UNCERTAINTY-ADAPTIVE FEEDBACK GUIDANCE FOR IMPROVED IMAGE GENERATION WITH DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

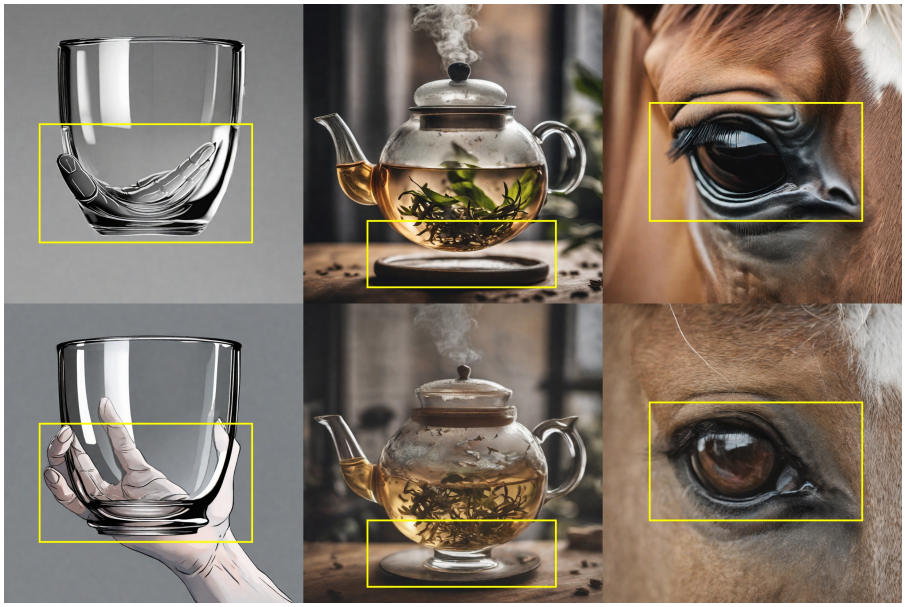


Figure 1: Qualitative comparison of text-to-image results with Stable Diffusion XL (SDXL). The top row shows SDXL with classifier-free guidance (CFG); the bottom row shows our method. Yellow boxes mark corresponding regions for comparison: (1) finger–glass geometry, (2) the teapot–saucer contact, and (3) the structure around the horse's eye. Prompts: (i) A hand holding a glass cup, detailed fingers, transparent reflections. (ii) A transparent glass teapot with tea leaves swirling inside, steam, macro photo. (iii) Extreme close-up of a horse's eye reflecting a field, detailed eyelashes.

ABSTRACT

The success of modern diffusion models heavily depends on the guidance mechanisms that align generated images with conditions. However, the widely adopted classifier-free guidance (CFG) applies a global guidance scale across the entire image. Hence, it can overlook the complex dynamics of the generative process, and may result in misaligned features or visual artifacts. To overcome this limitation, we introduce a novel guidance mechanism named Uncertainty-adaptive Feedback Guidance (UFG) that leverages pixel-wise uncertainty as an adaptive scale. By utilizing uncertainty estimates, our method adaptively determines the pixel-wise guidance scale by minimizing the one-step marginal variance at each denoising step. This guidance affects the next-step uncertainty estimation, and hence establishes a feedback loop. Through empirical evaluations across various image generation tasks, we demonstrate that scaling guidance based on pixel-wise uncertainty enhances overall generation quality and effectively mitigates visual artifacts. Consequently, UFG achieves strong performance compared to standard CFG, recent guidance variants, and existing uncertainty-aware baselines.

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

1 INTRODUCTION

Diffusion models have attained significant successes as a tool for image generation, finding widespread application across a broad spectrum of fields [1, 10, 15, 29, 41]. In particular, conditional diffusion models that generate images given class labels or text prompts have become especially prominent in real-world implementations. In this conditional generation process, guidance mechanisms are crucial for steering the image generation toward the desired conditions, thereby ensuring that the generated samples align well with the given prompts [10, 14, 24, 29].

One of the most widely adopted guidance mechanisms for conditional generation is classifier-free guidance (CFG) [14], which effectively amplifies the conditional signal during sampling. The standard CFG operates with a global guidance scale across the entire image. While this straightforward approach is effective at improving overall generation quality, it can overlook the complex dynamics of the generative process. During denoising, different regions of an image naturally exhibit varying degrees of uncertainty, which can often lead to localized visual artifacts [6, 20, 40, 44]. Consequently, relying on a global guidance scale would limit the potential to provide more fine-grained, optimal guidance at the pixel level.

To better address varying natures of the generation process, recent research has explored methods to estimate spatial uncertainty at the sample level [8, 18]. Specifically, Kou et al. [18] introduced a method for estimating the pixel-wise uncertainty of diffusion-generated images by utilizing Bayesian inference. Their empirical findings indicated that visual artifacts tend to manifest more frequently in images with high uncertainty. Consequently, they proposed to filter out images with high uncertainty to yield images with better quality. However, filtering out low-quality images after generation requires generating more images than necessary, and there is no guarantee of obtaining a high-quality image within a given time. De Vita & Belagiannis [8] advanced one step further by integrating aleatoric uncertainty into the guidance mechanism to improve the quality of generated images. However, their approach for uncertainty quantification is computationally demanding, which could limit the effectiveness of the proposed guidance and result in inferior quality of generated images compared to the standard guidance mechanism.

In this paper, we focus on developing a more sophisticated guidance mechanism that overcomes the limitations of the guidance with a global guidance scale. Specifically, we adopt the uncertainty estimation of Kou et al. [18] and propose a novel algorithm that incorporates pixel-wise uncertainty into the guided sampling process. Our method adaptively scales the guidance at each pixel based on its estimated uncertainty, replacing the conventional global guidance with a pixel-wise adaptive guidance. Since the guidance modifies the uncertainty estimate at the next step, this naturally forms a feedback loop that progressively stabilizes uncertain regions. Empirically, we validate our design of adaptive guidance and also investigate its characteristics compared to the global guidance scale. Through numerical studies, we demonstrate that our proposed guidance effectively stabilizes pixel-wise uncertainty during generation, naturally mitigating localized visual artifacts to yield high-fidelity images in various image generation tasks.

Our main contributions are summarized as follows:

- We propose a novel guidance mechanism that replaces the standard global guidance scale with a pixel-wise adaptive scale derived from variance minimization, leveraging Bayesian variance estimation within a feedback loop across sampling steps.
- Through numerical studies, we show that UFG achieves strong generation quality compared to standard CFG, recent guidance variants, and existing uncertainty-aware baselines in various image generation tasks. We further demonstrate that our uncertainty guidance stabilizes pixel-wise uncertainty, especially in highly uncertain regions.

2 PRELIMINARIES

2.1 CONDITIONAL DIFFUSION MODELS AND CLASSIFIER-FREE GUIDANCE

Assume that we have an image $\boldsymbol{x}_0 \in \mathbb{R}^{h\times w\times c} =: \mathbb{R}^d$, with height h, width w and channels c, along with a condition $\boldsymbol{c}$ (e.g., a class label or text prompt). Diffusion models [15, 34, 36] utilize a forward process to gradually transform $\boldsymbol{x}_0$ into Gaussian noise, by injecting noise. Then, by using the corresponding reverse process conditioned on $\boldsymbol{c}$, they can reconstruct images from the

noise, where each transition step is modeled as Gaussian. Following Kingma et al. [17], we can generally interpret these diffusion processes as follows. Let $\boldsymbol{x}_t$ for $t \in [0, T]$ be the latent variable on timestep t, where it follows the distribution $q_t(\boldsymbol{x}_t)$. We consider a forward process $q(\cdot)$, such that $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\alpha_t \boldsymbol{x}_0, \sigma_t^2 \mathbf{I})$, i.e. $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$, with $\alpha_t, \sigma_t \in \mathbb{R}^+$. The noise schedule is designed such that the signal-to-noise ratio α_t^2/σ_t^2 monotonically decreases to near zero as $t \to T$, ensuring that $q_T(\boldsymbol{x}_T) \approx \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$. The reverse process is learned by a conditional noise prediction model $\boldsymbol{\epsilon}_\theta$ with parameters θ, which minimizes the following loss function:

$$\mathbb{E}_{\boldsymbol{x}_0 \sim q_0(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), t \sim \mathrm{Unif}(0,T)} \left[\lambda(t) \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}, t, \boldsymbol{c})\|_2^2 \right], \tag{1}$$

where $\lambda(t)$ is a certain weight function depending on t. The corresponding image generation process is derived from the reverse-time SDE or probability flow ODE [2, 37], where the reverse-time SDE is given as follows: $d\boldsymbol{x}_t = \left[f(t)\boldsymbol{x}_t + \frac{g(t)^2}{\sigma_t} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) \right] dt + g(t) d\bar{\boldsymbol{w}}_t$, where $f(t) = \frac{d \log \alpha_t}{dt}$, $g(t)^2 = \frac{d\sigma_t^2}{dt} - 2\frac{d \log \alpha_t}{dt}\sigma_t^2$ and $\bar{\boldsymbol{w}}_t$ denotes the standard Wiener process in reverse time. We simulate the SDE / ODE as discretized samplers. One example of the sampler is to apply Euler-Maruyama (EM) discretization, which yields:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t - \left(f(t)\boldsymbol{x}_t + \frac{g(t)^2}{\sigma_t} \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) \right) + g(t)\boldsymbol{z}, \ \boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{2}$$

For simplicity, we use this EM sampler in the following. There exist other samplers [3, 23, 36] to execute the generation process.

Classifier-free guidance (CFG) [14] is a key technique to enhance generation quality of conditional diffusion models. Unlike classifier guidance [10], which requires an auxiliary classifier to guide the sampling process, CFG jointly trains a single model for both conditional and unconditional distributions. During training, the conditioning $\boldsymbol{c}$ is randomly replaced with a null token $\varnothing$ with a fixed probability. During sampling, the predicted noise is shifted away from the unconditional estimate towards the conditional one:

$$\boldsymbol{\epsilon}_\theta^{\mathrm{cfg}}(\boldsymbol{x}_t, t, \boldsymbol{c}) = (1 + \omega)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) - \omega \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \varnothing), \tag{3}$$

where $\omega \geq 0$ is the guidance scale that controls the trade-off between generation quality (fidelity) and diversity. A higher ω typically improves generation quality and alignment with a given condition at the cost of diversity.

2.2 Uncertainty quantification for diffusion models

While population-level metrics such as FID [13] and IS [32] evaluate the overall quality of generated image distributions, understanding the pixel-wise behavior of the generation process remains underexplored. Recent approaches address this issue by estimating uncertainty during the sampling process, which provides spatial information about where the model is less confident.

We begin by reviewing the pixel-wise uncertainty estimate proposed by BayesDiff [18]. The main goal of their work is to estimate the marginal variance $\mathrm{Var}(\boldsymbol{x}_t)$ for each timestep t. To do so, they consider the posterior distribution of the parameter θ of the noise prediction model, $p(\theta|\mathcal{D})$. Here, they collect the observation set $\mathcal{D}$ by re-performing the forward process on the training data. By considering this Bayesian inference, they can inject epistemic uncertainty into diffusion models, differentiating the pixel-wise variance estimates. Using technical details, they derive the following predictive distribution of $\boldsymbol{\epsilon}_t$ approximated with Gaussian:

$$p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D}) \approx \mathcal{N}(\mu_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}), \Sigma_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})). \tag{4}$$

Then, BayesDiff uses a random sample $\boldsymbol{\epsilon}_t \sim p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D})$ at each time point to construct a random sampling path where $\mathrm{Var}(\boldsymbol{x}_0)$ given $\boldsymbol{x}_T$ can be calculated. Finally, they propose to use $\mathrm{Var}(\boldsymbol{x}_0)$ as a sample quality. To yield better images, the authors suggest to filter out images with higher variances, i.e. $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_0))$.

Unlike BayesDiff, De Vita & Belagiannis [8] propose a pixel-wise aleatoric uncertainty measure based on the sensitivity of $\boldsymbol{\epsilon}_\theta$ to input perturbations at each denoising step, which they connect to the curvature of the noising distribution. Then the authors propose a new guidance using this uncertainty measure, to actively guide and yield better images. We call their method AU (aleatoric uncertainty) throughout the paper.

On the other hand, Jazbec et al. [16] move away from pixel-space and propose a sample-level uncertainty measure by computing the variability of generated images in a pretrained semantic feature space (e.g. CLIP) under multiple posterior parameter samples. The goal of Jazbec et al. [16] is to measure the quality of an individual sample in a post-hoc fashion.

3 UNCERTAINTY-ADAPTIVE FEEDBACK GUIDANCE (UFG)

3.1 MOTIVATION: FROM DIAGNOSIS TO GUIDANCE

BayesDiff empirically finds that an image with high uncertainty (measured as the sum of pixel-wise uncertainty estimates) tends to have low quality, which implies that it contains artifacts or is misaligned with a given condition. This suggests that reducing pixel-wise variance during sampling could improve the generation quality. Based on this observation, we formulate the choice of pixel-wise guidance scales as a variance minimization problem at each denoising step. The resulting optimal scale allocates stronger guidance to pixels with higher uncertainty to reduce the overall variance most effectively, thus suppressing uncertain regions. As a practical solution, we incorporate CFG into BayesDiff, to enable the adaptive guidance across pixels.

Tab. 1 summarizes how our method is positioned, compared to standard diffusion and BayesDiff. Our method actively reflects uncertainty in the guidance mechanism. Also, we consider both stochastic and deterministic noise predictions, which we detail in Secs. 3.3 and 3.4.

Table 1: Abstract comparison of standard Diffusion, BayesDiff and our method.

Methods	Noise prediction	Role of uncertainty
Diffusion	point estimate	None
BayesDiff [18]	distribution	Diagnosis
Ours	point estimate & distribution (CFG with adaptive scale)	Guidance

3.2 BAYESDIFF RECAP

Let the observation set be $\mathcal{D} = \{(\boldsymbol{\epsilon}^{(s)}_{t^{(s)}}, \boldsymbol{x}^{(s)}_{t^{(s)}}, t^{(s)}, \boldsymbol{c}^{(s)})\}_{s=1}^{S}$, a set of S observations created by re-performing the forward process. Then, we can consider the posterior distribution of noise prediction model $\pi(\theta|\mathcal{D})$, and approximate it as a Gaussian distribution $\mathcal{N}(\theta; \hat{\theta}^{\mathrm{MAP}}, H^{-1})$ by Laplace approximation. This yields the Gaussian predictive distribution as follows:

$$p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D}) \approx \mathcal{N}(\mu_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}), \Sigma_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})) \tag{5}$$

They [18] utilize a random vector $\boldsymbol{\epsilon}_t$ that follows $p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D})$ to replace the deterministic estimation on noise prediction $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$. That is, the sampling path from BayesDiff follows (with EM sampler):

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t - \left(f(t)\boldsymbol{x}_t + \frac{g(t)^2}{\sigma_t}\boldsymbol{\epsilon}_t \right) + g(t)\boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I}). \tag{6}$$

Upon Eq. (5), they calculate the distribution of $\boldsymbol{x}_{t-1}$ as well:

$$\boldsymbol{x}_{t-1} \sim \mathcal{N}(m_{t-1}, V_{t-1}) \tag{7}$$

The moments m_{t-1} and V_{t-1} can be calculated by using μ_{pred} and Σ_{pred} (see App. A.1 for detailed calculation). Finally, the diagonal elements of V_0 become the uncertainty estimate for each pixel.

