Classifier-Free Guidance is a Predictor-Corrector

Arwen Bradley*
Apple

Preetum Nakkiran*
Apple

Abstract

We investigate the theoretical foundations of classifier-free guidance (CFG). CFG is the dominant method of conditional sampling for text-to-image diffusion models, yet unlike other aspects of diffusion, it remains on shaky theoretical footing. In this paper, we first disprove common misconceptions, by showing that CFG interacts differently with DDPM [5] and DDIM [22], and neither sampler with CFG generates the gamma-powered distribution $p(x|c)^\gamma p(x)^{1-\gamma}$. Then, we clarify the behavior of CFG by showing that it is a kind of predictor-corrector method [24] that alternates between denoising and sharpening, which we call predictor-corrector guidance (PCG). We prove that in the SDE limit, CFG is actually equivalent to combining a DDIM predictor for the conditional distribution together with a Langevin dynamics corrector for a gamma-powered distribution (with a carefully chosen gamma). Our work thus provides a lens to theoretically understand CFG by embedding it in a broader design space of principled sampling methods.

1 Introduction

Classifier-free-guidance (CFG) has become an essential part of modern diffusion models, especially in text-to-image applications [2, 19, 14, 16]. CFG is intended to improve conditional sampling, e.g. generating images conditioned on a given class label or text prompt [4]. The traditional (non-CFG) way to do conditional sampling is to simply train a model for the conditional distribution $p(x \mid c)$, including the conditioning c as auxiliary input to the model. In the context of diffusion, this means training a model to approximate the conditional score $s(x, t, c) := \nabla_x \log p_t(x \mid c)$ at every noise level t, and sampling from this model via a standard diffusion sampler (e.g. DDPM). Interestingly, this standard way of conditioning usually does not perform well for diffusion models, for reasons that are unclear. In the text-to-image case for example, the generated samples tend to be visually incoherent and not faithful to the prompt, even for large-scale models [4, 19].

Guidance methods, such as CFG and its predecessor classifier guidance [21, 24, 1], are methods introduced to improve the quality of conditional samples. During training, CFG requires learning a model for both the unconditional and conditional scores ($\nabla_x \log p_t(x)$ and $\nabla_x \log p_t(x|c)$). Then, during sampling, CFG runs any standard diffusion sampler (like DDPM or DDIM), but replaces the true conditional scores with the "CFG scores"

$$\widetilde{s}(x, t, c) := \gamma \nabla_x \log p_t(x \mid c) + (1 - \gamma) \nabla \log p_t(x), \tag{1}$$

for some $\gamma > 0$. This turns out to produce much more coherent samples in practice, and so CFG is used in almost all modern text-to-image diffusion models [2]. A common intuition for why CFG works starts by observing that Equation (1) is the score of a *gamma-powered* distribution:

$$p_{t,\gamma}(x|c) := p_t(x)^{1-\gamma} p_t(x|c)^\gamma, \tag{2}$$

which is also proportional to $p_t(x)p_t(c|x)^\gamma$. Raising $p_t(c|x)$ to a power $\gamma > 1$ sharpens the classifier around its modes, thereby emphasizing the "best" exemplars of the given class or other conditioner at

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Figure 1: **CFG vs. PCG**. We prove that the DDPM variant of classifier-free guidance (top) is equivalent to a kind of predictor-corrector method (bottom), in the continuous limit. We call this latter method "predictor-corrector guidance" (PCG), defined in Section 3.1. The equivalence holds for all CFG guidance strengths γ, with corresponding PCG parameter $\gamma' = (2\gamma - 1)$, as given in Theorem 3. Samples from SDXL with prompt: "photograph of a cat eating sushi using chopsticks".

each noise level. Applying CFG — that is, running a standard sampler with the usual score replaced by the CFG score at each denoising step — is supposed to increase the influence of the conditioner on the final samples.

