm{d}s}\, Z, \tag{44}$$

so for flow model

$$X_t = \alpha_t X_0 + \sigma_t X_1, \tag{45}$$

we can find its FPE by turning it into OU-SDE by defining

$$\alpha_t = e^{\int_0^t f(s)\mathrm{d}s}, \quad \sigma_t^2 = \int_0^t \exp\left(2\int_s^t f(r)\mathrm{d}r\right) g^2(s)\mathrm{d}s = \alpha_t^2 \int_0^t \frac{g^2(s)}{\alpha_s^2}\mathrm{d}s, \tag{46}$$

the solution can be easily found:

$$f(t) = \frac{\dot{\alpha}_t}{\alpha_t}, \quad g^2(t) = 2\sigma_t\dot{\sigma}_t - \frac{2\dot{\alpha}_t\sigma_t^2}{\alpha_t}, \tag{47}$$

so using Lemma 2,

$$\frac{\mathrm{d}}{\mathrm{d}t} I(X_t; Y) = -\frac{1}{2}\left(2\sigma_t\dot{\sigma}_t - \frac{2\dot{\alpha}_t\sigma_t^2}{\alpha_t}\right) J(X_t; Y) \tag{48}$$

$$= -\left(\sigma_t\dot{\sigma}_t - \frac{\dot{\alpha}_t\sigma_t^2}{\alpha_t}\right) J(X_t; Y), \tag{49}$$

□

Proof of Theorem 2. Thus just like the proof of Theorem 1, let $L(t)$ denote the information bottleneck objective defined in equation 16. If t^* is an interior minimizer of $L(t)$, it must satisfy the first-order optimality condition

$$\frac{\mathrm{d}L(t^*)}{\mathrm{d}t^*} = 0. \tag{50}$$

Applying Lemma 3 to both mutual information terms appearing in $L(t)$, we obtain

$$\beta J(X_{t^*}; C) = J(X_{t^*}; X_0), \tag{51}$$

which completes the proof. □

C Estimation of Score on Rectified Flow

C.1 Asymptotic Consistency Near the Gaussian Endpoint

The score–velocity conversion used in flow matching [34] is exact for stochastic Gaussian interpolation paths with independent endpoints. In particular, for independent (X_0, X_1), one has

$$\nabla \log p_t(x) = \frac{\frac{\alpha_t}{\dot{\alpha}_t} v_t(x) - x}{\sigma_t^2 - \frac{\alpha_t}{\dot{\alpha}_t}\dot{\sigma}_t\sigma_t}.$$

For the rectified flow parameterization [31], this identity reduces to the surrogate score

$$\tilde{s}_t(x) = \frac{(t-1)v_t(x) - x}{t}.$$

For deterministic couplings such as n-rectified flow with $n \geq 2$, the above conversion is in general no longer exact, since the information-theoretic assumptions underlying Theorem 2 do not apply. Therefore, in this setting, $\tilde{s}_t$ should be understood as a practical approximation induced by the learned velocity field, rather than the true score $s_t = \nabla \log p_t$.

Nevertheless, this approximation becomes asymptotically accurate near the high-noise Gaussian endpoint. Indeed, as $t \to 1$,

$$\tilde{s}_t(x) = \frac{(t-1)v_t(x) - x}{t} \longrightarrow -x,$$

provided that $v_t(x)$ remains bounded near $t = 1$. On the other hand, if the endpoint distribution satisfies $p_1 = \mathcal{N}(0, I)$, then its true score is exactly

$$s_1(x) = \nabla \log p_1(x) = -x.$$

Hence,

$$\tilde{s}_t(x) \to s_1(x) \qquad \text{as } t \to 1.$$

This implies that $\tilde{s}_t(x)$ is an asymptotically accurate proxy near the Gaussian endpoint.