We can also incorporate CFG into BayesDiff (the authors used the following formulation in their code implementation without any explanation in the manuscript). The idea is to consider a new random vector $\boldsymbol{\epsilon}'_t$ that follows a CFG-motivated distribution. Note that the predicted noise from CFG is given in Eq. (3). Similarly, one can consider the following two random vectors:

$$\boldsymbol{\epsilon}^c_t \sim p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D}) \approx \mathcal{N}(\mu^c, \Sigma^c) \tag{8}$$

$$\boldsymbol{\epsilon}^u_t \sim p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \varnothing, \mathcal{D}) \approx \mathcal{N}(\mu^u, \Sigma^u), \tag{9}$$

where $\mu^c = \mu_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}), \Sigma^c = \Sigma_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})$, and μ^u, Σ^u for the unconditional counterpart are defined similarly. Then we can consider a random vector $\boldsymbol{\epsilon}'_t = (1+\omega)\boldsymbol{\epsilon}^c_t - \omega\boldsymbol{\epsilon}^u_t$ as a linear

combination which follows the distribution: $\mathcal{N}(\mu'_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}), \Sigma'_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}))$, where $\mu'_{\text{pred}} = \mu^c + \omega(\mu^c - \mu^u), \Sigma'_{\text{pred}} = (1+\omega)^2\Sigma^c + \omega^2\Sigma^u$. Replacing $\boldsymbol{\epsilon}_t$ as $\boldsymbol{\epsilon}'_t$ in the sampling path of Eq. (6) allows the CFG incorporation. Note that assuming the two random noises are independent (which we call independent coupling) yields a variance-inflated joint distribution of the two marginal distributions, which helps mitigate the *underestimation of variance due to the Laplace approximation* in $\pi(\theta|\mathcal{D})$.

3.3 Main proposal: BayesDiff with adaptive guidance

As mentioned in Sec. 3.1, our main idea is to use different scales of guidance on each pixel to actively guide pixels to minimize the marginal variance at each denoising step. By doing so, the uncertainty across pixels is stabilized, which helps avoid artifacts and misalignment caused by highly uncertain pixels, and thus improves the quality of generated images.

Suppose that we have a pixel-wise uncertainty score vector γ_t, which will be specified later. We use γ_t to modulate the guidance scale across pixels.

As in the previous subsection, we consider two marginal predictive distributions:

$$\boldsymbol{\epsilon}_t^c \sim p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D}) \approx \mathcal{N}(\mu^c, \Sigma^c) \tag{10}$$

$$\boldsymbol{\epsilon}_t^u \sim p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \varnothing, \mathcal{D}) \approx \mathcal{N}(\mu^u, \Sigma^u) \tag{11}$$

Then, we consider the following linear combination:

$$\boldsymbol{\epsilon}_t^* = (1+\gamma_t) \odot \boldsymbol{\epsilon}_t^c - \gamma_t \odot \boldsymbol{\epsilon}_t^u \tag{12}$$

whose distribution is readily computed from the independent coupling:

$$p^*(\boldsymbol{\epsilon}_t^*|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D}) \approx \mathcal{N}(\mu^*_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}), \Sigma^*_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})) \tag{13}$$

where

$$\mu^*_{\text{pred}} = \mu^c + \gamma_t \odot (\mu^c - \mu^u), \tag{14}$$

$$\Sigma^*_{\text{pred}} = (1+\gamma_t)^{\circ 2} \odot \Sigma^c + (\gamma_t)^{\circ 2} \odot \Sigma^u. \tag{15}$$

Here, $x^{\circ 2} := xx^\top$ for a vector x. We name this p^* as the *guided predictive distribution*.

Now, we consider a random vector $\boldsymbol{\epsilon}_t^*$ that follows the guided predictive distribution p^*, and modify the sampling path of Eq. (6) by $\boldsymbol{\epsilon}_t^*$. Using the EM discretization[1], we obtain the following sampling path:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t - \left(f(t)\boldsymbol{x}_t + \frac{g(t)^2}{\sigma_t}\boldsymbol{\epsilon}_t^* \right) + g(t)\boldsymbol{z}, \;\; \boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{16}$$

Similar to BayesDiff, we can calculate the modified distribution of $\boldsymbol{x}_{t-1}$ as follows:

$$\boldsymbol{x}_{t-1} \sim \mathcal{N}(m^*_{t-1}, V^*_{t-1}) \tag{17}$$

where m^*_{t-1} and V^*_{t-1} are computed from quantities $\mu^c, \mu^u, \Sigma^c, \Sigma^u$ and γ_t. We defer the detailed computation to App. A.1 for better readability.

The modified uncertainty V^*_{t-1} is used in the next step to scale the guidance, and this procedure is repeated across the sampling steps. From this perspective, we name our proposed framework Uncertainty-adaptive Feedback Guidance (UFG). Fig. 2 compares the sampling processes of BayesDiff and UFG. We explain the choice of γ_t in the next subsection. We summarize the image generation process of UFG in Algorithm 1.

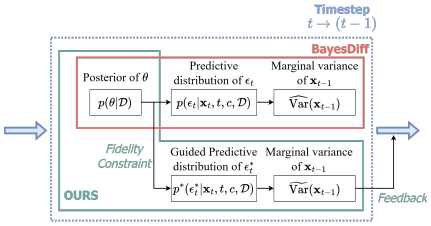


Figure 2: Overview of BayesDiff and UFG.

We allow a variant of UFG to simplify the sampling process. Specifically, we can simply use $\mu^*_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})$ as $\boldsymbol{\epsilon}_t^*$, the mean vector of the guided predictive distribution p^*. This enables us to

[1]We present this discretization for simplicity. Our methodology readily extends to other advanced samplers; see App. A.2 for details.

Algorithm 1: Uncertainty-adaptive Feedback Guidance (UFG)

Input : Starting point $\boldsymbol{x}_T$, Monte Carlo sample size M, pre-trained noise prediction model $\boldsymbol{\epsilon}_\theta$, condition $\boldsymbol{c}$, guidance strength λ.

Output : Image generation $\boldsymbol{x}_0$ and pixel-wise uncertainty $\mathrm{Var}(\boldsymbol{x}_0)$.

Construct the pixel-wise variance prediction function Σ_{pred} via Laplace approximation.

$\mathbb{E}(\boldsymbol{x}_T) \leftarrow \boldsymbol{x}_T$, $\mathrm{Var}(\boldsymbol{x}_T) \leftarrow \mathbf{0}$, $\mathrm{Cov}(\boldsymbol{x}_T, \boldsymbol{\epsilon}_T) \leftarrow \mathbf{0}$, $\gamma_T \leftarrow \mathbf{0}$

for $t = T \to 1$ **do**

  Compute μ^c, Σ^c from $p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D})$ and μ^u, Σ^u from $p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \varnothing, \mathcal{D})$.

  Compute guided predictive parameters:

$$\mu^*_{\mathrm{pred}} \leftarrow \mu^c + \gamma_t \odot (\mu^c - \mu^u)$$
$$\Sigma^*_{\mathrm{pred}} \leftarrow (1+\gamma_t)^{\circ 2} \odot \Sigma^c + (\gamma_t)^{\circ 2} \odot \Sigma^u$$

  Sample $\boldsymbol{\epsilon}^*_t \sim \mathcal{N}(\mu^*_{\mathrm{pred}}, \Sigma^*_{\mathrm{pred}})$.

  Obtain $\boldsymbol{x}_{t-1}$ via Eq. (16).

  Estimate $\mathbb{E}(\boldsymbol{x}_{t-1}) \leftarrow m^*_{t-1}$ and $\mathrm{Var}(\boldsymbol{x}_{t-1}) \leftarrow V^*_{t-1}$.

  Sample $\boldsymbol{x}^{(m)}_{t-1} \sim \mathcal{N}(m^*_{t-1}, V^*_{t-1})$, $m = 1, \ldots, M$.

  Estimate $\mathrm{Cov}(\boldsymbol{x}_{t-1}, \boldsymbol{\epsilon}_{t-1})$.

  Update γ_{t-1}.

follow the path of CFG point estimate, while variance estimation is still possible. We call this variant UFG-M (UFG with Mean vector). Hence, we can investigate the effect of pixel-wise adaptive guidance, by comparing CFG and UFG-M. The detailed algorithm is provided in App. A.3.

We also allow the practical acceleration to skip uncertainty quantification on a subset of the denoising steps, as BayesDiff did. In those steps, we adopt the deterministic sampling as standard diffusion models. Separately, the M Monte Carlo evaluations at each remaining step can be replaced by a first-order Taylor approximation of the moments, which needs one extra evaluation at the mean instead of M. We call this variant UFG-accel, and it works together with the step skipping above. UFG-accel enables pixel-wise uncertainty estimation at substantially reduced computational cost. See App. A.4 for details.

3.4 Choice of γ_t

From the empirical findings of BayesDiff which indicate that lower variance implies better individual quality, we aim to choose the pixel-wise guidance scale that directly minimizes the one-step marginal variance, as mentioned in Sec. 3.1. Specifically, we consider the following constrained problem for a timestep t :

$$\min_{s \in \mathbb{R}^d} \mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_{t-1}(s))) \text{ s.t. } \frac{1}{d}\sum_{i=1}^{d} s_i \le \lambda,\ s_i \ge 0\ \forall i. \tag{18}$$

Here, a vector s denotes the scale of guidance at time t, and $\boldsymbol{x}_{t-1}(s)$ denotes the next step guided by s.

Proposition 3.1. *The above optimization admits a unique solution given as:*

$$s^*_i = \frac{1}{2q_i}\left[-\ell_i - \nu\right]_+, \tag{19}$$

where q_i, ℓ_i are computable values by reusing the samples $\{\boldsymbol{x}^{(m)}_t\}_{m=1}^M$, $[u]_+ := \max\{u, 0\}$, and $\nu \ge 0$ is a threshold determined by λ.

App. C gives the definitions of the provided terms, the proof, the corresponding calculations, and a numerical study. We specifically choose γ_t as s^*; in other words, choosing γ_t as pixel-wise guidance minimizes the variance and hence yields better sample quality, sequentially across timesteps[2]. Also, note that we can reuse the CFG scale ω for λ to avoid additional tuning of λ, since the average of pixel-wise guidance scale are controlled by λ.

[2]A simplified rule scales γ_t by the pixel-wise standard deviation $\eta_t^{1/2} := \mathrm{diag}(V_t^*)^{1/2}$; see App. D.3.

4 EXPERIMENTS

We perform experiments across various settings to interpret the uncertainty estimates and evaluate the sampling quality under uncertainty guidance of our proposed method. We show that applying uncertainty-adaptive feedback guidance to each generated image improves the overall image quality across the population. To demonstrate this, we conduct experiments on (i) class-conditional image generation in Sec. 4.2 and (ii) text-to-image generation tasks in Sec. 4.3. We also provide a visual analysis of the proposed generation process in Sec. 4.4, to justify the quality improvement and artifact mitigation effect of pixel-wise adaptive guidance scale. We also perform image editing via DDIM inversion, with qualitative comparisons in App. E.4.

4.1 EXPERIMENTAL DETAILS

Here we provide some experimental details. For more detailed explanations, see App. B.1 for the definitions of evaluation metrics we use, and see App. B.2 for other details on models, datasets, hyperparameters, and other details. In the tables presenting the results, we **boldfaced** the best value, and underlined the second-best value.

Baselines We compare our proposed method UFG with CFG [14], two recent guidance variants CFG++ [7] and CADS [30], and two representative uncertainty-aware diffusion approaches: BayesDiff [18] and AU [8], under the same backbone models. For UFG, we report both the Monte Carlo version (UFG (MC)) and the Taylor-based version (UFG-accel) introduced in App. A.4. UFG-accel keeps the sampling cost close to CFG, about $1.2\times$, well below the overhead of UFG (MC). We report the full timing comparison in App. E.5. Since the original BayesDiff does not incorporate CFG, we augment it with CFG (BayesDiff+CFG) to ensure a fair comparison.

Evaluation metrics For class-conditional image generation, we evaluate generation quality by Fréchet Inception Distance (FID) [13], precision and recall [19]. For text-to-image generation, we evaluate generation quality by FID, CLIP-score [12] and VQAScore [22] to evaluate semantic similarity between an image and a text prompt. For class-conditional image generation, we evaluate all methods using the `torch-fidelity` library[25][3] and the ImageNet validation set to ensure a fair comparison. For text-to-image generation, we use `text2image-benchmark` library[26][4] on MSCOCO validation set.

4.2 CLASS-CONDITIONAL IMAGE GENERATION

Setup We conduct class-conditional generation experiments on ImageNet [9] at 256×256 and 512×512 resolutions. We perform Bayesian inference on a subset of ImageNet training data following the protocol of BayesDiff. We adopt U-ViT [4] as a backbone model and use 2nd-order DPM-Solver [23] for sampling. For each method, we synthesize 50,000 images and evaluate all samples.

Results Tab. 2 summarizes the results. BayesDiff+CFG/UFG and CFG/UFG-M share the same noise realization and differ only in the guidance scale, so the gap within each pair isolates the effect of the pixel-wise scale. At 256×256, UFG-accel gives the best FID among all methods and BayesDiff+CFG gives the best Precision, with our other variants close behind both and still ahead of CADS and CFG++. UFG-M is still best among our variants at 512×512, and its FID now edges past CFG++ as well, trailing only CADS overall. CADS and CFG++ lead on Recall throughout, with UFG-M closest to them among our variants. We read the FID gain over CFG as a sign that the variance-minimizing scale helps the model recover details it would otherwise skip in uncertain regions. Results on DiT-XL/2 are in App. E.1. Qualitative comparisons are shown in Figs. 11 and 12 of App. E.1.

4.3 TEXT-TO-IMAGE GENERATION

Setup We evaluate our method on text-to-image generation using MSCOCO-30K dataset [21]. We perform Bayesian inference on a subset of LAION-ART [33] as BayesDiff does. We adopt Stable

[3] `https://github.com/toshas/torch-fidelity`

[4] `https://github.com/boomb0om/text2image-benchmark`

Table 2: Comparison on quality measures of class-conditional image generation results.

Method	ImageNet 256×256			ImageNet 512×512		
	FID ↓	Recall ↑	Precision ↑	FID ↓	Recall ↑	Precision ↑
AU	5.7765	0.5795	0.6740	14.3728	0.4991	0.6282
BayesDiff+CFG	2.9359	0.5840	**0.7162**	4.8589	0.5150	0.6727
UFG (MC) ✓	2.9425	0.5849	0.7126	4.4817	0.5368	0.6938
UFG-accel ✓	**2.8577**	0.5907	0.7148	4.5532	0.5331	0.6905
CFG	3.2458	0.5990	0.7129	4.6782	0.5358	**0.7086**
UFG-M (MC) ✓	2.8773	0.6012	<u>0.7151</u>	<u>4.3728</u>	0.5372	<u>0.6951</u>
UFG-M (accel) ✓	<u>2.8652</u>	0.5949	0.7141	4.5532	0.5431	0.6905
CADS	2.9518	**0.6090**	0.6776	**4.3587**	**0.5497**	0.6818
CFG++	3.1537	<u>0.6024</u>	0.7067	4.4494	<u>0.5440</u>	0.6919

Diffusion XL (SDXL) [27] and Z-Image [43] as a backbone model. We use DDIM [35] for SDXL and a flow-matching Euler sampler with 50 steps for Z-Image. Results on Stable Diffusion v1.5 [29] are provided in App. E.2.