However, CFG does not inherit the theoretical correctness guarantees of standard diffusion, because the CFG scores do not necessarily correspond to a valid diffusion forward process. The fundamental issue (which is known, but still worth emphasizing) is that $p_{t,\gamma}(x|c)$ is not the same as the distribution obtained by applying a forward diffusion process to the gamma-powered data distribution $p_{0,\gamma}(x|c)$. That is, letting $N_t[p]$ denote the distribution produced by starting from a distribution p and running the diffusion forward process up to time t, we have

$$p_{t,\gamma}(x|c) := N_t[p_0(x|c)]^\gamma \cdot N_t[p_0(x)]^{1-\gamma} \neq N_t\left[p_0(x|c)^\gamma p_0(x)^{1-\gamma}\right].$$

Since the distributions $\{p_{t,\gamma}(x|c)\}_t$ *do not correspond to any known forward diffusion process*, we cannot properly interpret the CFG score (1) as a denoising direction; and using the CFG score in a sampling loop like DDPM or DDIM is no longer theoretically guaranteed to produce a sample from $p_{0,\gamma}(x|c)$ or any other known distribution. Although this flaw is known in theory (e.g. Du et al. [3], Karras et al. [7]), it is largely ignored in practice and in much of the literature. The theoretical foundations of CFG are thus unclear, and important questions remain open. Is there a principled way to think about why CFG works? And what does it even mean for CFG to "work" – what problem is CFG solving? We make progress towards understanding the foundations of CFG, and in the process we uncover several new aspects and connections to other methods.

1. First, we disprove common misconceptions about CFG by counterexample. We show that the DDPM and DDIM variants of CFG can generate different distributions, neither of which is the gamma-powered data distribution $p_0(x)^{1-\gamma}p_0(x|c)^\gamma$.

2. We define a family of methods called predictor-corrector guidance (PCG), as a natural way to approximately sample from gamma-powered distributions. PCG alternates between denoising steps and Langevin dynamics steps. Unlike typical predictor-corrector methods [24], in PCG the corrector operates on a different (sharper) distribution than the predictor.

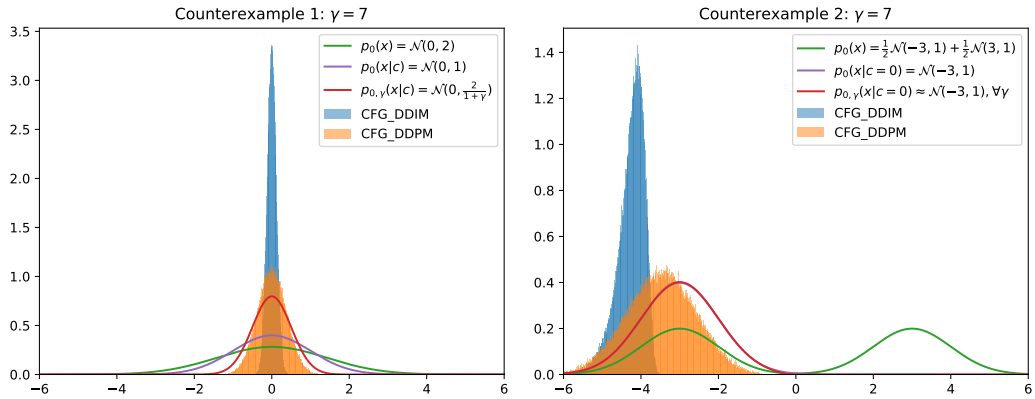

Figure 2: **Counterexamples: $\mathsf{CFG}_{\mathsf{DDIM}} \neq \mathsf{CFG}_{\mathsf{DDPM}} \neq$ gamma-powered.** $\mathsf{CFG}_{\mathsf{DDIM}}$ and $\mathsf{CFG}_{\mathsf{DDPM}}$ do not generate the same output distribution, even when using the same score function. Moreover, neither generated distribution is the gamma-powered distribution $p_{0,\gamma}(x|c)$. (Left) Counterexample 1 (section C.1): $\mathsf{CFG}_{\mathsf{DDIM}}$ yields a sharper distribution than $\mathsf{CFG}_{\mathsf{DDPM}}$, and both are sharper than $p_{0,\gamma}(x|c)$. (Right) Counterexample 2 (section C.2): Neither $\mathsf{CFG}_{\mathsf{DDIM}}$ nor $\mathsf{CFG}_{\mathsf{DDPM}}$ yield even a scaled version of the gamma-powered distribution $p_{0,\gamma}(x|c) = \mathcal{N}(-3, 1)$. The $\mathsf{CFG}_{\mathsf{DDPM}}$ distribution is mean-shifted relative to $p_{0,\gamma}(x|c)$. The $\mathsf{CFG}_{\mathsf{DDIM}}$ distribution is mean-shifted and not even Gaussian (note the asymmetrical shape).