C.2 Consistency of Score Differences via Local PDE Dynamics

We next compare the evolution of the two score functions. The surrogate score satisfies

$$\partial_t \tilde{s}_t = \frac{(1-t)v_t + x_t}{t^2} + \frac{v_t + (t-1)\frac{\partial v_t}{\partial t}}{t} = \frac{v_t + x_t}{t^2} + \frac{t-1}{t}\frac{\partial v_t}{\partial t},$$

and when $t \to 1$, it becomes

$$\partial_t \tilde{s}_t \approx x_t + v_t. \tag{52}$$

Meanwhile, the true score satisfies

$$\partial_t s_t = -\nabla(v_t \cdot s_t) - \nabla(\nabla \cdot v_t) = -J(v_t)s_t - J(s_t)v_t - \nabla(\nabla \cdot v_t).$$

Near the Gaussian endpoint, we use the local approximation $s_t(x_t) \approx -x_t$, so that $J(s_t) \approx -I$. If, in addition, v_t is locally affine in x near $t = 1$, then $\nabla(\nabla \cdot v_t) \approx 0$, which yields

$$\partial_t s_t \approx J(v_t)x_t + v_t.$$

Subtracting the surrogate dynamics equation 52 from the true score evolution, the local discrepancy in the time derivative is given by

$$\partial_t(s_t - \tilde{s}_t) \approx (J(v_t) - I)x_t. \tag{53}$$

The equation above is fit for any conditional velocity. And as our target is to find out whether

$$\|s_t(x_t|c)\|^2 - \|s_t(x_t|\varnothing)\|^2 \approx \|\tilde{s}_t(x_t|c)\|^2 - \|\tilde{s}_t(x_t|\varnothing)\|^2.$$

To rigorously justify the approximation, we examine the evolution between the conditional and unconditional score differences. Since both the true scores and their surrogates converge to the same value (specifically $-x_t$) as $t \to 1$, the critical factor determining the approximation quality is the evolution of their difference. We thus focus on the quantity:

$$\Delta_t = (s_t(x_t|c) - s_t(x_t|\varnothing)) - (\tilde{s}_t(x_t|c) - \tilde{s}_t(x_t|\varnothing)).$$

Leveraging the local dynamics derived earlier, the time derivative of this discrepancy is governed by the Jacobian term:

$$\partial_t \Delta_t = \partial_t\big((s_t - s_{t,c}) - (\tilde{s}_t - \tilde{s}_{t,c})\big) \approx \big(J(v_t) - J(v_{t,c})\big)x_t,$$
$$\Delta_1 = (s_1(x_t|c) - s_1(x_t|\varnothing)) - (\tilde{s}_1(x_t|c) - \tilde{s}_1(x_t|\varnothing)) = 0.$$

As $t \to 1$, the input data is dominated by noise, effectively masking the conditional information and making velocity fields v_t and $v_{t,c}$ become increasingly similar near the Gaussian endpoint [13]. Thus we assert that:

$$J(v_t) - J(v_{t,c}) \to 0 \quad \text{as} \quad t \to 1.$$

Consequently, the discrepancy $\Delta(t)$ vanishes asymptotically. This implies that the rate of change of the score difference $\|s_{t,c}\|^2 - \|s_t\|^2$ in equation 27 is preserved under the surrogate approximation$\|\tilde{s}_{t,c}\|^2 - \|\tilde{s}_t\|^2$. Therefore, the proposed surrogate model asymptotically captures the correct guidance signal, validating the intended approximation.