Results We provide evaluation on T2I generation in Tab. 3. On Z-Image, UFG-accel gives the best FID, with our other variants close behind and still ahead of CADS and CFG++. UFG also gives the best CLIP-score, with our other variants close behind. VQAScore differences across methods stay small. SDXL looks different, with CFG++ taking the best FID and UFG-M second, but our variants still take the best CLIP-score and VQAScore overall. CLIP-score overlooks fine-grained visual detail [42], so the two metrics are best read together. We also demonstrate the enhanced sample-wise fidelity through qualitative examples in Figs. 13 to 18 in App. E.2, including examples using Z-Image, SD1.5, and SDXL.

Table 3: Comparison on quality measures of T2I results.

Method	Z-Image			SDXL		
	FID ↓	CLIP-score ↑	VQAScore ↑	FID ↓	CLIP-score ↑	VQAScore ↑
AU	14.1935	0.3180	<u>0.9051</u>	14.1278	0.3196	0.8720
BayesDiff+CFG	13.8128	0.3083	0.9047	14.5777	0.3201	0.8738
UFG (MC) ✓	13.6256	**0.3192**	0.9048	14.1121	**0.3204**	**0.8749**
UFG-accel ✓	**12.9434**	<u>0.3185</u>	**0.9056**	14.1417	<u>0.3203</u>	0.8742
CFG	14.2414	0.3080	0.8956	14.1478	0.3197	0.8720
UFG-M (MC) ✓	<u>13.2179</u>	0.3181	<u>0.9051</u>	14.1239	**0.3204**	0.8740
UFG-M (accel) ✓	13.4512	0.3181	0.9049	<u>14.0395</u>	<u>0.3203</u>	<u>0.8746</u>
CADS	13.8491	0.3177	0.9043	14.9043	0.3163	0.8525
CFG++	14.0670	0.3164	0.8998	**13.9986**	0.3197	0.8723

4.4 Understanding the role of uncertainty guidance

Setup We quantify the effect of UFG (the Monte Carlo version, UFG (MC)) on pixel-wise variance over 500 initial noises on ImageNet 256×256 with U-ViT ($\lambda = \omega = 0.4$). Specifically, we compare BayesDiff+CFG (global scale) and UFG with the same initial noises. We also provide corresponding visual comparisons between BayesDiff and UFG to evaluate their generative processes. To sum up, we illustrate how the generated images and uncertainty estimates evolve across timesteps, starting from the same initial noise.

Results We first compare results from BayesDiff(+CFG) and UFG, paired with the same initial noise $\boldsymbol{x}_T$ with a simple experiment. For each generated sample $\boldsymbol{x}_0$, let $\tilde{\sigma}^2(\boldsymbol{x}_0) \in [0,1]^d$ denote the min-max normalization of the $\mathrm{diag}(\mathrm{Var}(\boldsymbol{x}_0))$. We then calculate the *uncertainty dispersion* of $\boldsymbol{x}_0$, defined as $\psi(\boldsymbol{x}_0) := \frac{1}{d}\sum_{i=1}^{d}\left|\tilde{\sigma}_i^2(\boldsymbol{x}_0) - \frac{1}{d}\sum_{i'=1}^{d}\tilde{\sigma}_{i'}^2(\boldsymbol{x}_0)\right|$.

In other words, we examine whether the pixel-wise variances are uniformly distributed across pixels, since lower dispersion indicates that no localized region remains highly uncertain. We visualize the paired comparison result in Fig. 3. We can observe that nearly every sample lies above the $y = x$ line, indicating that UFG consistently yields a lower uncertainty dispersion on a per-sample basis. Consequently, this suggests that applying a pixel-wise adaptive guidance scale, as UFG does, stabilizes the pixel-wise variances.

We can also observe this phenomenon by a visual example of Fig. 4. It shows the sampling trajectories and corresponding uncertainty estimates from BayesDiff and UFG, starting from the same initial noise $\boldsymbol{x}_T$. Corresponding generated images are presented in Fig. 11. In BayesDiff, a horizontal artifact is gradually generated across the coffeepot body and extends into the background wall. We can also observe that the uncertainty estimates close to that area are relatively high. On the other hand, the high uncertainty estimates around the horizontal artifact do not appear in UFG. As shown in Fig. 3, UFG consistently reduces the uncertainty dispersion, which suggests that highly uncertain regions such as the horizontal artifact in Fig. 4 are effectively suppressed through our guidance. Additional qualitative results in App. E.3 confirm that this pattern holds consistently across different classes.

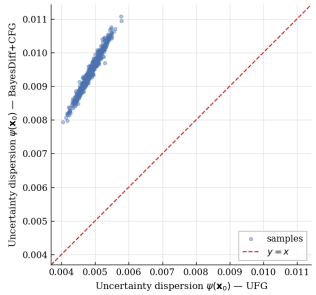


Figure 3: Paired comparison of the uncertainty dispersion.

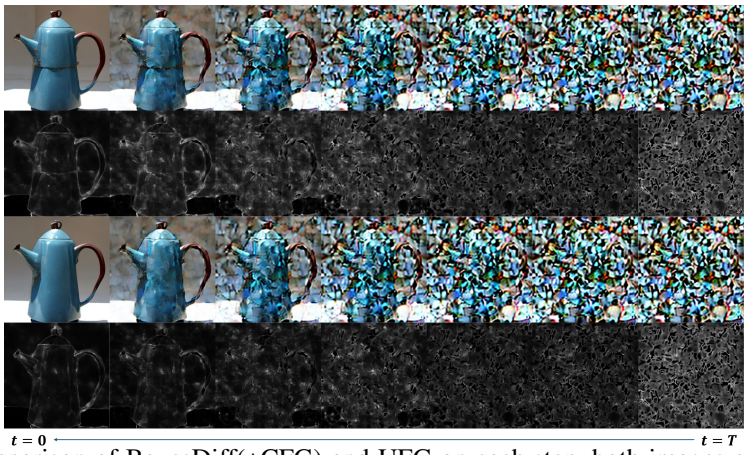


Figure 4: Comparison of BayesDiff(+CFG) and UFG on each step, both images and uncertainty estimates. Class: "Coffeepot" (505). (Top 2 rows) BayesDiff, (Bottom 2 rows) UFG.

4.5 Ablation studies and additional results

We conduct ablation studies on three components of UFG: (i) effect of λ, (ii) effect of Monte Carlo sample size, and (iii) effect of the simplified rule for γ_t. The results are provided in App. D. These studies validate the component choices used in our experiments. Additional visual results are provided in App. E for each experimental setup, including image editing via DDIM inversion, where UFG shows better editing quality than other baselines. We further test UFG against a heuristic dynamic guidance baseline, an alternative ensemble-based uncertainty estimator, and highly compositional prompts, reported in Apps. E.7 to E.9. The heuristic comparison isolates the benefit of Proposition 3.1's exact solution over a plain heuristic. Together with the sampling cost and the APG combination in Apps. E.5 and E.6, these examine UFG from several complementary angles.

5 Concluding remarks

We propose a novel guidance mechanism named Uncertainty-adaptive Feedback Guidance that can effectively utilize pixel-wise adaptive guidance based on uncertainty estimation. Rather than applying a uniform guidance scale across all pixels, UFG applies an adaptive scale to each pixel using pixel-wise uncertainty. Through various experiments, including class-conditional and text-to-image generation, as well as image editing, we demonstrate the advantages of UFG through quantitative metrics and qualitative examples. We believe that this work provides a solid framework built around pixel-wise uncertainty, reducing localized artifacts without a large cost in sample diversity.

Promising future research directions would be: (i) post-hoc refinement of a generated image by identifying the high-uncertainty pixels and applying adaptive guidance only to them, (ii) replacing the current schedule-dependent, per-step Bayesian approximation with a treatment of the full sampling path, and (iii) extension to video generation tasks.

AI USE STATEMENT

In this work, we used generative AI tools for none of the tasks with required disclosure. We have not used generative AI tools for developing theoretical models, formulating or proving mathematical claims, or any other task with required disclosure, and generating synthetic data sets, assisting with translation, and supporting qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure, specifically to aid or polish writing and for retrieval and discovery (e.g., finding related work). We have reviewed all AI-assisted work: every suggested wording change was edited and approved by the authors, and every AI-surfaced reference was checked against the primary source before it was cited. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work proposes a new guidance mechanism for diffusion models, which can be applied to text-to-image generation. Therefore, as with any image generation research, there is a potential risk of misuse such as generating harmful content. However, our method does not introduce a new architecture or enable fundamentally new capabilities. Therefore, we believe that our research does not create new risks beyond those already associated with existing diffusion models.

REPRODUCIBILITY STATEMENT

Detailed derivations and the full algorithm of UFG are provided in App. A, the proof of Proposition 3.1 and the corresponding numerical study in App. C, and experimental details (evaluation metrics, backbones, samplers, hyperparameters) in App. B.

REFERENCES

[1] Md Manjurul Ahsan, Shivakumar Raman, Yingtao Liu, and Zahed Siddique. A comprehensive survey on diffusion models and their applications. *Applied Soft Computing*, 181:113470, 2025.

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

[3] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0xiJLKH-ufZ.

[4] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023.

[5] Lucas Berry, Axel Brando, and David Meger. Shedding light on large generative networks: Estimating epistemic uncertainty in diffusion models. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

[6] Bin Cao, Jianhao Yuan, Yexin Liu, Jian Li, Shuyang Sun, Jing Liu, and Bo Zhao. Synartifact: Classifying and alleviating artifacts in synthetic images via vision-language model. *arXiv preprint arXiv:2402.18068*, 2024.

[7] Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. *arXiv preprint arXiv:2406.08070*, 2024.

[8] Michele De Vita and Vasileios Belagiannis. Diffusion model guided sampling with pixel-wise aleatoric uncertainty estimation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3844–3854. IEEE, 2025.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

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

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

[13] 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.

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

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[16] Metod Jazbec, Eliot Wong-Toi, Guoxuan Xia, Dan Zhang, Eric Nalisnick, and Stephan Mandt. Generative uncertainty in diffusion models. In Silvia Chiappa and Sara Magliacane (eds.), *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 1837–1858. PMLR, 21–25 Jul 2025. URL https://proceedings.mlr.press/v286/jazbec25a.html.

[17] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[18] Siqi Kou, Lei Gan, Dequan Wang, Chongxuan Li, and Zhijie Deng. Bayesdiff: Estimating pixel-wise uncertainty in diffusion via bayesian inference. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YcM6ofShwY.

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

[20] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. Rich human feedback for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19401–19411, 2024.

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

[22] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pp. 366–384. Springer, 2024.

[23] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

[24] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16784–16804. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nichol22a.html.

[25] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. URL https://github.com/toshas/torch-fidelity. Version: 0.4.0, DOI: 10.5281/zenodo.3786539.

[26] I. Pavlov, A. Ivanov, and S. Stafievskiy. Text-to-Image Benchmark: A benchmark for generative models. https://github.com/boomb0om/text2image-benchmark, September 2023. Version 0.1.0.

[27] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

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

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[30] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. In *International Conference on Learning Representations*, 2024.

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

[32] 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.

[33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[34] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[36] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[38] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

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

[40] Yiyang Wang, Xi Chen, Xiaogang Xu, Sihui Ji, Yu Liu, Yujun Shen, and Hengshuang Zhao. Diffdoctor: Diagnosing image diffusion models before treating. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18917–18926, 2025.

[41] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023.

[42] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=KRLUvxh8uaX`.

[43] Z-Image Team. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

[44] Lingzhi Zhang, Zhengjie Xu, Connelly Barnes, Yuqian Zhou, Qing Liu, He Zhang, Sohrab Amirghodsi, Zhe Lin, Eli Shechtman, and Jianbo Shi. Perceptual artifacts localization for image synthesis tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7579–7590, 2023.

A Detailed algorithm of UFG

A.1 Calculation of the moments

We can calculate the moments of

$$\boldsymbol{x}_{t-1} \sim \mathcal{N}(m^*_{t-1}, V^*_{t-1}) \tag{20}$$

as BayesDiff does.

Consider the update rule of Eq. (16), i.e.,

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t - \left(f(t)\boldsymbol{x}_t + \frac{g(t)^2}{\sigma_t}\boldsymbol{\epsilon}^*_t \right) + g(t)\boldsymbol{z},\ \boldsymbol{z} \sim \mathcal{N}(0, \mathbf{I}). \tag{21}$$

Note that we are considering the guided predictive distribution:

$$\boldsymbol{\epsilon}^*_t \sim p^*(\boldsymbol{\epsilon}_t | \boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D}). \tag{22}$$

From the update rule, we have the following:

$$\begin{aligned} m^*_{t-1} &= (1 - f(t))\mathbb{E}\boldsymbol{x}_t - \frac{g(t)^2}{\sigma_t}\mathbb{E}\boldsymbol{\epsilon}^*_t \\ &= (1 - f(t))m^*_t - \mathbb{E}_{\boldsymbol{x}_t}[\mathbb{E}_{\boldsymbol{\epsilon}^*_t|\boldsymbol{x}_t}[\boldsymbol{\epsilon}^*_t|\boldsymbol{x}_t]] \\ &= (1 - f(t))m^*_t - \mathbb{E}_{\boldsymbol{x}_t}[\mu^*_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})] \end{aligned} \tag{23}$$

$$\begin{aligned} V^*_{t-1} &= (1 - f(t))^2 \text{Var}(\boldsymbol{x}_t) - (1 - f(t))\frac{g(t)^2}{\sigma_t}\text{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_t) + \frac{g(t)^4}{\sigma_t^2}\text{Var}(\boldsymbol{\epsilon}^*_t) + g(t)^2\mathbf{I} \\ &\approx (1 - f(t))^2 V^*_t - (1 - f(t))\text{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_t) + \frac{g(t)^4}{\sigma_t^2}\text{Var}(\boldsymbol{\epsilon}^*_t) + g(t)^2\mathbf{I} \end{aligned} \tag{24}$$

We need to calculate $\text{Var}(\boldsymbol{\epsilon}^*_t)$ and $\text{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_t)$.

This is done by Monte Carlo, as BayesDiff does. We draw M samples from $\boldsymbol{x}^{(m)}_t \sim \mathcal{N}(m^*_t, V^*_t)$ from the previous step ($m = 1, \ldots, M$). For each sample, evaluate:

$$\mu^*_{t,m} := \mu_{\text{pred}}(\boldsymbol{x}^{(m)}_t, t, \boldsymbol{c}), \Sigma^*_{t,m} := \Sigma_{\text{pred}}(\boldsymbol{x}^{(m)}_t, t, \boldsymbol{c}). \tag{25}$$

Then we can approximate the quantities as follows:

$$\hat{\mu}_t := \frac{1}{M}\sum_{m=1}^{M} \mu^*_{t,m} \approx \mathbb{E}\boldsymbol{\epsilon}^*_t \tag{26}$$

$$\hat{\Sigma}_t := \frac{1}{M}\sum_{m=1}^{M} \Sigma_{t,m} \approx \mathbb{E}[\Sigma^*_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})] \tag{27}$$

$$\hat{C}_t := \frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{x}^{(m)}_t (\mu^*_{t,m})^\top) - \left(\frac{1}{M}\sum_{m=1}^{M} \boldsymbol{x}^{(m)}_t \right) \hat{\mu}^\top_t \approx \text{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_t) \tag{28}$$

A Taylor-based alternative that replaces the M evaluations with one is given in App. A.4.