3. We prove that in the continuous-time limit, CFG is equivalent to PCG with a careful choice of parameters. This gives a principled way to interpret CFG: it is implicitly an annealed Langevin dynamics.
4. For demonstration purposes, we implement the PCG sampler for Stable Diffusion XL and observe that it produces samples qualitatively similar to CFG, with guidance scales determined by our theory. Further, we explore the design axes exposed by the PCG framework, namely guidance strength and Langevin parameters, to clarify their respective effects.

Throughout this work we will use standard definitions of the diffusion process, CFG, and Langevin dynamics, which we provide in Appendix B.

2 Misconceptions about CFG

We first observe that the exact definition of CFG matters: specifically, the sampler with which it used. Without CFG, DDPM and DDIM generate equivalent distributions. However, we will prove that with CFG, DDPM and DDIM can generate different distributions, as follows:

Theorem 1 (DDIM $\neq$ DDPM; informal). *There exists a joint distribution $p(x, c)$ over inputs $x \in \mathbb{R}$ and conditioning $c \in \mathbb{R}$, such that the following holds. Consider generating a sample via CFG with conditioning $c = 0$, guidance-scale $\gamma \gg 0$, and using either DDPM or DDIM samplers. Then, the generated distributions will be approximately*

$$\widehat{p}_{\text{ddpm}} \approx \mathcal{N}(0, \gamma^{-1}); \quad \widehat{p}_{\text{ddim}} \approx \mathcal{N}(0, 2^{-\gamma}). \tag{3}$$

In particular, the DDIM variant of CFG is exponentially sharper than the DDPM variant.

Next, we disprove the misconception that CFG generates the gamma-powered distribution data:

Theorem 2 (CFG $\neq$ gamma-sharpening, informal). *There exists a joint distribution $p(x, c)$ and a $\gamma > 0$ such that neither $\mathsf{CFG}_{\mathsf{DDIM}}$ nor $\mathsf{CFG}_{\mathsf{DDPM}}$ produces the gamma-powered distribution $p_{0,\gamma}(x|c) \propto p_0(x)^{1-\gamma} p_0(x|c)^{\gamma}$.*

We prove both claims in Appendix C using simple Gaussian constructions. The main results are shown in Figure 2.

Algorithm 1: $\mathsf{PCG}_{\mathsf{DDIM}}$, theory. (see Algorithm 2 for practical implementation.)

Input: Conditioning c, guidance weight $\gamma \geq 0$
Constants: $\beta_t := \beta(t)$ from Song et al. [24]

$x_1 \sim \mathcal{N}(0, I)$
for $(t = 1 - \Delta t;\ t \geq 0;\ t \leftarrow t - \Delta t)$ **do**
  $s_{t+\Delta t} := \nabla \log p_{t+\Delta t}(x_{t+\Delta t}|c)$
  $x_t \leftarrow x_{t+\Delta t} + \frac{1}{2}\beta_t(x_{t+\Delta t} + s_{t+\Delta t})\Delta t$ ▷ DDIM step on $p_{t+\Delta t}(x + \Delta t|c)$
  $\varepsilon := \beta_t \Delta t$ ▷ Langevin step size
  for $k = 1, \ldots K$ **do**
    $\eta \sim \mathcal{N}(0, I_d)$
    $s_{t,\gamma} := (1-\gamma)\nabla \log p_t(x_t) + \gamma \nabla \log p_t(x_t|c)$
    $x_t \leftarrow x_t + \frac{\varepsilon}{2} s_{t,\gamma} + \sqrt{\varepsilon}\eta$ ▷ Langevin dynamics on $p_{t,\gamma}(x|c)$
  end
end
return x_0

3 CFG as a predictor-corrector

The previous sections illustrated the subtlety in understanding CFG. We can now state our main structural characterization, that CFG is equivalent to a kind of *predictor-corrector* method [24].

3.1 Predictor-Corrector Guidance

As a warm-up, suppose we actually wanted to sample from the gamma-powered distribution:

$$p_\gamma(x|c) \propto p(x)^{1-\gamma} p(x|c)^\gamma. \tag{4}$$

A natural strategy is to run Langevin dynamics w.r.t. p_γ. This is possible in theory because we can compute the score of p_γ from the known scores of $p(x)$ and $p(x \mid c)$:

$$\nabla_x \log p_\gamma(x \mid c) = (1-\gamma)\nabla_x \log p(x) + \gamma \nabla_x \log p(x \mid c). \tag{5}$$