D Implementation Details and Code

This appendix provides the detailed implementation blueprint of our proposed method. As our method is a *training-free* and *plug-and-play* algorithm, the core implementation consists of a lightweight wrapper function that modifies the sampling process. Below is the code for our key procedures for SDE samplers of flow matching method. ODE samplers and DDPM/DDIM method is similar to this.

```
def get_score(vt, xt, t, path_type="linear"):
    t = expand_t_like_x(t, xt)
    if path_type == "linear":
        alpha_t, d_alpha_t = 1 - t, torch.ones_like(xt, device=xt.
    device) * -1
        sigma_t, d_sigma_t = t, torch.ones_like(xt, device=xt.device)
    elif path_type == "cosine":
        alpha_t = torch.cos(t * np.pi / 2)
        sigma_t = torch.sin(t * np.pi / 2)
        d_alpha_t = -np.pi / 2 * torch.sin(t * np.pi / 2)
        d_sigma_t =  np.pi / 2 * torch.cos(t * np.pi / 2)
    else:
        raise NotImplementedError
    mean = xt
    reverse_alpha_ratio = alpha_t / d_alpha_t
    var = sigma_t - reverse_alpha_ratio * d_sigma_t
    score = (reverse_alpha_ratio * vt - mean)/ var
    return score

def euler_maruyama_sampler_plus(
        model,
        latents,
        y,
        num_steps=20,
        heun=False,  # not used, just for compatability
        cfg_scale=1.0,
        guidance_low=0.0,
        guidance_high=1.0,
        path_type="linear",
):
    thrd=-5.9
    if True:
        y_null = torch.tensor([1000] * y.size(0), device=y.device)
    _dtype = latents.dtype
    t_steps = torch.linspace(1., 0.04, num_steps, dtype=torch.float64)
    t_steps = torch.cat([t_steps, torch.tensor([0.], dtype=torch.
    float64)])
    x_next = latents.to(torch.float64)
    device = x_next.device
    flag = False
    with torch.no_grad():
        for i, (t_cur, t_next) in enumerate(zip(t_steps[:-2], t_steps
    [1:-1])):
            dt = t_next - t_cur
            x_cur = x_next
            model_input = torch.cat([x_cur] * 2, dim=0)
```

```
            y_cur = torch.cat([y, y_null], dim=0)
            kwargs = dict(y=y_cur)
            time_input = torch.ones(model_input.size(0)).to(device=
    device, dtype=torch.float64) * t_cur
            diffusion = compute_diffusion(t_cur)
            eps_i = torch.randn_like(x_cur).to(device)
            deps = eps_i * torch.sqrt(torch.abs(dt))
            v_cur = model(
                model_input.to(dtype=_dtype), time_input.to(dtype=
    _dtype), **kwargs
            )[0].to(torch.float64)
            s_cur = get_score_from_velocity(v_cur, model_input,
    time_input, path_type=path_type)
            d_cur = v_cur - 0.5 * diffusion * s_cur
            if t_cur <= guidance_high:
                flag=True
            if not flag:
                score = get_score(v_cur, model_input, time_input,
    path_type=path_type)
                s_cur_cond, s_cur_uncond = score.chunk(2)
                cond_sq = s_cur_cond.pow(2).mean(dim=[1, 2, 3])  # [B]
                uncond_sq = s_cur_uncond.pow(2).mean(dim=[1, 2, 3])
                E_cond = cond_sq.mean()
                E_uncond = uncond_sq.mean()
                beta = max(float(E_cond) - float(E_uncond), 0) + 1e-12
                log_beta = np.log(beta)
                if log_beta > thrd:
                    flag = True
            d_cur_cond, d_cur_uncond = d_cur.chunk(2)
            if flag:
                cfg_scale_cp = cfg_scale
            else:
                cfg_scale_cp = 1.0
            d_cur = d_cur_uncond + cfg_scale_cp * (d_cur_cond -
    d_cur_uncond)
            x_next = x_cur + d_cur * dt + torch.sqrt(diffusion) * deps
    t_cur, t_next = t_steps[-2], t_steps[-1]
    dt = t_next - t_cur
    x_cur = x_next
    if True:
        model_input = torch.cat([x_cur] * 2, dim=0)
        y_cur = torch.cat([y, y_null], dim=0)
    kwargs = dict(y=y_cur)
    time_input = torch.ones(model_input.size(0)).to(
        device=device, dtype=torch.float64
    ) * t_cur
    v_cur = model(
        model_input.to(dtype=_dtype), time_input.to(dtype=_dtype), **
    kwargs
    )[0].to(torch.float64)
    s_cur = get_score_from_velocity(v_cur, model_input, time_input,
    path_type=path_type)
    diffusion = compute_diffusion(t_cur)
    d_cur = v_cur - 0.5 * diffusion * s_cur
    if not flag:
        score = get_score(v_cur, model_input, time_input, path_type=
    path_type)
        s_cur_cond, s_cur_uncond = score.chunk(2)
        cond_sq = s_cur_cond.pow(2).mean(dim=[1, 2, 3])  # [B]
        uncond_sq = s_cur_uncond.pow(2).mean(dim=[1, 2, 3])
        E_cond = cond_sq.mean()
        E_uncond = uncond_sq.mean()
        beta = max(float(E_cond) - float(E_uncond), 0) + 1e-12
        log_beta = np.log(beta)
        if log_beta > thrd:
```

```
            flag = True
    d_cur_cond, d_cur_uncond = d_cur.chunk(2)
    if flag:
        cfg_scale_cp = cfg_scale
    else:
        cfg_scale_cp = 1.0
    d_cur = d_cur_uncond + cfg_scale_cp * (d_cur_cond - d_cur_uncond)
    mean_x = x_cur + dt * d_cur
    return mean_x
```

E Additional Analysis and Experiments

E.1 Properties of Stopping Time

We collected a large-scale dataset of stopping times $\{\tau_i\}_{i=1}^{N}$ for CFG activation on SiT-XL/2 ($\log \beta^* = 5.9$). To characterize the stochastic nature of this stopping time, we fitted the empirical distribution of τ against several candidate probability distributions, including Gamma, Log-Normal, Exponential, and Weibull distributions.