In summary, the calculation is done as follows:

$$m^*_{t-1} = (1 - f(t))m_t - \frac{g(t)^2}{\sigma_t}\hat{\mu}_t \tag{29}$$

$$V^*_{t-1} = (1 - f(t))^2 V^*_t - (1 - f(t))\frac{g(t)^2}{\sigma_t}\hat{C}_t + \frac{g(t)^4}{\sigma_t^2}\hat{\Sigma}_t + g(t)^2\mathbf{I} \tag{30}$$

A.2 Extension to other sampling methods

The extension to other sampling methods is analogous to the calculation from BayesDiff. The only difference is the distribution of $\boldsymbol{\epsilon}_t$, since we are considering $\boldsymbol{\epsilon}^*_t$ and its distribution p^* when sampling. Here, we provide the iteration rule for readers' convenience.

Analytic-DPM [3] The sampling method of Analytic-DPM is as follows:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha'_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha'_t}{\sqrt{1-\bar{\alpha}'_t}}\boldsymbol{\epsilon}^*_t\right) + \sigma_t \boldsymbol{z} \tag{31}$$

The moments can be calculated as follows:

$$\mathbb{E}[\boldsymbol{x}_{t-1}] = \frac{1}{\sqrt{\alpha'_t}}\mathbb{E}[\boldsymbol{x}_t] - \mathbb{E}[\boldsymbol{\epsilon}^*_t] \tag{32}$$

$$\mathrm{Var}(\boldsymbol{x}_{t-1}) = \frac{1}{\alpha'_t}\mathrm{Var}(\boldsymbol{x}_t) - 2\frac{1-\alpha'_t}{\alpha'_t\sqrt{1-\bar{\alpha}'_t}}\mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_t) + \frac{(1-\alpha'_t)^2}{\alpha'_t(1-\bar{\alpha}'_t)}\mathrm{Var}(\boldsymbol{\epsilon}^*_t) + \sigma_t^2 \tag{33}$$

DDIM [35] The sampling method of DDIM is as follows:

$$\boldsymbol{x}_{t-1} = \alpha_{t-1}\left(\frac{\boldsymbol{x}_t - \sigma_t\boldsymbol{\epsilon}^*_t}{\alpha_t}\right) + \sigma_{t-1}\boldsymbol{\epsilon}^*_t \tag{34}$$

The moments can be calculated as follows:

$$\mathbb{E}[\boldsymbol{x}_{t-1}] = \frac{\alpha_{t-1}}{\alpha_t}\mathbb{E}[\boldsymbol{x}_t] + \left(\sigma_{t-1} - \frac{\alpha_{t-1}}{\alpha_t}\sigma_t\right)\mathbb{E}[\boldsymbol{\epsilon}^*_t] \tag{35}$$

$$\begin{aligned}\mathrm{Var}(\boldsymbol{x}_{t-1}) = &\frac{\alpha_{t-1}^2}{\alpha_t^2}\mathrm{Var}(\boldsymbol{x}_t) + 2\frac{\alpha_{t-1}}{\alpha_t}\left(\sigma_{t-1} - \frac{\alpha_{t-1}}{\alpha_t}\sigma_t\right)\mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_t) \\ &+ \left(\sigma_{t-1} - \frac{\alpha_{t-1}}{\alpha_t}\sigma_t\right)^2\mathrm{Var}(\boldsymbol{\epsilon}^*_t)\end{aligned} \tag{36}$$

2nd-order DPM-Solver [23] The sampling method of 2nd-order DPM-Solver is as follows, involving hidden states of $\boldsymbol{x}_{s_t}$ between $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$:

$$\boldsymbol{x}_{s_t} = \frac{\alpha_{s_t}}{\alpha_t}\boldsymbol{x}_t - \sigma_{s_t}\left(e^{\frac{h_t}{2}} - 1\right)\boldsymbol{\epsilon}^*_t \tag{37}$$

$$\boldsymbol{x}_{t-1} = \frac{\alpha_{t-1}}{\alpha_t}\boldsymbol{x}_t - \sigma_{t-1}\left(e^{h_t} - 1\right)\boldsymbol{\epsilon}^*_{s_t} \tag{38}$$

where $\beta_t = \log\frac{\alpha_t}{\sigma_t}$ and $h_t = \beta_{t-1} - \beta_t$.

The moments can be calculated as follows:

$$\mathbb{E}[\boldsymbol{x}_{t-1}] = \frac{\alpha_{t-1}}{\alpha_t}\mathbb{E}[\boldsymbol{x}_t] - \sigma_{t-1}(e^{h_t} - 1)\mathbb{E}[\boldsymbol{\epsilon}^*_{s_t}] \tag{39}$$

$$\mathrm{Var}(\boldsymbol{x}_{t-1}) = \frac{\alpha_{t-1}^2}{\alpha_t^2}\mathrm{Var}(\boldsymbol{x}_t) - 2\frac{\alpha_{s_t}}{\alpha_t}\sigma_{t-1}(e^{h_t} - 1)\mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_{s_t}) + \sigma_{t-1}^2(e^{h_t} - 1)^2\mathrm{Var}(\boldsymbol{\epsilon}^*_{s_t}) \tag{40}$$

Additional $\boldsymbol{x}_{s_t}$ are sampled from $\boldsymbol{x}_{s_t}|\boldsymbol{x}_t$ to calculate $\mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_{s_t})$ by Monte Carlo, as in Kou et al. [18].

A.3 Simple variant: UFG-M

We provide a pseudocode for the simple variant, UFG-M, in Algorithm 2. The change in $\boldsymbol{\epsilon}_t$ is annotated with green.

A.4 Practical acceleration

We admit the same practical acceleration as BayesDiff does. It can be done by skipping MC estimation and sampling $\boldsymbol{\epsilon}^*_t$ on several timesteps, depending on the pre-defined schedule. In other words, we use deterministic sampling of $\boldsymbol{\epsilon}^*_t = \mu^*_{\mathrm{pred}}$ and let $\mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}^*_t), \mathrm{Var}(\boldsymbol{\epsilon}^*_t)$ be zero, at some timesteps.

Algorithm 2: UFG with Mean vector (UFG-M)

Input : Starting point $\boldsymbol{x}_T$, Monte Carlo sample size M, pre-trained noise prediction model $\boldsymbol{\epsilon}_\theta$, condition $\boldsymbol{c}$, guidance strength λ.

Output : Image generation $\boldsymbol{x}_0$ and pixel-wise uncertainty $\mathrm{Var}(\boldsymbol{x}_0)$.

Construct the pixel-wise variance prediction function Σ_{pred} via Laplace approximation.

$\mathbb{E}(\boldsymbol{x}_T) \leftarrow \boldsymbol{x}_T$, $\mathrm{Var}(\boldsymbol{x}_T) \leftarrow \mathbf{0}$, $\mathrm{Cov}(\boldsymbol{x}_T, \boldsymbol{\epsilon}_T^*) \leftarrow \mathbf{0}$, $\gamma_T \leftarrow \mathbf{0}$

for $t = T \to 1$ **do**

  Compute μ^c, Σ^c from $p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D})$ and μ^u, Σ^u from $p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \varnothing, \mathcal{D})$.

  Compute guided predictive parameters:

$$\mu^*_{\mathrm{pred}} \leftarrow \mu^c + \gamma_t \odot (\mu^c - \mu^u)$$
$$\Sigma^*_{\mathrm{pred}} \leftarrow (1+\gamma_t)^{\circ 2} \odot \Sigma^c + (\gamma_t)^{\circ 2} \odot \Sigma^u$$

  Let $\boldsymbol{\epsilon}_t^* = \mu^*_{\mathrm{pred}}$.

  Obtain $\boldsymbol{x}_{t-1}$ via Eq. (16).

  Estimate $\mathbb{E}(\boldsymbol{x}_{t-1}) \leftarrow m^*_{t-1}$ and $\mathrm{Var}(\boldsymbol{x}_{t-1}) \leftarrow V^*_{t-1}$.

  Sample $\boldsymbol{x}_{t-1}^{(m)} \sim \mathcal{N}(m^*_{t-1}, V^*_{t-1})$, $m = 1, \ldots, M$.

  Estimate $\mathrm{Cov}(\boldsymbol{x}_{t-1}, \boldsymbol{\epsilon}^*_{t-1})$.

  Update γ_{t-1}.

Taylor-based moment estimation (UFG-accel) Step skipping reduces the number of steps that need uncertainty quantification. The cost of the remaining steps comes from App. A.1, which evaluates the network at M Monte Carlo samples $\boldsymbol{x}_t^{(m)}$. UFG-accel replaces these M evaluations with a first-order Taylor expansion around m_t^*. In the tables we call the Monte Carlo estimator UFG (MC) and the Taylor-based one UFG-accel.

Let $\boldsymbol{x}_t \sim \mathcal{N}(m_t^*, V_t^*)$. Expanding the guided predictive mean around m_t^* gives

$$\mu^*_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}) \approx \mu^*_{\mathrm{pred}}(m_t^*, t, \boldsymbol{c}) + J_t(\boldsymbol{x}_t - m_t^*), \tag{41}$$

where J_t is the Jacobian of μ^*_{pred} at m_t^*. Since $\mathbb{E}[\boldsymbol{x}_t - m_t^*] = 0$, the linear term vanishes in expectation, and the neglected remainder is of second order in the entries of V_t^*. Hence a single evaluation at the mean gives

$$\hat{\mu}_t \approx \mu^*_{\mathrm{pred}}(m_t^*, t, \boldsymbol{c}), \qquad \hat{\Sigma}_t \approx \Sigma^*_{\mathrm{pred}}(m_t^*, t, \boldsymbol{c}), \tag{42}$$

where the second approximation follows in the same way. The covariance is the one term that does not collapse to a single point. We write $\mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{\epsilon}_t^*) = \mathbb{E}[\boldsymbol{x}_t \boldsymbol{\epsilon}_t^{*\top}] - \mathbb{E}[\boldsymbol{x}_t]\,\mathbb{E}[\boldsymbol{\epsilon}_t^*]^\top$ and take the second term as $m_t^* \hat{\mu}_t^\top$. For the first term, the tower property gives $\mathbb{E}[\boldsymbol{x}_t \boldsymbol{\epsilon}_t^{*\top}] = \mathbb{E}_{\boldsymbol{x}_t}[\boldsymbol{x}_t\, \mu^*_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})^\top]$. We estimate it with the one sample $\boldsymbol{x}_t$ that the sampler already holds, so that

$$\hat{C}_t \approx \boldsymbol{x}_t\, \mu^*_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})^\top - m_t^* \hat{\mu}_t^\top. \tag{43}$$

The evaluation of $\mu^*_{\mathrm{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c})$ is needed anyway to draw $\boldsymbol{\epsilon}_t^*$. The only extra cost per step is therefore one evaluation at m_t^*, that is one conditional and one unconditional forward pass, instead of M of each. Algorithm 3 lists the changes with respect to Algorithm 1. The guidance scale γ_t is still the solution s^* of Proposition 3.1.

The two accelerations act on different parts of a step. Step skipping decides at which steps uncertainty is quantified, and the Taylor approximation decides how the moments are computed at those steps. At a skipped step nothing changes, and at every other step UFG-accel uses the single evaluation above. They can therefore be combined. Our experiments use the same skipping schedule for UFG (MC) and UFG-accel.

Algorithm 3: UFG with Taylor-based moments (UFG-accel)

Input : Starting point $\boldsymbol{x}_T$, pre-trained noise prediction model $\boldsymbol{\epsilon}_\theta$, condition $\boldsymbol{c}$, guidance strength λ.
Output : Image generation $\boldsymbol{x}_0$ and pixel-wise uncertainty $\mathrm{Var}(\boldsymbol{x}_0)$.

Construct the pixel-wise variance prediction function Σ_{pred} via Laplace approximation.
$\mathbb{E}(\boldsymbol{x}_T) \leftarrow \boldsymbol{x}_T$, $\mathrm{Var}(\boldsymbol{x}_T) \leftarrow \mathbf{0}$, $\mathrm{Cov}(\boldsymbol{x}_T, \boldsymbol{\epsilon}_T) \leftarrow \mathbf{0}$, $\gamma_T \leftarrow \mathbf{0}$
for $t = T \to 1$ **do**
  Compute μ^c, Σ^c from $p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \boldsymbol{c}, \mathcal{D})$ and μ^u, Σ^u from $p(\boldsymbol{\epsilon}_t|\boldsymbol{x}_t, t, \varnothing, \mathcal{D})$.
  Compute guided predictive parameters μ^*_{pred} and Σ^*_{pred} as in Algorithm 1.
  Sample $\boldsymbol{\epsilon}^*_t \sim \mathcal{N}(\mu^*_{\mathrm{pred}}, \Sigma^*_{\mathrm{pred}})$ and obtain $\boldsymbol{x}_{t-1}$ via Eq. (16).
  Evaluate $\mu^*_{\mathrm{pred}}, \Sigma^*_{\mathrm{pred}}$ once at m^*_t to get $\hat{\mu}_t, \hat{\Sigma}_t$, and set $\hat{C}_t \leftarrow \boldsymbol{x}_t(\mu^*_{\mathrm{pred}})^\top - m^*_t\hat{\mu}_t^\top$.
  Estimate $\mathbb{E}(\boldsymbol{x}_{t-1}) \leftarrow m^*_{t-1}$ and $\mathrm{Var}(\boldsymbol{x}_{t-1}) \leftarrow V^*_{t-1}$ from $\hat{\mu}_t, \hat{\Sigma}_t, \hat{C}_t$.
  Update γ_{t-1}.

B EXPERIMENTAL DETAILS

B.1 EVALUATION MEASURES

FID Fréchet Inception Distance (FID) [13] quantifies the distributional distance between a set of real images and a set of generated images in the feature space of a pretrained Inception-v3 network [38]. Let $\phi(\cdot)$ denote the Inception-v3 feature extractor that maps an image to a 2048 dimensional vector. Given a set of real images $\{x_0^{(i)}\}_{i=1}^{N_r}$ drawn from the data distribution $q_0(x_0)$ and a set of generated images $\{\hat{x}_0^{(j)}\}_{j=1}^{N_g}$, we fit multivariate Gaussians to each feature set:

$$\phi(x_0^{(i)}) \sim \mathcal{N}(\mu_r,\, \Sigma_r), \qquad \phi(\hat{x}_0^{(j)}) \sim \mathcal{N}(\mu_g,\, \Sigma_g), \tag{44}$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the empirical mean and covariance of the real and generated feature vectors, respectively. Then, the FID is defined as the Fréchet distance between these two Gaussians:

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\Big(\Sigma_r + \Sigma_g - 2\,(\Sigma_r\,\Sigma_g)^{1/2}\Big). \tag{45}$$

A lower FID indicates that the distribution of generated images more closely matches that of the real images in Inception feature space. In our class-conditional experiments, we use the ImageNet validation set (50,000 images) as the validation set and compute FID using the `torch-fidelity` library [25]. For text-to-image experiments, we compute FID against the MSCOCO 2014 validation set using the `text2image-benchmark` library [26].