However this won't work in practice, due to the well-known issue that vanilla Langevin dynamics has impractically slow mixing times for many distributions of interest [23]. The usual remedy for this is to use some kind of annealing, and the success of diffusion teaches us that the diffusion process defines a good annealing path [24, 3]. Combining these ideas yields an algorithm remarkably similar to the predictor-corrector methods introduced in Song et al. [24]. For example, consider the following diffusion-like iteration, starting from $x_T \sim \mathcal{N}(0, \sigma_T)$ at $t = T$. At timestep t,

1. Predictor: Take one diffusion denoising step (e.g. DDIM or DDPM) w.r.t. $p_t(x \mid c)$, using score $\nabla_x \log p_t(x \mid c)$, to move to time $t' = t - \Delta t$.
2. Corrector: Take one (or more) Langevin dynamics steps w.r.t. distribution $p_{t',\gamma}$, using score

$$\nabla_x \log p_{t',\gamma}(x \mid c) = (1-\gamma)\nabla_x \log p_{t'}(x) + \gamma \nabla_x \log p_{t'}(x \mid c).$$

It is reasonable to expect that running this iteration down to $t = 0$ will produce a sample from approximately $p_\gamma(x|c)$, since it can be thought of as annealed Langevin dynamics where the predictor is responsible for the annealing. We name this algorithm predictor-corrector guidance (PCG). Notably, PCG differs from the predictor-corrector algorithms in Song et al. [24] because our predictor and corrector operate w.r.t. *different* annealing distributions: the predictor tries to anneal along the set of distributions $\{p_t(x|c)\}_{t\in[0,1]}$, whereas the corrector anneals along the set $\{p_{t,\gamma}(x|c)\}_{t\in[0,1]}$. Remarkably, it turns out that for specific choices of the denoising predictor and Langevin step size, PCG with $K = 1$ is equivalent (in the SDE limit) to CFG, but with a different γ.

3.2 SDE limit of PCG

Consider the version of PCG defined in Algorithm 1, which uses DDIM as predictor and a particular LD on the gamma-powered distribution as corrector. We take $K = 1$, i.e. a single LD step per

iteration. Crucially, we set the LD step size such that the Langevin noise scale exactly matches the noise scale of a (hypothetical) DDPM step at the current time (similar to Du et al. [3]). In the limit as $\Delta t \to 0$, Algorithm 1 becomes the following SDE (see Appendix D):

$$dx = \underbrace{\Delta\mathsf{DDIM}(x,t)}_{\text{Predictor}} + \underbrace{\Delta\mathsf{LD}_{\mathsf{G}}(x,t,\gamma)}_{\text{Corrector}} =: \Delta\mathsf{PCG}_{\mathsf{DDIM}}(x,t,\gamma), \tag{6}$$

$$\text{where } \Delta\mathsf{DDIM}(x,t) = -\frac{1}{2}\beta_t(x + \nabla \log p_t(x|c))dt$$

$$\Delta\mathsf{LD}_{\mathsf{G}}(x,t,\gamma) = -\frac{1}{2}\beta_t\Big((1-\gamma)\nabla \log p_t(x) + \gamma\nabla \log p_t(x|c)\Big)dt + \sqrt{\beta_t}d\overline{w}.$$

Above, $\Delta\mathsf{DDIM}(x,t)$ is the *differential* of the DDIM ODE (11), i.e. the ODE can be written as $dx = \Delta\mathsf{DDIM}(x,t)$. And $\Delta\mathsf{LD}_{\mathsf{G}}(x,t,\gamma)$, where G stands for "guidance", is the limit as $\Delta t \to 0$ of the Langevin dynamics step in PCG, which behaves like a differential of LD (see Appendix D).

We can now show that the PCG SDE (6) matches CFG, but with a different γ. In the statement, $\Delta\mathsf{CFG}_{\mathsf{DDPM}}(x,t,\gamma)$ denotes the differential of the $\mathsf{CFG}_{\mathsf{DDPM}}$ SDE (12), similar to the notation above. This result is trivial to prove using our definitions, but the statement itself appears to be novel.

Theorem 3 (CFG is predictor-corrector)**.** *In the SDE limit, CFG is equivalent to a predictor-corrector. That is, the following differentials are equal:*

$$\Delta\mathsf{CFG}_{\mathsf{DDPM}}(x,t,\gamma) = \Delta\mathsf{DDIM}(x,t) + \Delta\mathsf{LD}_{\mathsf{G}}(x,t,2\gamma-1) =: \Delta\mathsf{PCG}_{\mathsf{DDIM}}(x,t,2\gamma-1) \tag{7}$$

Notably, the guidance scales of CFG and the above Langevin dynamics are not identical.