Descriptive statistics of the collected dataset reveal a mean stopping time of $\bar{\tau} \approx 0.680$ with a standard deviation of $\sigma_\tau \approx 0.050$. This indicates that while the stopping time exhibits a clear central tendency, it retains a moderate degree of variability across different sampling trajectories.

The goodness of fit was evaluated using the log-likelihood metric. We find out that the Weibull distribution achieves the highest log-likelihood ($\mathcal{L} \approx 713.24$), significantly outperforming the Exponential ($\mathcal{L} \approx -268.89$) and Gamma ($\mathcal{L} \approx 675.64$) baselines. This suggests that the optimal stopping time τ in our diffusion sampling process follows a Weibull distribution with shape parameter $k \approx 16.78$ and scale parameter $\lambda \approx 0.70$.

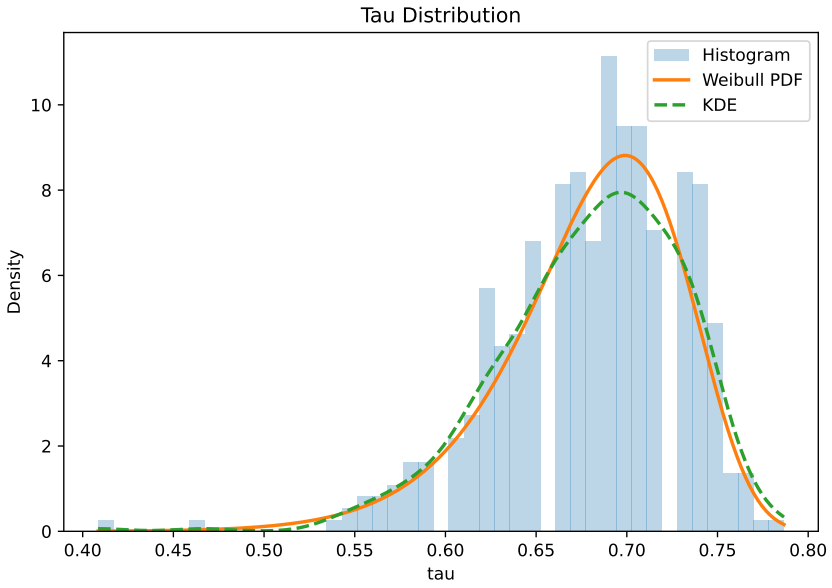


Figure 4: Probability density function (PDF) of the stopping time τ. The histogram represents the empirical distribution of sampled τ values. The solid curve is the fitted Weibull PDF, and the dashed curve is the Kernel Density Estimation (KDE) curve.

The visual comparison presented in Figure 4 further corroborates the statistical findings. The histogram, which depicts the empirical frequency of τ, aligns closely with both the Weibull PDF (solid line) and the KDE curve (dashed line).

While the mean stopping time $\bar{\tau}$ closely approximates the heuristic interval boundary $t_h = 0.7$ used in prior art (e.g., Interval Guidance), the distribution reveals a crucial nuance. The survival function

of the Weibull fit suggests that although the peak intensity occurs around 0.7, there exists a "long tail" of activation events extending down to 0.4.

Table 7: Comparison of fixed and stochastic stopping strategies on ImageNet 256. We evaluate a deterministic variant that uses the empirical mean stopping time ($t_h = \bar{\tau} \approx 0.68$) against Interval Guidance and our proposed stochastic stopping method. While the mean-based threshold improves over the baseline, it remains inferior to the stochastic strategy, highlighting the importance of modeling the variability of stopping time.