Precision & Recall We adopt the improved precision and recall metric proposed by Kynkäänniemi et al. [19], which separately measures the fidelity and diversity of generated samples. Using the same Inception-v3 feature extractor $\phi(\cdot)$, let $\Phi_r = \{\phi(\boldsymbol{x}_0^{(i)})\}_{i=1}^{N_r}$ and $\Phi_g = \{\phi(\hat{\boldsymbol{x}}_0^{(j)})\}_{j=1}^{N_g}$ denote the feature sets of real and generated images, respectively. Specifically, a neighborhood is defined around each sample using its k-th nearest neighbor. For each sample $\boldsymbol{u} \in \Phi$, a ball of radius $\|\boldsymbol{u} - \mathrm{NN}_k(\boldsymbol{u}, \Phi)\|_2$ is placed at $\boldsymbol{u}$, where $\mathrm{NN}_k(\boldsymbol{u}, \Phi)$ denotes the k-th nearest neighbor of $\boldsymbol{u}$ in Φ. A feature vector $\boldsymbol{v}$ is considered to belong to Φ if it falls inside at least one such ball. This is formalized by the following binary function:

$$f(\boldsymbol{v}, \Phi) = \begin{cases} 1 & \text{if } \exists\, \boldsymbol{u} \in \Phi \text{ s.t. } \|\boldsymbol{v} - \boldsymbol{u}\|_2 \le \|\boldsymbol{u} - \mathrm{NN}_k(\boldsymbol{u}, \Phi)\|_2, \\ 0 & \text{otherwise.} \end{cases} \tag{46}$$

Precision and recall are then defined as:

$$\mathrm{Precision} = \frac{1}{N_g}\sum_{j=1}^{N_g} f\Big(\phi(\hat{\boldsymbol{x}}_0^{(j)}),\, \Phi_r\Big), \qquad \mathrm{Recall} = \frac{1}{N_r}\sum_{i=1}^{N_r} f\Big(\phi(\boldsymbol{x}_0^{(i)}),\, \Phi_g\Big). \tag{47}$$

Intuitively, precision measures the fraction of generated samples that lie close to real samples, reflecting sample fidelity. Recall measures the fraction of real samples that lie close to generated samples, reflecting diversity. Higher values are better for both metrics.

CLIP-score CLIP-score [12] measures the semantic alignment between a generated image and its corresponding text prompt without requiring ground-truth reference images. Given a generated image x and a text prompt $\boldsymbol{c}$, it is defined as the cosine similarity between the image embedding and the text embedding extracted from a pretrained CLIP model [28], a vision-language model trained on large scale image-text pairs:

$$\text{CLIP-score}(\boldsymbol{x}, \boldsymbol{c}) = \cos\big(E_I(\boldsymbol{x}),\, E_T(\boldsymbol{c})\big), \tag{48}$$

where E_I and E_T denote the CLIP image encoder and text encoder, respectively. A higher CLIP-score indicates stronger alignment between the generated image and the given prompt. We use the ViT-B/32 [11] architecture as the CLIP backbone for all evaluations.

VQAScore VQAScore [22] measures the alignment between a generated image and its text prompt by leveraging a visual question answering (VQA) model. Given an image x and a text prompt $\boldsymbol{c}$, it is defined as the probability that the VQA model answers "Yes" to the question "Does this figure show '$\{c\}$'? Please answer yes or no.":

$$\text{VQAScore}(\boldsymbol{x}, \boldsymbol{c}) = P\big(\text{“Yes”} \mid \boldsymbol{x},\, \text{“Does this figure show ‘}\{\boldsymbol{c}\}\text{’?”}\big). \tag{49}$$

Compared to CLIP-score, VQAScore is more sensitive to compositional structures such as object attributes and spatial relationships, as the bidirectional image-question encoder allows the visual and textual representations to interact before scoring. We use CLIP-FlanT5-XL [22] as the VQA backbone.

B.2 Implementation Details

Class-conditional generation We adopt U-ViT-H/2 [4] as the backbone diffusion model for both 256×256 and 512×512 resolutions on ImageNet [9]. We use the 2nd-order DPM-Solver [23] with $T = 50$ sampling steps and $M = 10$ Monte Carlo samples for uncertainty estimation. The CFG scale and the UFG scaling parameter are set to $\omega = \lambda = 0.4$ for 256×256 and $\omega = \lambda = 0.7$ for 512×512, since they are considered as the best values for each resolution in U-ViT. For the last-layer Laplace approximation and step skipping schedule, we follow the configuration of BayesDiff [18].

Text-to-image generation For text-to-image experiments, we use Stable Diffusion v1.5 [29] and SDXL [27] with DDIM [35] as the sampler over $T = 50$ steps. The CFG scale and the UFG scaling parameter are both set to $\omega = \lambda = 7.5$, which is a standard value for SD, and we draw $M = 10$ Monte Carlo samples for uncertainty estimation. For Z-Image, we use the CFG scale $w = 4$, $M = 4$ Monte Carlo samples, and 512×512 images. Bayesian inference is performed on a subset of LAION-ART [33], and evaluation is conducted on the MSCOCO-30K dataset [21]. For the last-layer Laplace approximation and step skipping schedule, we follow the configuration of BayesDiff [18].

Baselines CFG++ [7] uses $\lambda = 0.377$ (256×256) and 0.458 (512×512) on U-ViT, $\lambda = 0.06$ on DiT-XL/2, 0.506 on Z-Image and 0.6 on SDXL. CADS [30] uses the noise schedule $\tau \in [0.7, 0.9]$ with scale $s = 0.05$ on U-ViT at 256×256 and DiT-XL/2 (with $\psi = 1$), $\tau \in [0.625, 1.0]$ with $s = 0.07$ on U-ViT at 512×512 and on Z-Image, and $\tau \in [0.6, 0.9]$ with $s = 0.25$ on SDXL. AU on DiT-XL/2 uses the conditional model without CFG and DDPM sampling with 250 steps, with $M = 5$, percentile 0.95, learning rate 1.0, gradient clamp ± 10 and gradient scale 0.25, applied in the first 5 steps only.

Computational environments Unless otherwise noted, all experiments are conducted on a machine equipped with two NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs, each with 96GB memory, an AMD EPYC 9254 24-Core Processor, and 128GB RAM.

C ABOUT CHOICE OF γ_t

We provide the detailed proof of Proposition 3.1 and the corresponding numerical study here.

Proof of Proposition 3.1. Let

$$\mu^c(\boldsymbol{x}_t) := \mu_{\text{pred}}(\boldsymbol{x}_t, t, \boldsymbol{c}), \qquad \mu^u(\boldsymbol{x}_t) := \mu_{\text{pred}}(\boldsymbol{x}_t, t, \varnothing), \qquad \Delta\mu(\boldsymbol{x}_t) := \mu^c(\boldsymbol{x}_t) - \mu^u(\boldsymbol{x}_t).$$

For a pixel-wise guidance scale $s = (s_1, \ldots, s_d)$, define

$$\boldsymbol{\epsilon}^*_{t,i} = (1 + s_i)\boldsymbol{\epsilon}^c_{t,i} - s_i\boldsymbol{\epsilon}^u_{t,i}, \qquad \mathbb{E}[\boldsymbol{\epsilon}^*_{t,i} \mid \boldsymbol{x}_t] = \mu^c_i(x_t) + s_i\,\Delta\mu_i(x_t).$$

Consider the reverse update (Here, we abbreviated $\boldsymbol{x}_{t-1}(s)$ as $\boldsymbol{x}_{t-1}$ for simplicity.)

$$\boldsymbol{x}_{t-1} = A\boldsymbol{x}_t - B\boldsymbol{\epsilon}^*_t + g(t)\xi.$$

By the laws of total covariance and total variance, the one-step marginal variance admits the decomposition

$$\begin{aligned}\mathrm{Var}(\boldsymbol{x}_{t-1,i}) &= A^2\mathrm{Var}(\boldsymbol{x}_{t,i}) - 2AB\mathrm{Cov}(\boldsymbol{x}_{t,i}, \boldsymbol{\epsilon}^*_{t,i}) + B^2\mathrm{Var}(\boldsymbol{\epsilon}^*_{t,i}) + g(t)^2 \\ &= p_i + \ell_i s_i + q_i s_i^2,\end{aligned} \tag{50}$$

where

$$p_i = A^2\tau_{t,i} - 2AB\,c_i^{(0)} + B^2V_0^{(i)} + g(t)^2, \tag{51}$$

$$\ell_i = -2AB\,c_i^{(1)} + 2B^2V_1^{(i)}, \tag{52}$$

$$q_i = B^2V_2^{(i)}. \tag{53}$$

Here

$$\tau_{t,i} := \mathrm{Var}(\boldsymbol{x}_{t,i}),$$

$$c_i^{(0)} := \mathrm{Cov}(\boldsymbol{x}_{t,i}, \mu_i^c(\boldsymbol{x}_t)), \qquad c_i^{(1)} := \mathrm{Cov}(\boldsymbol{x}_{t,i}, \Delta\mu_i(\boldsymbol{x}_t)),$$

and

$$V_0^{(i)} := \sigma_i^{(0)} + \bar{\Sigma}_i^c, \qquad V_1^{(i)} := \sigma_i^{(1)} + \bar{\Sigma}_i^c, \qquad V_2^{(i)} := \sigma_i^{(2)} + \bar{\Sigma}_i^c + \bar{\Sigma}_i^u,$$

with

$$\sigma_i^{(0)} := \mathrm{Var}(\mu_i^c(\boldsymbol{x}_t)), \qquad \sigma_i^{(1)} := \mathrm{Cov}(\mu_i^c(\boldsymbol{x}_t), \Delta\mu_i(\boldsymbol{x}_t)), \qquad \sigma_i^{(2)} := \mathrm{Var}(\Delta\mu_i(\boldsymbol{x}_t)),$$

and

$$\bar{\Sigma}_i^c := \mathbb{E}[\Sigma_i^c(\boldsymbol{x}_t)], \qquad \bar{\Sigma}_i^u := \mathbb{E}[\Sigma_i^u(\boldsymbol{x}_t)].$$

Assume $q_i > 0$ for all i, so that Eq. (50) is strictly convex in s_i.

We solve

$$\min_{s\in\mathbb{R}^d} \sum_{i=1}^d \big(p_i + \ell_i s_i + q_i s_i^2\big) \qquad \text{s.t.} \quad \frac{1}{d}\sum_{i=1}^d s_i \le \lambda, \quad s_i \ge 0 \ \ \forall i. \tag{54}$$

The Lagrangian is

$$\mathcal{L}(s, \nu, \kappa) = \sum_{i=1}^d \big(p_i + \ell_i s_i + q_i s_i^2\big) + \nu\Big(\sum_{i=1}^d s_i - d\lambda\Big) - \sum_{i=1}^d \kappa_i s_i,$$

where $\nu \ge 0$ and $\kappa_i \ge 0$. The KKT conditions are

$$\ell_i + 2q_i s_i + \nu - \kappa_i = 0, \tag{55}$$

$$\kappa_i s_i = 0, \tag{56}$$

$$\nu\Big(\sum_{i=1}^d s_i - d\lambda\Big) = 0, \tag{57}$$

$$s_i \ge 0, \qquad \nu \ge 0, \qquad \kappa_i \ge 0. \tag{58}$$

Hence

$$s_i^* = \frac{1}{2q_i}\,[-\ell_i - \nu]_+. \tag{59}$$

If the unconstrained minimizer $(\frac{1}{2q_i}[-\ell_i]_+)_{i=1}^d$ is already feasible, i.e. their average is less than λ, then $\nu = 0$. Otherwise, $\nu > 0$ is uniquely determined by the budget condition

$$\frac{1}{d}\sum_{i=1}^{d} \frac{1}{2q_i}\,[-\ell_i - \nu]_+ = \lambda, \tag{60}$$

which can be done easily by a binary search.

□

The Monte Carlo estimation of each coefficient can be done as follows. Note that we have samples $\{\boldsymbol{x}_t^{(m)}\}_{m=1}^M$ at timestep t. For each sample, compute

$$\mu_i^{c,(m)} := \mu_i^c(\boldsymbol{x}_t^{(m)}), \qquad \mu_i^{u,(m)} := \mu_i^u(\boldsymbol{x}_t^{(m)}), \qquad \Delta\mu_i^{(m)} := \mu_i^{c,(m)} - \mu_i^{u,(m)}.$$

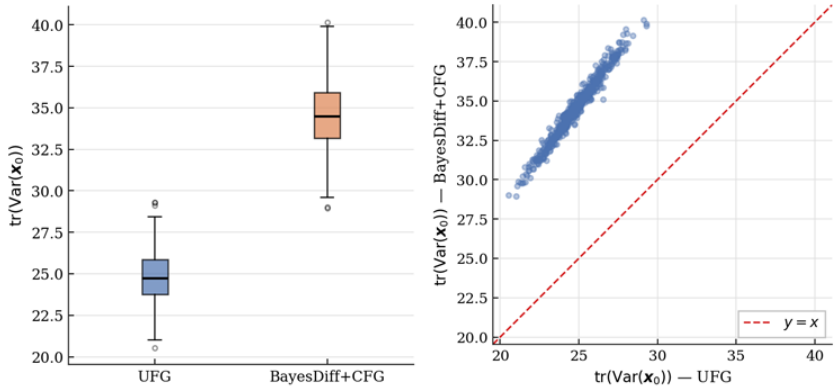
If predictive variances are available, also evaluate

$$\Sigma_i^{c,(m)} := \Sigma_i^c(\boldsymbol{x}_t^{(m)}), \qquad \Sigma_i^{u,(m)} := \Sigma_i^u(\boldsymbol{x}_t^{(m)}).$$

Then all coefficients above are estimated by sample moments:

$$\begin{aligned}
&\hat{\gamma}_{t,i} = \widehat{\mathrm{Var}}_m\big(\boldsymbol{x}_{t,i}^{(m)}\big),\\
&\hat{c}_i^{(0)} = \widehat{\mathrm{Cov}}_m\big(\boldsymbol{x}_{t,i}^{(m)}, \mu_i^{c,(m)}\big),\ \hat{c}_i^{(1)} = \widehat{\mathrm{Cov}}_m\big(\boldsymbol{x}_{t,i}^{(m)}, \Delta\mu_i^{(m)}\big),\\
&\hat{\sigma}_i^{(0)} = \widehat{\mathrm{Var}}_m\big(\mu_i^{c,(m)}\big),\ \hat{\sigma}_i^{(1)} = \widehat{\mathrm{Cov}}_m\big(\mu_i^{c,(m)}, \Delta\mu_i^{(m)}\big),\ \hat{\sigma}_i^{(2)} = \widehat{\mathrm{Var}}_m\big(\Delta\mu_i^{(m)}\big),\\
&\hat{\Sigma}_i^c = \frac{1}{M}\sum_{m=1}^{M}\Sigma_i^{c,(m)},\ \hat{\Sigma}_i^u = \frac{1}{M}\sum_{m=1}^{M}\Sigma_i^{u,(m)}.
\end{aligned}$$

Numerical study We further demonstrate this variance reduction effect by the following numerical study on ImageNet 256×256 using U-ViT with $\lambda = \omega = 0.4$. For 500 shared initial noise samples, we run both UFG and BayesDiff+CFG and record the trace of the variance at $t = 0$. Fig. 5(a) shows that UFG concentrates at substantially lower variance, with the median dropping from approximately 34 to 25. The paired scatter plot in Fig. 5(b) confirms that nearly every sample falls above the $y = x$ line, indicating that this reduction holds consistently on a per-sample basis rather than being driven by a few outliers.