Proof.

$$\begin{aligned}
\Delta\mathsf{PCG}_{\mathsf{DDIM}}(x,t,\gamma) &= \Delta\mathsf{DDIM}(x,t) + \Delta\mathsf{LD}_{\mathsf{G}}(x,t,\gamma) \\
&= -\frac{1}{2}\beta_t(x + (1-\gamma)\nabla \log p_t(x) + (1+\gamma)\nabla \log p_t(x|c))dt + \sqrt{\beta_t}d\overline{w} \\
&= -\frac{1}{2}\beta_t x\Delta t - \beta_t\nabla_x \log p_{t,\gamma'}(x|c)\Delta t + \sqrt{\beta_t}d\overline{w}, \quad \gamma' := \frac{\gamma}{2} + \frac{1}{2} \\
&= \Delta\mathsf{CFG}_{\mathsf{DDPM}}(x,t,\gamma')
\end{aligned}$$

□

As an aside, taking $\gamma = 1$ in Theorem 3 recovers the standard fact that DDPM is equivalent, in the limit, to DDIM interleaved with LD (e.g. Karras et al. [6]). Because for $\gamma = 1$, $\mathsf{CFG}_{\mathsf{DDPM}}$ is just DDPM, so Theorem 3 reduces to: $\Delta\mathsf{DDPM}(x,t) = \Delta\mathsf{DDIM}(x,t) + \Delta\mathsf{LD}_{\mathsf{G}}(x,t,1)$. This fact, that in the non-CFG case Langevin dynamics is equivalent to iteratively noising-then-denoising, has been used implicitly or explicitly in a number of prior works. For example, Karras et al. [6] use a "churn" operation in their stochastic sampler, and Lugmayr et al. [11] incorporate a conceptually similar noise-then-denoise step in their inpainting pipeline.

Experiments. In Appendix E, we include several examples running predictor-corrector guidance on Stable Diffusion XL [16], as a sanity-check for our theory.

4 Discussion: Generalization and CFG

There have been many recent works toward understanding CFG. To better situate our work, it helps to first discuss the overall research agenda. We want to study the question of why CFG helps in practice: specifically, why it improves both image quality and prompt adherence, compared to conditional sampling. We can approach this question by applying a standard generalization decomposition. Let $p(x|c)$ be the "ground truth" population distribution; let $p^*_\gamma(x|c)$ be the distribution generated by the ideal CFG sampler, which exactly solves the CFG reverse SDE for the ground-truth scores (note that at $\gamma = 1$, $p^*_1(x|c) = p(x|c)$); and let $\widehat{p}_\gamma(x|c)$ denote the distribution of the real CFG sampler, with learnt scores and finite discretization. Now, for any image distribution q, let $\mathsf{PerceivedQuality}[q] \in \mathbb{R}$ denote a measure of perceived sample quality of this distribution to humans. We cannot mathematically specify this notion of quality, but we will assume it exists for

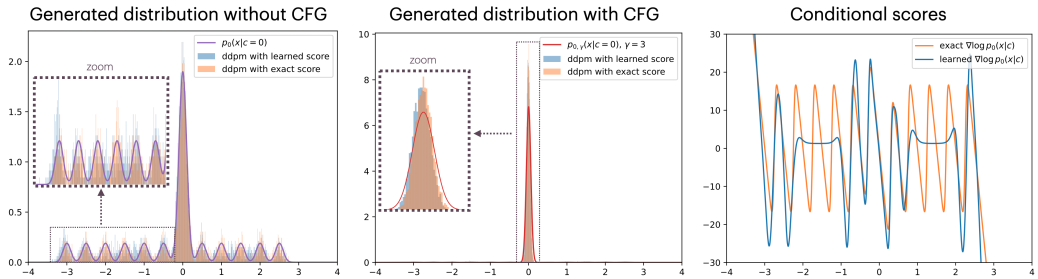

Figure 3: **An example where guidance benefits generalization.** Suppose that the conditional distribution for $c = 0$ is a a GMM with a dominant cluster, as shown in purple, and the unconditional distribution is uniform (details in Appendix C.6). We sample with DDPM using exact scores vs. scores learned by training a small MLP with early stopping. The scores are learned more accurately near the dominant cluster. (Left) For conditional sampling (no guidance), DDPM is expected to sample from the conditional distribution (purple curve). However, DDPM-with-learned-scores (blue) samples less accurately than DDPM-with-exact-scores (orange) away from the dominant cluster (where the learned scores are inaccurate) (note the prevalence of blue samples in low-probability regions). (Center) With guidance $\gamma = 3$, $p_{0,\gamma}(x|c = 0)$ (red) and both samplers concentrate around the dominant cluster (where the learned scores are accurate), reducing the generalization gap between the learned and exact models. (Right) Exact vs. learned condition scores $\nabla \log p(x|c = 0)$.