ImageNet $B = 48$			
Model-256, 50k samples, 250 inference timesteps	FID↓	IS↑	sFID↓
SiT-XL/2 (REPA)($\omega = 1.35$, SDE)	1.80	284.0	**4.5**
SiT-XL/2 (REPA, Interval) ($\omega = 1.8, t_l = 0, t_h = 0.7$)	<u>1.42</u>	305.7	<u>4.7</u>
SiT-XL/2 (REPA) ($\omega = 1.8, t_l = 0, t_h = 0.68$)	1.43	299.3	<u>4.7</u>
SiT-XL/2 (REPA) ($\omega = 1.9, t_l = 0, t_h = 0.68$)	<u>1.42</u>	**310.4**	<u>4.7</u>
SiT-XL/2 (REPA) + Ours ($\omega = 1.9, \log \beta^* = 5.9$)	**1.41**	<u>306.6</u>	<u>4.7</u>

We additionally conduct an ablation by replacing the random stopping time with a fixed threshold set to the empirical mean $\bar{\tau} \approx 0.68$. The results are summarized in Table 7. Compared to the baseline Interval Guidance ($t_h = 0.7$), using the mean stopping time yields consistent improvements in IS, indicating that the statistical estimation of τ is indeed informative. However, this deterministic variant still underperforms our proposed stochastic stopping strategy, which achieves the best overall FID.

This observation highlights that while the average stopping time captures the central tendency of activation, it fails to account for the intrinsic variability revealed by the Weibull distribution. In contrast, our method leverages this stochasticity to adaptively determine activation timing on a per-sample basis, thereby better aligning with the heterogeneous emergence of conditional structures during sampling. This result further supports the necessity and effectiveness of modeling τ as a random variable rather than a fixed constant.

E.2 Ablation on Parameter $\log \beta^*$

Table 8: **Ablation comparison on $\log \beta^*$.** Comparison of different evaluation metrics on Class-Conditional ImageNet datasets with different $\log \beta^*$.

ImageNet(256×256), 10k samples, 50 inference timesteps		
SiT-XL/2 (REPA), SDE	FID↓	IS ↑
Baseline($\omega = 1.8, t_h = 0.7, t_l = 0, B = 32$)	5.09	287. 4
Ours($\omega = 1.9, \log \beta^* = 5.5$)	5.27	282.3
Ours($\omega = 1.9, \log \beta^* = 5.7$)	5.07	291.1
Ours($\omega = 1.9, \log \beta^* = 5.9$)	4.92	299.7
Ours($\omega = 1.9, \log \beta^* = 6.1$)	4.85	307.5
Ours($\omega = 1.9, \log \beta^* = 6.3$)	4.82	314.2
Ours($\omega = 1.9, \log \beta^* = 6.5$)	4.83	321.6
Ours($\omega = 1.9, \log \beta^* = 6.7$)	4.83	329.5
Ours($\omega = 1.9, \log \beta^* = 6.9$)	4.89	335.4

We study the effect of the activation threshold $\log \beta^*$ in our dynamic CFG strategy on class-conditional ImageNet 256×256 generation. As shown in Table 8, we vary $\log \beta^*$ while keeping other hyper-parameters fixed. We observe a clear non-monotonic trend. When $\log \beta^*$ is relatively small (e.g., 5.5–5.7), the performance is slightly worse than or comparable to the baseline, indicating that activating guidance too early introduces noisy or less informative conditional signals, which degrades sample

quality. As $\log \beta^*$ increases (5.9–6.3), both FID and IS improve consistently, suggesting that delaying guidance allows the diffusion trajectory to first form a stable coarse structure before incorporating conditional information. This trend is also visually summarized in Figure 5, where the trajectory of points demonstrates the trade-off between quality and diversity, with the peak performance occurring around $\log \beta^* = 6.3$ before fidelity degrades at higher thresholds.

The best performance is achieved around $\log \beta^* \in [6.3, 6.7]$, where FID reaches 4.82 and IS improves significantly over the baseline. This aligns with our theoretical analysis that conditional information becomes more informative after a critical stage characterized by the Fisher-information balance. However, when $\log \beta^*$ becomes too large (e.g., 6.9), we observe a slight degradation in FID despite continued improvement in IS. This suggests that overly delayed activation reduces the effective window during which guidance can refine samples, leading to suboptimal fidelity.

Overall, these results validate our hypothesis that guidance should be activated neither too early nor too late, and that an appropriate threshold plays a crucial role in achieving optimal performance.