(a) Boxplot of $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_0))$ on each. (b) Paired comparison of $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_0))$.

Figure 5: Variance reduction effect of UFG on ImageNet 256×256. We compare the $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_0))$ obtained from each method, by boxplots and paired scatterplot.

D ABLATION STUDIES

D.1 GUIDANCE STRENGTH λ

Setup To investigate the effect of the guidance strength λ on generation quality, we conduct ablation experiments using U-ViT trained on ImageNet at 256×256 and 512×512 resolutions. For each configuration, we synthesize 50,000 images and measure FID, precision, and recall. We vary λ across a range of values while keeping all other hyperparameters fixed, and report the results in Figs. 6 and 7.

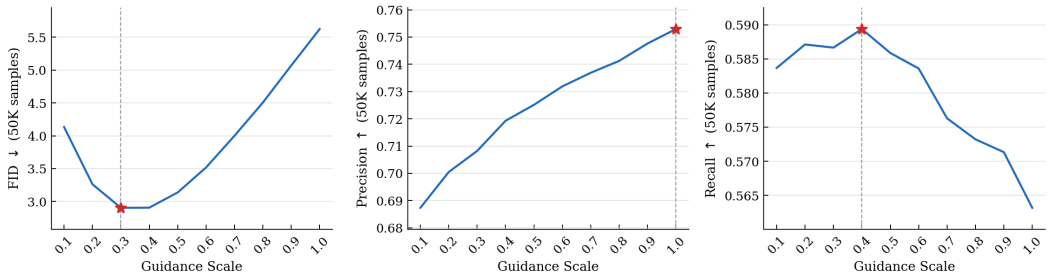


Figure 6: Effect of guidance strength λ on FID (↓), Precision (↑), and Recall (↑) for U-ViT on ImageNet 256×256.

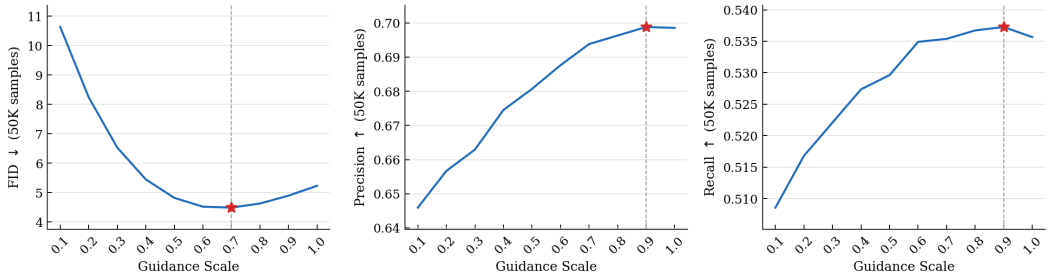


Figure 7: Effect of guidance strength λ on FID (↓), Precision (↑), and Recall (↑) for U-ViT on ImageNet 512×512.

Results Figs. 6 and 7 summarize the effect of λ on ImageNet 256×256 and 512×512, respectively. Across both resolutions, a consistent non-monotonic trend is observed. As λ increases from zero, FID initially decreases and precision improves, indicating that moderate guidance effectively enhances generation quality. However, beyond a certain threshold, further increasing λ degrades metrics. This behavior aligns with the trade-off commonly observed in classifier-free guidance: while stronger guidance sharpens conditional alignment, excessively large guidance scales distort the learned distribution. In our setting this effect is amplified because λ controls the total budget of pixel-wise guidance, so an excessively large λ inflates the guidance even for pixels with moderate uncertainty. These results confirm that UFG requires a moderate guidance strength to balance artifact suppression against distributional fidelity, and the optimal λ may vary across resolutions and may also vary across backbone architectures.

D.2 MONTE CARLO SAMPLE SIZE M

Setup To examine the sensitivity of UFG to the number of Monte Carlo samples used for uncertainty estimation, we vary $M \in \{3, 5, 7, 10, 15, 20\}$ while keeping the guidance strength fixed at $\lambda = 0.4$. All experiments use U-ViT on ImageNet 256×256. For each value of M, we synthesize 10,000 images and measure FID, Precision, and Recall. Results are reported in Fig. 8.

Results As shown in Fig. 8, all three metrics remain virtually unchanged across the entire range of M, demonstrating that UFG is robust to the choice of Monte Carlo sample size. Notably, the y-axis scales reveal that the observed fluctuations are negligible. FID varies within a band of 0.025,

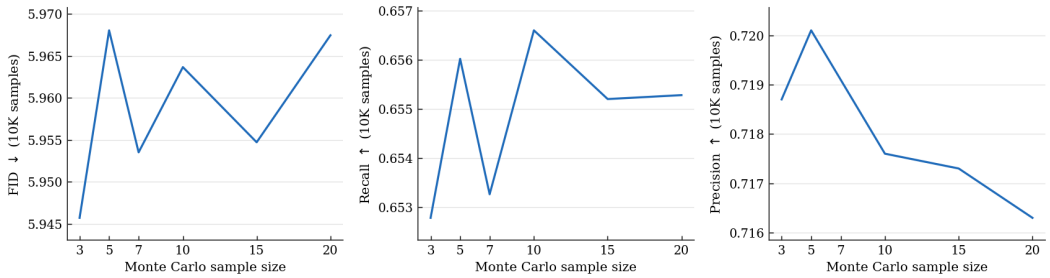


Figure 8: Effect of Monte Carlo sample size M on FID (↓), Recall (↑), and Precision (↑) for U-ViT on ImageNet 256×256.

Precision stays between 0.716 and 0.720, and Recall differs by less than 0.004. In particular, $M = 3$ achieves comparable performance to larger sample sizes, confirming that the Monte Carlo sample size is not a sensitive hyperparameter of UFG.

D.3 Simplified rule via power exponent of variance estimate

Setup Proposition 3.1 provides the exact pixel-wise guidance scale s^* that minimizes the one-step variance $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_{t-1}))$. Rather than computing s^*, we can consider a simpler alternative. Use a normalized power of the variance estimate, as follows:

$$\gamma_{t,i} \propto \eta_{t,i}^{p}, \quad \eta_t = \mathrm{diag}(V_t^*), \tag{61}$$

scaled so that $\frac{1}{d}\sum_i \gamma_{t,i} = \lambda$. We compare this simplified rule against the exact optimum s^* for $p \in \{0.5, 0.75, 1.0, 1.5\}$, keeping $\lambda = \omega = 0.4$ and all other hyperparameters fixed. We generate 500 images on ImageNet 256×256 using U-ViT and record $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_t))$ at each step.

Results Fig. 9 plots the ratio of the mean $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_t))$ relative to the exact optimum at timesteps $t \in \{20, 15, 10, 5\}$. For $p \leq 1$, the variance stays within approximately 10% of the optimum across all timesteps, indicating that the simplified rule yields results similar to the exact solution. In contrast, $p = 1.5$ progressively deviates as the chain advances, reaching roughly 20% above the optimum at $t = 5$. Fig. 10 shows the per-sample distribution of $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_t))$ at the same timesteps. At early steps ($t = 20, 15$), all values of p produce distributions close to the optimum. At later steps ($t = 10, 5$), $p = 1.5$ shifts both the median and the upper tail upward, with several outliers exceeding twice the median of the optimum at $t = 5$. The degradation at large p can be attributed to the inflated guided predictive variance Σ^*_{pred} in Eq. (15), whose effect accumulates over subsequent steps. Since any $p \in [0.5, 1.0]$ performs comparably to the exact optimum, we suggest using $p = 0.5$ as the default simplified rule, corresponding to the standard deviation scale $\eta_t^{1/2}$, which is a conventional choice for quantifying uncertainty.

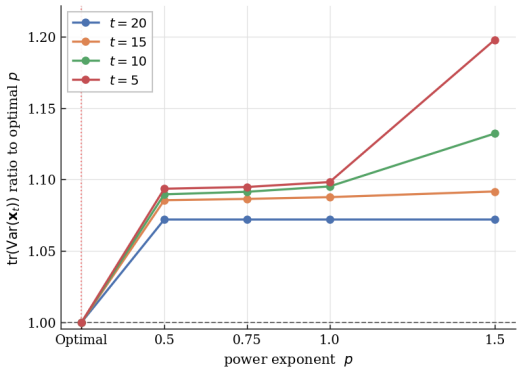
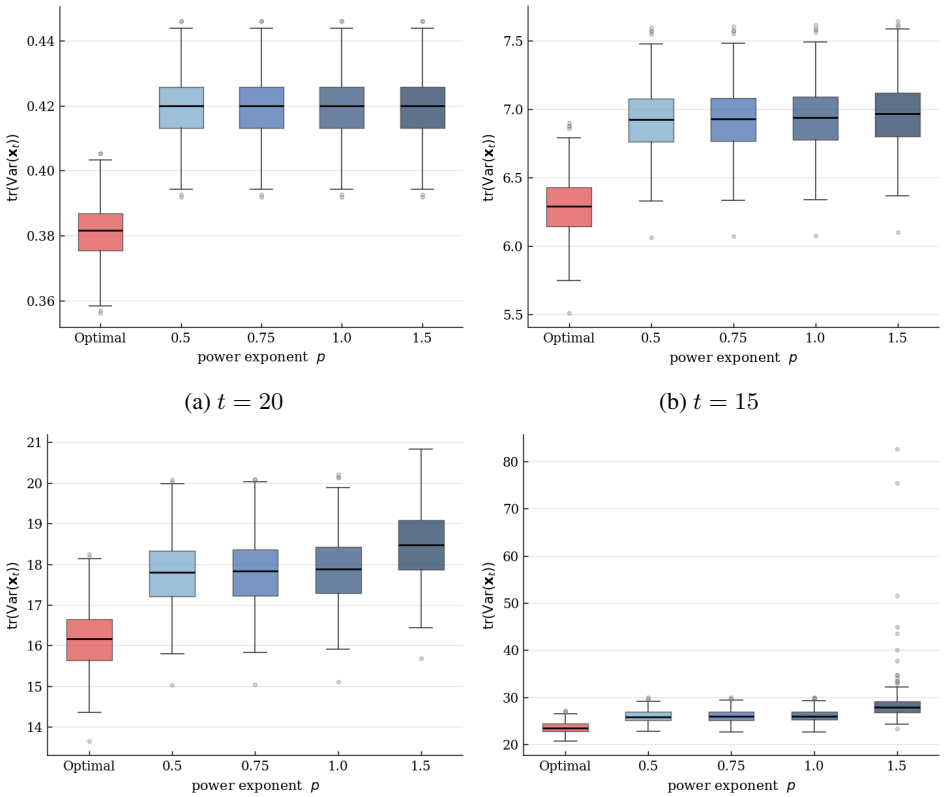


Figure 9: Effect of power exponent p on $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_t))$ ratio.



(a) $t = 20$ (b) $t = 15$

(c) $t = 10$ (d) $t = 5$

Figure 10: Per-sample distribution of $\mathrm{tr}(\mathrm{Var}(\boldsymbol{x}_t))$ by varying power exponent p at various timesteps t.

E ADDITIONAL EXPERIMENTS

E.1 CLASS CONDITIONAL IMAGE GENERATION

Figs. 11 and 12 present additional qualitative comparisons for class-conditional generation on ImageNet 256×256 and 512×512, respectively. Each figure shows CFG (top), BayesDiff+CFG (middle), and UFG (bottom) applied to the same initial noise.

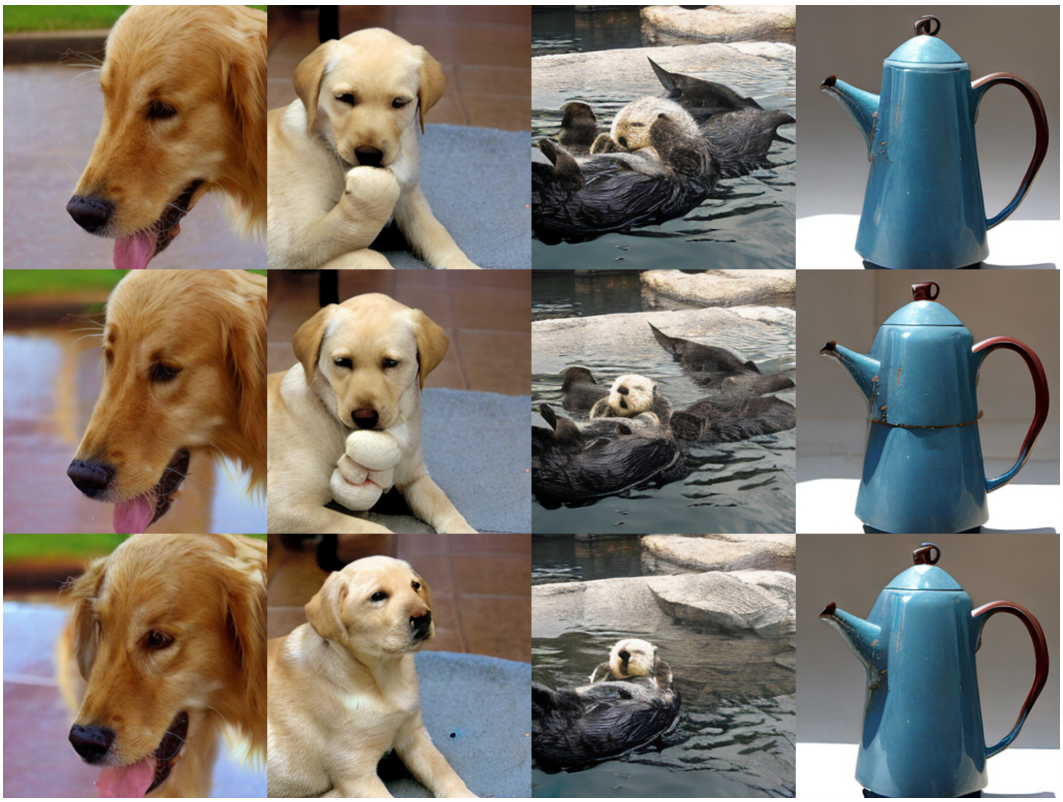


Figure 11: Comparison on conditional image generation for ImageNet 256×256. (Top row) CFG, (Middle row) BayesDiff+CFG, (Bottom row) UFG. Class labels: (i) Golden retriever (207), (ii) Labrador retriever (208), (iii) Otter (360), (iv) Coffeepot (505).

Results on DiT-XL/2 We use DiT-XL/2 with the DDPM sampler for 250 steps, a CFG scale of 1.5, and $M = 10$ Monte Carlo samples. Tab. 4 reports the results on ImageNet 256×256, with FID computed against the ADM reference batch. UFG-accel reaches the second-lowest FID (2.2492), below both CFG (2.2863) and BayesDiff+CFG (2.2738), and CADS has the lowest FID overall (1.9697). UFG (MC) sits slightly above CFG on FID but gives the best Precision among all methods (0.7128).