analysis. Notably, PerceivedQuality is *not* a measurement of how close a distribution is to the ground-truth $p(x|c)$ — it is possible for a generated distribution to appear even "higher quality" than the ground-truth, for example. We can now decompose:

$$\underbrace{\mathsf{PerceivedQuality}[\widehat{p}_\gamma]}_{\text{Real CFG}} = \underbrace{\mathsf{PerceivedQuality}[p^*_\gamma]}_{\text{Ideal CFG}} - \underbrace{\left(\mathsf{PerceivedQuality}[p^*_\gamma] - \mathsf{PerceivedQuality}[\widehat{p}_\gamma]\right)}_{\text{Generalization Gap}}. \tag{8}$$

Therefore, if the LHS increases with γ, it must be because at least one of the two occurs:

1. The ideal CFG sampler improves in quality with increasing γ. That is, CFG distorts the population distribution in a favorable way (e.g. by sharpening it, or otherwise).
2. The generalization gap decreases with increasing γ. That is, CFG has a type of regularization effect, bringing population and empirical processes closer.

In fact, it is likely that both occur. The original motivation for CG and CFG involved the first effect: CFG was intended to produce "lower-temperature" samples from a sharpened population distribution [1, 4]. This is particularly relevant if the model is trained on poor-quality datasets (e.g. cluttered images from the web), so we want to use guidance to sample from a higher-quality distribution (e.g. images of an isolated subject). On the other hand, recent studies have given evidence for the second effect. For example, Karras et al. [7] argues that unguided diffusion sampling produces "outliers," which are avoided when using guidance — this can be thought of as guidance reducing the generalization gap, rather than improving the ideal sampling distribution. Another interpretation of the second effect is that guidance could enforce a good inductive bias: it "simplifies" the family of possible output distributions in some sense, and thus simplifies the learning problem, reducing the generalization gap. Figure 3 shows a example where this occurs. Finally, this generalization decomposition applies to any intervention to the SDE, not just increasing guidance strength. For example, increasing the Langevin steps in PCG (parameter K) also shrinks the generalization gap, since it reduces the discretization error.

In this framework, our work makes progress towards understanding both terms on the RHS of Equation 8, in different ways. For the first term, we identify structural properties of ideal CFG, by showing that p^*_γ can be equivalently generated by a standard technique (an annealed Langevin dynamics). For the second term, the PCG framework highlights the ways in which errors in the

learned score can contribute to a generalization gap, during both the denoising step and the LD step (the latter would move toward an inaccurate steady-state distribution).

4.1 Open Questions and Limitations

In addition to the above, there are a number of other questions left open by our work. First, we study only the stochastic variant of CFG (i.e. $\mathsf{CFG}_{\mathsf{DDPM}}$), and it is not clear how to adapt our analysis to the more commonly used deterministic variant ($\mathsf{CFG}_{\mathsf{DDIM}}$). This is subtle because the two CFG variants can behave very differently in theory, but appear to behave similarly in practice. It is thus open to identify plausible theoretical conditions which explain this similarity[1]; we give a suggestive experiment in Figure 4. More broadly, it is open to find explicit characterizations of CFG's output distribution, in terms of the original $p(x)$ and $p(x|c)$ — although it is possible tractable expressions do not exist.

Finally, we presented PCG primarily as a tool to understand CFG, not as a practical algorithm in itself. Nevertheless, the PCG framework outlines a broad family of guided samplers, which may be promising to explore in practice. For example, the predictor can be any diffusion denoiser, including CFG itself. The corrector can operate on any distribution with a known score, including compositional distributions as in Du et al. [3], or any other distribution that might help sharpen or otherwise improve on the conditional distribution. Finally, the number of Langevin steps could be adapted to the timestep, similar to Kynkäänniemi et al. [8], or alternative samplers could be considered [3, 13, 12].