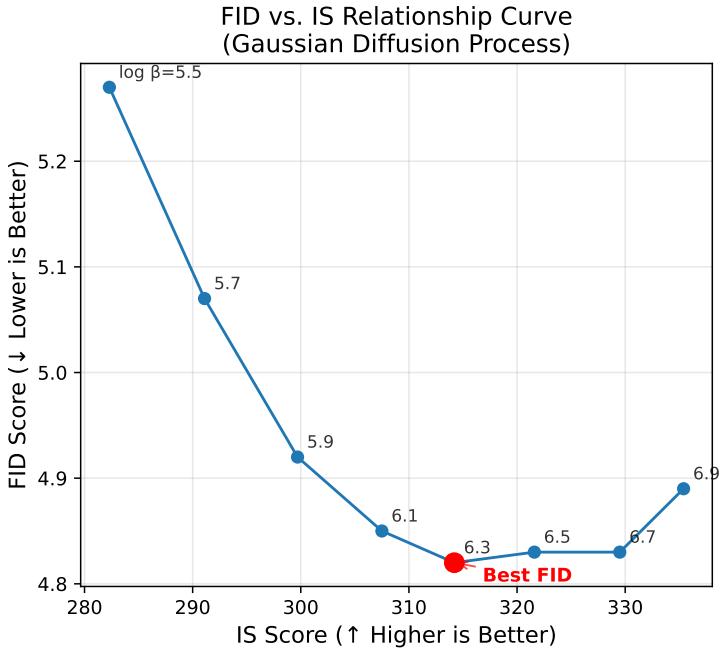
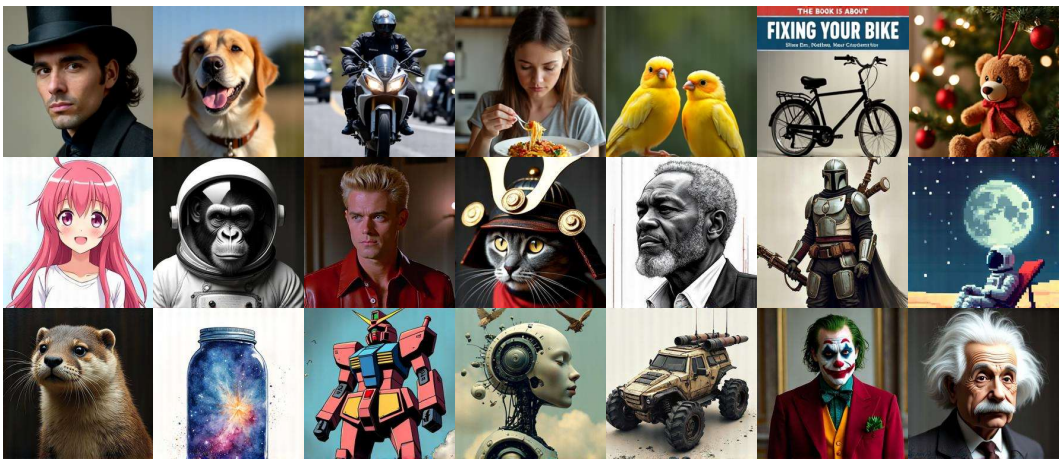


Figure 5: FID-IS trajectory under varying $\log \beta^*$ based on Table 8. Each point corresponds to a specific $\log \beta^*$ value along the trajectory. As $\log \beta^*$ increases, IS improves while FID exhibits a U-shaped trend, reflecting the quality-diversity trade-off in generation.

E.3 Additional visual results

Figure 6 shows additional visual results using our method on Flux and SiT models.



(a) Images generated by the FLUX with our method.



(b) Images generated by the SiT-XL/2 (REPA) model with our method.

Figure 6: Additional visual results on different models.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See limitations in Section 3 and future work in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 5 and other settings are just same as the baseline.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our proposed method is a training-free, plug-and-play algorithm that modifies the inference step of standard diffusion models. The core logic consists of a lightweight statistical estimation and a dynamic thresholding mechanism, which is simple to implement. To ensure reproducibility, we have provided the Python code in Appendix D. Since our method acts as a wrapper around existing samplers, no complex retraining code is required.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5 and other settings are just same as the baseline.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We analyze the distribution of τ and validate its effectiveness in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See settings in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.