Table 4: Class-conditional results on ImageNet 256×256 with DiT-XL/2.

Method	FID ↓	Recall ↑	Precision ↑
AU	9.4203	0.5913	0.6431
BayesDiff+CFG	2.2738	0.6140	<u>0.7028</u>
UFG (MC) ✓	2.3552	0.5852	**0.7128**
UFG-accel ✓	<u>2.2492</u>	0.6124	<u>0.7028</u>
CFG	2.2863	0.6149	0.7020
CADS	**1.9697**	**0.6370**	0.6857
CFG++	2.7909	<u>0.6263</u>	0.6814



Figure 12: Comparison on conditional image generation for ImageNet 512×512. (Top row) CFG, (Middle row) BayesDiff+CFG, (Bottom row) UFG. Class labels: (i) Arctic fox (279). (ii) Lesser panda (387). (iii) Macaw (88). (iv) Otter (360).

E.2 TEXT-TO-IMAGE GENERATION

Fig. 13 presents additional qualitative comparisons for text-to-image generation using Z-Image, Figs. 14 and 15 extend the comparison to SD1.5, and Figs. 16 to 18 extend it further to SDXL.



Figure 13: Comparison of T2I results using Z-Image. (Top row) CFG, (Bottom row) UFG. Yellow boxes mark corresponding regions for comparison: (1) the finger-brake lever contact, (2) the fingers and lid of the teapot, and (3) the pins and springs inside the padlock. Prompts: (i) A hand gripping a bicycle handlebar with a brake lever, close-up, detailed knuckles. (ii) A hand pouring tea from a ceramic pot into a small cup, steam, detailed fingers. (iii) A brass padlock cut open showing pins and springs inside, macro.

Results on Stable Diffusion v1.5 UFG gives the best FID and VQAScore on SD1.5 among the compared methods (Tab. 5), with CLIP-score close behind BayesDiff+CFG.

Table 5: Comparison on quality measures of T2I results, using SD1.5 as a backbone.

Method	FID ↓	CLIP-score ↑	VQAScore ↑
AU	12.8314	0.3133	0.7699
BayesDiff+CFG	<u>12.5014</u>	**0.3154**	<u>0.7704</u>
CFG	13.1138	0.3133	0.7702
UFG (MC) ✓	**10.8647**	<u>0.3142</u>	**0.7705**



Figure 14: Comparison of T2I results using SD1.5. (Top row) CFG, (Middle row) BayesDiff+CFG, (Bottom row) UFG. Prompts: (i) An open book lies beside a pair of sunglasses. (ii) Across the blanket, a small stuffed rabbit is lying still. (iii) A ceramic mug sits on a wooden table with a soft shadow. (iv) An empty teacup sits on a folded napkin.



Figure 15: Comparison of T2I results using SD1.5. (Top row) CFG, (Middle row) BayesDiff+CFG (Bottom row) UFG. Prompts: (i) Medieval castle surrounded by water. (ii) A silver spoon rests beside a bowl of blueberries. (iii) Two coins and a leather wallet are placed on a shelf. (iv) On the desk, a pair of headphones rests next to a notebook.

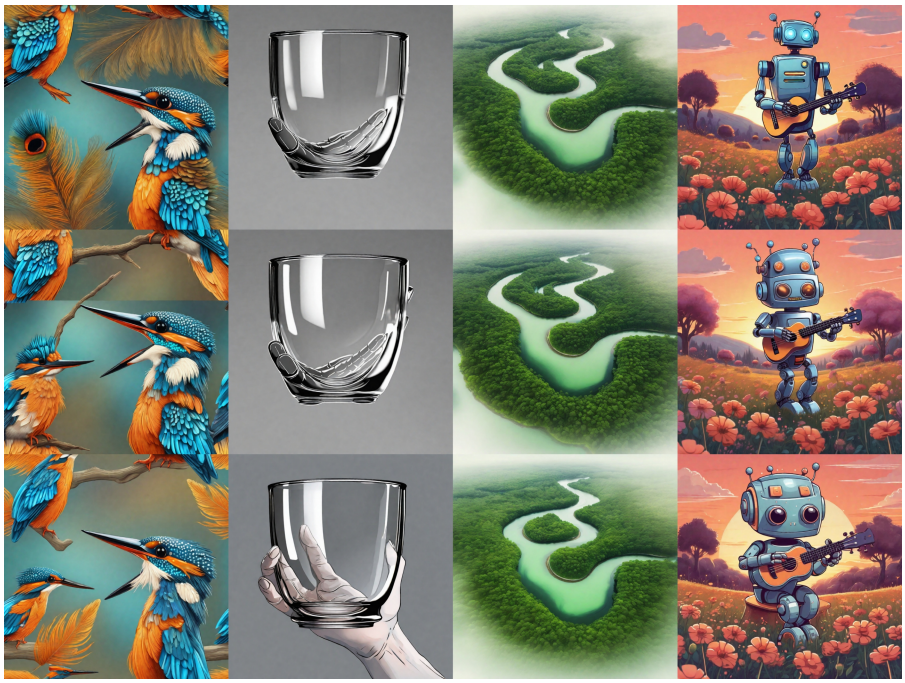


Figure 16: Comparison of T2I results using SDXL. (Top row) CFG, (Middle row) BayesDiff+CFG (Bottom row) UFG. Prompts: (i) Detailed feathers of a kingfisher bird, bright colors. (ii) A hand holding a glass cup, detailed fingers, transparent reflections. (iii) Aerial view of a winding river through a green forest. (iv) A cute, small retro-futuristic robot playing a ukulele in a field of flowers at sunset.



Figure 17: Comparison of T2I results using SDXL. (Top row) CFG, (Middle row) BayesDiff+CFG (Bottom row) UFG. Prompts: (i) A brass key hangs from a worn leather strap. (ii) A glass vase holds white lilies on a marble table. (iii) Near the window, a crystal bowl holds green grapes. (iv) On the shelf, a porcelain teapot stands beside two cups.

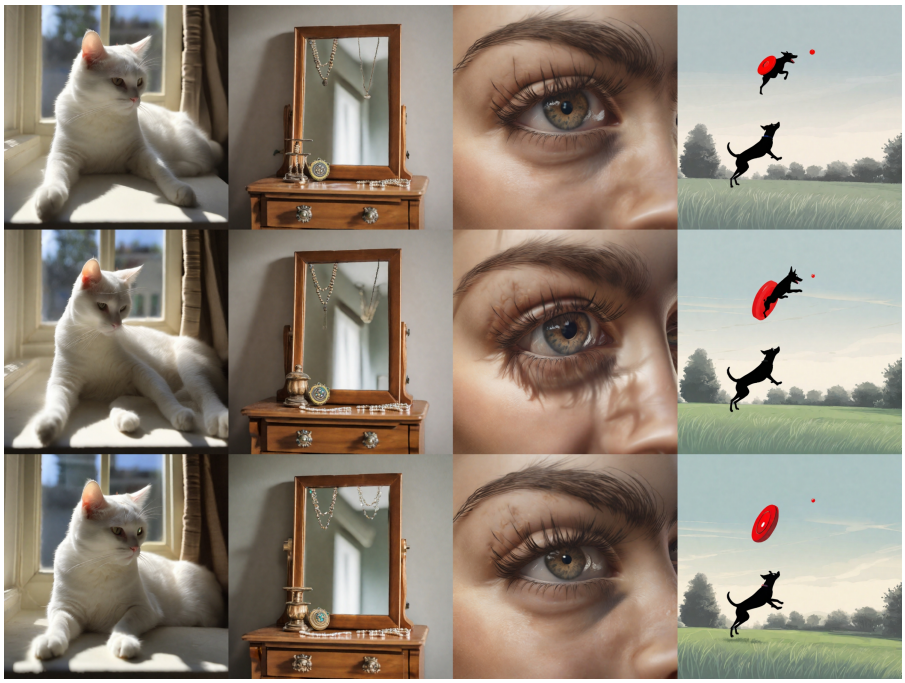


Figure 18: Comparison of T2I results using SDXL. (Top row) CFG, (Middle row) BayesDiff+CFG (Bottom row) UFG. Prompts: (i) The white cat is stretching across a sunlit windowsill. (ii) A small mirror reflects a necklace on a wooden dresser. (iii) Close-up portrait of a woman, detailed eyes, natural skin texture, fine hair strands, photorealistic. (iv) The black dog is jumping to catch a red frisbee.

E.3 VISUAL INTERPRETATION OF BAYESDIFF AND UFG

Figs. 19 to 22 present additional sampling trajectories and uncertainty estimates for ImageNet 256×256, following the same layout as Fig. 4. Across all examples, BayesDiff produces visible artifacts in regions where the uncertainty estimates are high, while UFG reduces both the artifacts and the corresponding uncertainty through pixel-wise guidance scaling.

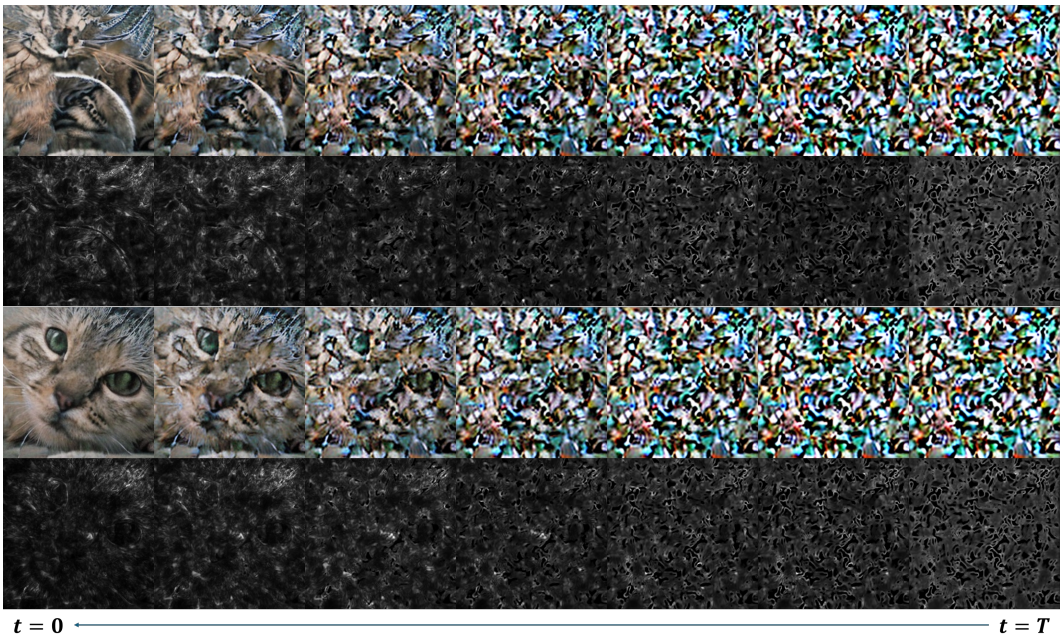


Figure 19: Comparison of BayesDiff(+CFG) and UFG on each step. Class: "Tabby cat" (281). (Top 2 rows) BayesDiff - images and uncertainty estimates, (Bottom 2 rows) UFG - images and uncertainty estimates.

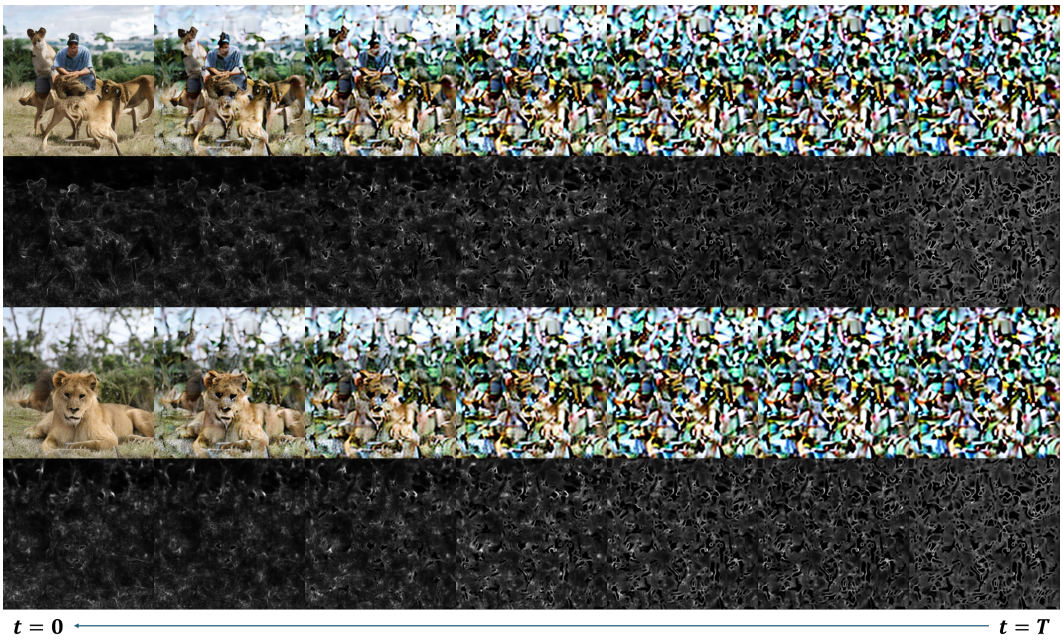


Figure 20: Comparison of BayesDiff(+CFG) and UFG on each step. Class: "Lion" (291). (Top 2 rows) BayesDiff - images and uncertainty estimates, (Bottom 2 rows) UFG - images and uncertainty estimates.

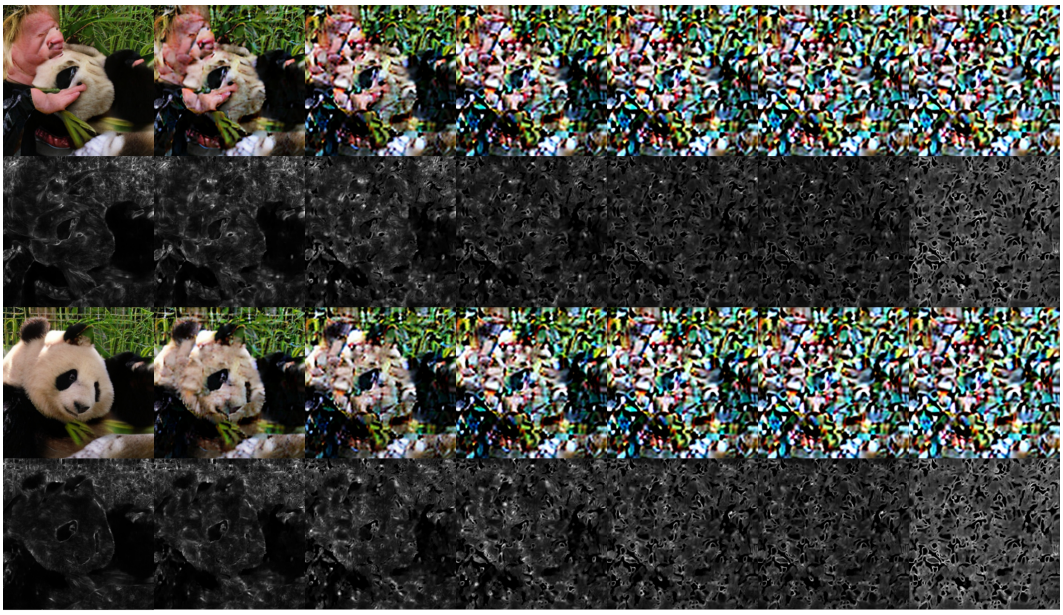


Figure 21: Comparison of BayesDiff(+CFG) and UFG on each step. Class: "Giant panda" (388). (Top 2 rows) BayesDiff - images and uncertainty estimates, (Bottom 2 rows) UFG - images and uncertainty estimates.