5 Conclusion

In this paper, we have shown that while CFG is not a diffusion sampler on the gamma-powered data distribution $p_0(x)^{1-\gamma}p_0(x|c)^{\gamma}$, it can be understood as a particular kind of predictor-corrector, where the predictor is a DDIM denoiser, and the corrector at each step t is one step of Langevin dynamics on the gamma-powered noisy distribution $p_t(x)^{1-\gamma'}p_t(x|c)^{\gamma'}$, with $\gamma' = (2\gamma - 1)$. Although Song et al. [24]'s Predictor-Corrector algorithm has not been widely adopted in practice, perhaps due to its computation expense relative to samplers like DPM++ [10], it turns out to provide a lens to understand the unreasonable practical success of CFG. On a practical note, PCG encompasses a rich design space of possible predictors and correctors for future exploration, that may help improve the prompt-alignment, diversity, and quality of diffusion generation.

[1] Curiously, $\mathsf{CFG}_{\mathsf{DDIM}}$ is the correct probability-flow ODE for $\mathsf{CFG}_{\mathsf{DDPM}}$ if and only if the true intermediate distribution at time t is $p_{t,\gamma}$. However we know this is not the true distribution in general, from Section 2.

References

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

[2] S. Dieleman. Guidance: a cheat code for diffusion models, 2022. URL https://benanne.github.io/2022/05/26/guidance.html.

[3] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. S. Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pages 8489–8510. PMLR, 2023.

[4] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

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

[7] T. Karras, M. Aittala, T. Kynkäänniemi, J. Lehtinen, T. Aila, and S. Laine. Guiding a diffusion model with a bad version of itself. *arXiv preprint arXiv:2406.02507*, 2024.

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

[9] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. 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.

[10] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022.

[11] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

[12] Y.-A. Ma, T. Chen, and E. Fox. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

[13] R. M. Neal. Mcmc using hamiltonian dynamics. *arXiv preprint arXiv:1206.1901*, 2012.

[14] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[15] G. Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

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

[17] C. P. Robert, G. Casella, and G. Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[18] G. O. Roberts and R. L. Tweedie. Exponential convergence of langevin distributions and their discrete approximations. 1996.

[19] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[20] P. J. Rossky, J. D. Doll, and H. L. Friedman. Brownian dynamics as smart monte carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978.

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

[22] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

[23] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[24] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. URL https://arxiv.org/pdf/2011.13456.pdf.

A Appendix

B Preliminaries

We adopt the continuous-time stochastic differential equation (SDE) formalism of diffusion from Song et al. [24]. These continuous-time results can be translated to discrete-time algorithms; we give explicit algorithm descriptions for our experiments.

B.1 Diffusion Samplers

Forward diffusion processes start with a conditional data distribution $p_0(x|c)$ and gradually corrupt it with Gaussian noise, with $p_t(x|c)$ denoting the noisy distribution at time t. The forward diffusion runs up to a time T large enough that p_T is approximately pure noise. To sample from the data distribution, we first sample from the Gaussian distribution p_T and then run the diffusion process in reverse (which requires an estimate of the score, usually learned by a neural network). A variety of samplers have been developed to perform this reversal. DDPM [5] and DDIM [22] are standard samplers that correspond to discretizations of a reverse-SDE and reverse-ODE, respectively. Due to this correspondence, we refer to the reverse-SDE as DDPM and the reverse-ODE as DDIM for short. We will mainly consider the *variance-preserving* (VP) diffusion process from Ho et al. [5], although most of our discussion applies equally to other settings (such as variance-exploding). The forward process, reverse-SDE, and equivalent reverse-ODE for the VP conditional diffusion are [24]

$$\text{Forward SDE}: dx = -\frac{1}{2}\beta_t x dt + \sqrt{\beta_t} dw. \tag{9}$$

$$\text{DDPM SDE}: \quad dx = -\frac{1}{2}\beta_t x\, dt - \beta_t \nabla_x \log p_t(x|c) dt + \sqrt{\beta_t} d\overline{w} \tag{10}$$

$$\text{DDIM ODE}: \quad dx = -\frac{1}{2}\beta_t x\, dt - \frac{1}{2}\beta_t \nabla_x \log p_t(x|c) dt. \tag{11}$$

The unconditional version of each sampler simply replaces $p_t(x|c)$ with $p_t(x)$. Note that the *score* $\nabla_x \log p_t(x|c)$ appears in both (10) and (11). Intuitively, the score points in a direction toward higher probability, and so it helps to reverse the forward diffusion process. The score is unknown in general, but can be learned via standard diffusion training methods.