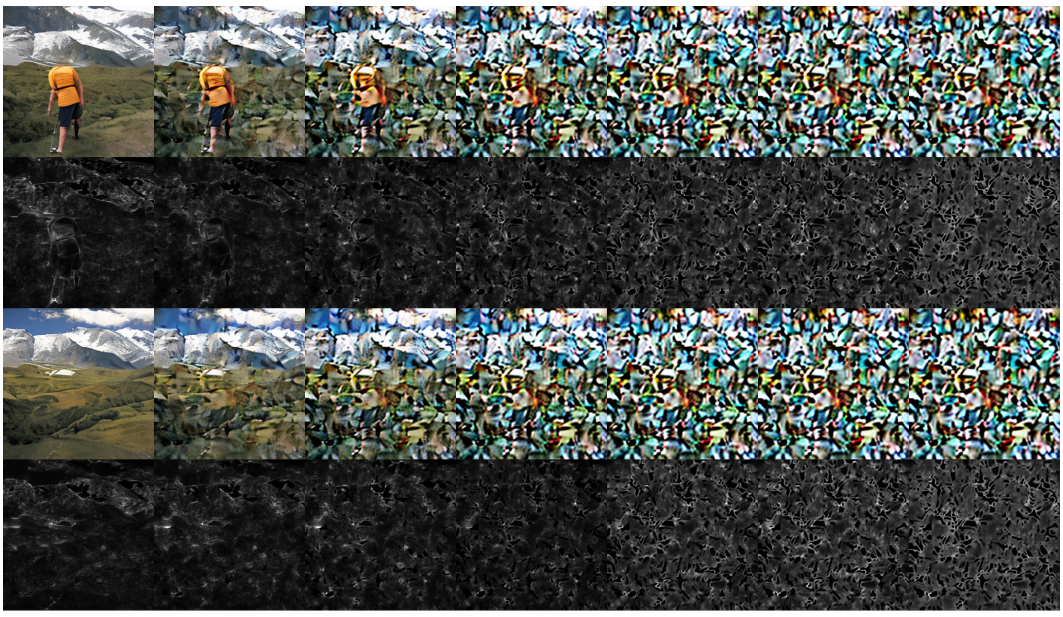


Figure 22: Comparison of BayesDiff(+CFG) and UFG on each step. Class: "Alp" (970). (Top 2 rows) BayesDiff - images and uncertainty estimates, (Bottom 2 rows) UFG - images and uncertainty estimates.

E.4 IMAGE EDITING VIA DDIM INVERSION

Setup We conduct real image editing experiments via DDIM inversion [10, 35], using SDXL. We first invert a real image using DDIM inversion with the conditioning prompt "a photograph of [source concept]." In the subsequent sampling stage, we replace [source concept] with [target concept] and use the modified prompt for generation. For example, we set [source concept] to "Cat" and [target concept] to "Black cat." We apply UFG during the sampling stage with $\lambda = 7.5$, and compare against CFG and BayesDiff with $\omega = 7.5$ under the same inversion and sampling configuration.

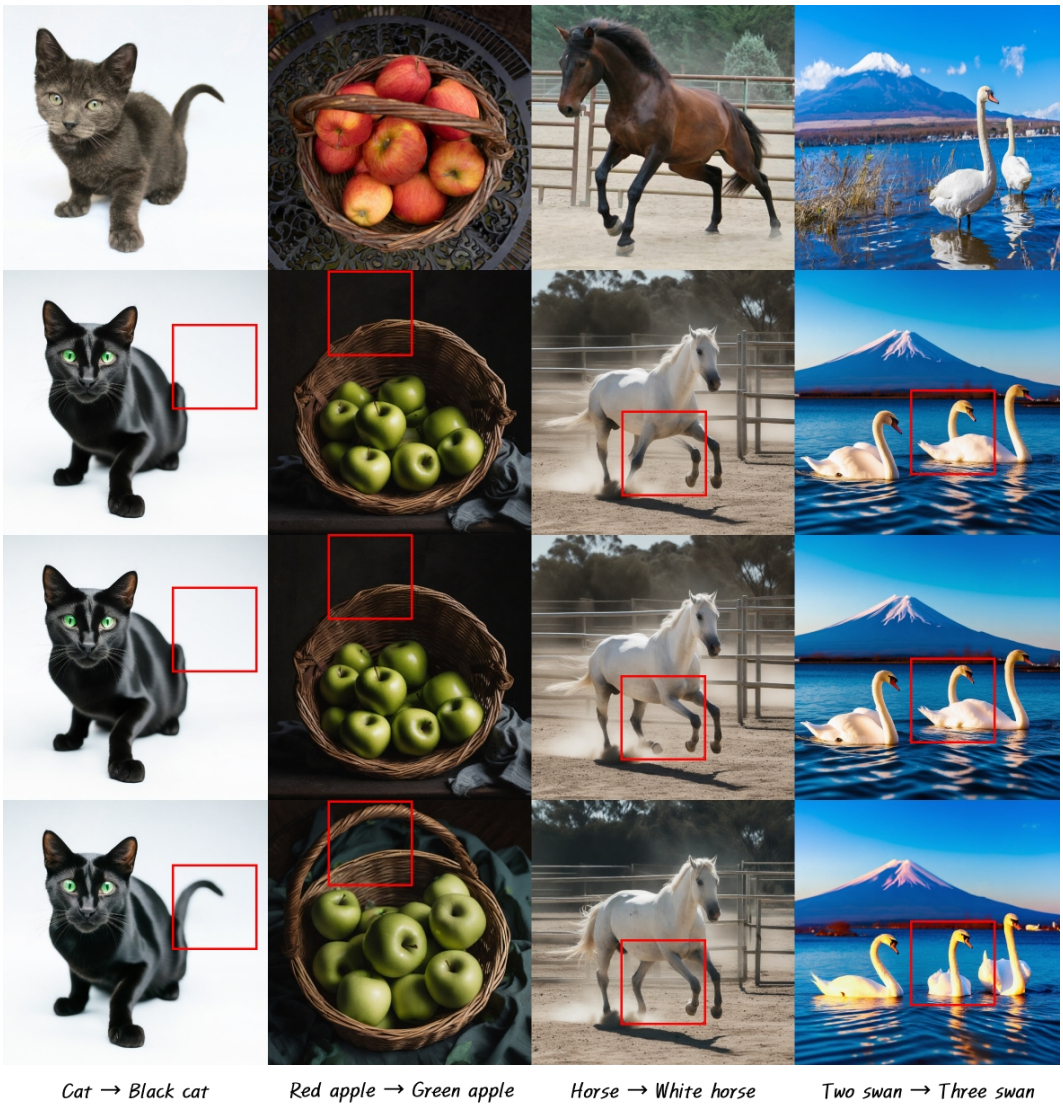


Figure 23: Comparison of real image editing via DDIM inversion using SDXL. (Top row) Source images. (Second row) CFG. (Third row) BayesDiff+CFG. (Bottom row) UFG.

Results Fig. 23 presents qualitative comparisons of the edited images. In the DDIM inversion setting, regions where the source concept is replaced by the target concept (e.g., the added third swan, the recolored horse legs) must be reconstructed under a new prompt while preserving the surrounding structure. These regions naturally incur higher uncertainty, as the starting noise was obtained by inverting the source image, not the target. As highlighted by the red bounding boxes, UFG produces more faithful edits precisely in such regions, confirming that the pixel-wise feedback mechanism generalizes beyond image generation to conditional editing scenarios.

Overview of the remaining subsections The remaining subsections look at UFG from several other angles. App. E.5 measures the added cost of Monte Carlo sampling and how the Taylor-based acceleration removes most of it, and App. E.6 checks whether the same pixel-wise scale still helps when it multiplies a different guidance update, APG, rather than CFG. Simpler alternatives get tested too. App. E.7 asks whether solving the variance-minimization problem beats a plain uncertainty-agnostic heuristic, and App. E.8 swaps the Laplace approximation for an ensemble to see whether UFG depends on that particular uncertainty estimator. App. E.9 then turns to more demanding, compositional prompts.

E.5 Computational cost

Setup We measured the sampling time and peak GPU memory of each method with one fresh process per method. SDXL runs at 1024×1024 with batch size 1, $M = 10$, and 50 DDIM steps on one NVIDIA RTX 4090 (24 GB). The time covers prompt encoding, sampling, and VAE decoding, and excludes loading the Laplace checkpoint. After 3 warmup images, we ran 3 repeats of 20 captions and report the mean, with peak memory taken as the maximum allocated.

Results Tab. 6 reports the cost per image on SDXL. The accelerated variants stay close to CFG (about $1.2\times$), while the sampled version is substantially slower (about $2.9\times$); the number of network evaluations per image follows the same pattern. Peak memory is essentially the same across all four methods, since the Laplace head and the Monte Carlo tensors are small next to the UNet, the text encoders, and the VAE.

Table 6: Sampling cost per image. SDXL: 1024^2, $M = 10$, batch size 1, 20 captions, one RTX 4090.

Model	Method	sec / image ↓	Memory (GB) ↓
SDXL	CFG	7.17	10.49
SDXL	UFG (MC) ✓	20.90	10.49
SDXL	UFG-accel ✓	8.79	10.49
SDXL	UFG-M (accel) ✓	8.77	10.49

E.6 Combining UFG with APG

Setup APG [31] modifies how the guidance update is applied. It splits the update into a part parallel to the conditional prediction and a part orthogonal to it, down-weights the parallel part, and adds rescaling and momentum. We replace the global guidance scale in APG by the pixel-wise scale γ_t of UFG-M. The scale multiplies the projected update as a whole, so that $\hat{x}_0 = x_0^c + \gamma_t \odot (\Delta_\perp + \eta\, \Delta_\parallel)$, where $\Delta_\perp$ and $\Delta_\parallel$ are the orthogonal and parallel parts of the guidance difference. With $\eta = 0$ only the orthogonal part remains. Everything else, including the APG parameters ($r = 0$, clamp off), is kept.

Results Adding the pixel-wise UFG-M scale to APG gives the best FID of the three on Z-Image (Tab. 7), at a small cost in VQAScore. CLIP-score is essentially unchanged.

Table 7: Combining UFG-M with APG on Z-Image.

Method	FID ↓	CLIP-score ↑	VQAScore ↑
CFG	14.2414	0.3080	0.8956
APG	<u>11.3009</u>	**0.3185**	**0.9029**
UFG-M + APG ✓	**10.2608**	<u>0.3180</u>	<u>0.8996</u>

E.7 Comparison with heuristic dynamic guidance

Setup We compare UFG-M with two simple dynamic guidance strategies. The first is a timestep-dependent CFG baseline (CFG (scheduled)) in which the guidance scale ω_t increases linearly from 0 to 2ω, following Wang et al. [39]. The second is uncertainty-normalized scaling (UFG-M (Simple)), which uses the simplified rule of App. D.3 with $p = 1/2$ and therefore adjusts the guidance scale by

uncertainty without solving the variance-minimization problem. All methods use U-ViT on ImageNet 256×256 with the same ω.

Results UFG-M (MC) gives the lowest FID of the four methods and ties UFG-M (Simple) for the highest Precision. Scheduled CFG has the highest Recall but the lowest Precision: it buys diversity with fidelity. UFG-M (Simple), which scales guidance by uncertainty without solving the variance-minimization problem, falls between CFG and UFG-M (MC) on FID and Recall, and matches it on Precision. Scaling by uncertainty already helps, and solving the variance-minimization problem adds a further gain on FID and Recall.

Table 8: Comparison with heuristic dynamic guidance on ImageNet 256×256.

Method	FID ↓	Recall ↑	Precision ↑
CFG	3.2458	0.5990	0.7129
CFG (scheduled)	2.9285	**0.6094**	0.7039
UFG-M (Simple) ✓	<u>2.9180</u>	0.6009	**0.7151**
UFG-M (MC) ✓	**2.8773**	<u>0.6012</u>	**0.7151**

E.8 UFG WITH AN ENSEMBLE-BASED UNCERTAINTY ESTIMATOR

Setup UFG is not specific to the Laplace approximation. The guidance scale γ_t needs only a set of samples $\{\boldsymbol{x}_t^{(m)}\}_{m=1}^M$ that represents the uncertainty of the denoising prediction, so any estimator that produces several plausible predictions can be plugged in. As an example we use an ensemble, following DECU [5]. DECU trains an ensemble over the conditional label embedding layer. We instead train an ensemble over the last layer of the noise prediction model, and let each member play the role of a posterior weight sample under the last-layer Laplace approximation. The member predictions go through the same procedure to form $\{\boldsymbol{x}_t^{(m)}\}_{m=1}^M$, and γ_t is computed as before.

Results Swapping the Laplace approximation for an ensemble (Tab. 9) keeps the FID gain over CFG. UFG (Ensemble) also improves on BayesDiff+CFG's FID, matches UFG (MC) on Recall, and edges it out on FID and Precision. UFG-M (Ensemble) falls behind UFG-M (MC) on FID and Recall, though it edges ahead on Precision, and its FID no longer beats BayesDiff+CFG.

Table 9: UFG with an ensemble-based uncertainty estimator on ImageNet 256×256.

Method	FID ↓	Recall ↑	Precision ↑
AU	5.7765	0.5795	0.6740
BayesDiff+CFG	2.9359	0.5840	0.7162
UFG (MC) ✓	2.9425	0.5849	0.7126
UFG (Ensemble) ✓	**2.8638**	0.5848	<u>0.7164</u>
CFG	3.2458	<u>0.5990</u>	0.7129
UFG-M (MC) ✓	<u>2.8773</u>	**0.6012**	0.7151
UFG-M (Ensemble) ✓	2.9693	0.5871	**0.7179**

E.9 HIGHLY COMPOSITIONAL PROMPTS

Setup The Laplace approximation is fit on a LAION-ART subset and evaluated on MSCOCO, so the main results already involve a shift in the data. To test compositional prompts more directly, we generated 500 highly compositional prompts with an LLM and generated images from them with each method on SDXL. An example is "a red fox sniffing at a lantern while a hedgehog watches from the side, set in a foggy harbor at dawn in cold overcast light". We report CLIP-score and VQAScore, and skip FID since it is unreliable with this few samples.

Results UFG gives the best CLIP-score and VQAScore on these prompts (Tab. 10), with BayesDiff+CFG second on both, AU last on CLIP-score, and CFG last on VQAScore. We attribute this to the same pixel-wise mechanism: more compositional prompts likely raise predictive uncertainty in more regions, giving the adaptive scale more room to help.

Table 10: T2I performance on highly compositional prompts.

Method	CLIP-score ↑	VQAScore ↑
CFG	0.3736	0.6000
AU	0.3734	0.6036
BayesDiff+CFG	<u>0.3747</u>	<u>0.6058</u>
UFG ✓	**0.3748**	**0.6060**