B.2 Classifier-Free Guidance

CFG replaces the usual conditional score $\nabla_x \log p_t(x|c)$ in (10) or (11) at each timestep t with the alternative score $\nabla_x \log p_{t,\gamma}(x|c)$. In SDE form, the CFG updates are

$$\text{CFG}_{\text{DDPM}}: \quad dx = -\frac{1}{2}\beta_t x\, dt - \beta_t \nabla_x \log p_{t,\gamma}(x|c) dt + \sqrt{\beta_t} d\overline{w} \tag{12}$$

$$\text{CFG}_{\text{DDIM}}: \quad dx = -\frac{1}{2}\beta_t x\, dt - \frac{1}{2}\beta_t \nabla \log p_{t,\gamma}(x|c) dt, \tag{13}$$

$$\text{where } \nabla_x \log p_{t,\gamma}(x|c) = (1-\gamma)\nabla_x \log p_t(x) + \gamma \nabla_x \log p_t(x|c).$$

B.3 Langevin Dynamics

Langevin dynamics [20, 15] is another sampling method, which starts from an arbitrary initial distribution and iteratively transforms it into a desired one. Langevin dynamics (LD) is given by the

following SDE [17]

$$dx = \frac{\varepsilon}{2}\nabla \log \rho(x)dt + \sqrt{\varepsilon}dw. \tag{14}$$

LD converges (under some assumptions) to the steady-state $\rho(x)$ [18]. That is, letting $\rho_s(x)$ denote the solution of LD at time s, we have $\lim_{s\to\infty} \rho_s(x) = \rho(x)$. Similar to diffusion sampling, LD requires the score of the desired distribution ρ (or a learned estimate of it).

C 1D Gaussian Counterexamples

C.1 Counterexample 1

We first present a setting that allows us to *exactly* solve the ODE and SDE dynamics of CFG in closed-form, and hence to find the exact distribution sampled by running CFG. This would be intractable in general, but it is possible for a specific problem, as follows.

Consider the setting where $p_0(x)$ and $p_0(x|c=0)$ are both zero-mean Gaussians, but with different variances. Specifically, (x_0, c) are jointly Gaussian, with $p(c) = \mathcal{N}(0,1)$, $p_0(x|c) = c + \mathcal{N}(0,1)$. Therefore

$$\begin{aligned} p_0(x) &= \mathcal{N}(0,2) \\ p_0(x|c=0) &= \mathcal{N}(0,1) \\ p_{0,\gamma}(x|c=0) &= \mathcal{N}(0, \frac{2}{\gamma+1}) \end{aligned} \tag{15}$$

For this problem, we can solve $\mathsf{CFG}_{\mathsf{DDIM}}$ (13) and $\mathsf{CFG}_{\mathsf{DDPM}}$ (12) analytically; that is, we solve initial-value problems for the reversed dynamics to find the sampled distribution of $\widehat{x}_t$ in terms of the initial-value x_T. Applying these results to $t = 0$ and averaging over the known Gaussian distribution of x_T gives the exact distribution of $\widehat{x}_0$ that CFG samples. The full derivation is in Appendix C.3. The final CFG-sampled distributions are:

$$\mathsf{CFG}_{\mathsf{DDPM}}: \quad \widehat{x}_0 \sim \mathcal{N}\left(0, \frac{2 - 2^{2-2\gamma}}{2\gamma - 1}\right) \tag{16}$$

$$\mathsf{CFG}_{\mathsf{DDIM}}: \quad \widehat{x}_0 \sim \mathcal{N}\left(0, 2^{1-\gamma}\right). \tag{17}$$

This shows that for any $\gamma > 1$, the $\mathsf{CFG}_{\mathsf{DDIM}}$ distribution is sharper than the $\mathsf{CFG}_{\mathsf{DDPM}}$ distribution, and both are sharper than the gamma-powered distribution $p_{0,\gamma}(x|c=0)$. (Even though the distributions all have the same mean, their different variances make them distinct.) In fact, for $\gamma \gg 1$, the variance of DDPM-CFG is approximately $\frac{2}{2\gamma-1}$, which is about twice the variance of $p_{0,\gamma}(x|c=0)$. In Figure 2, we compare the $\mathsf{CFG}_{\mathsf{DDIM}}$ and $\mathsf{CFG}_{\mathsf{DDPM}}$ distributions – sampled using an exact denoiser (see Appendix C.8) within DDIM/DDPM sampling loops – to the unconditional, conditional, and gamma-powered distributions.

C.2 Counterexample 2

In the above counterexample, the $\mathsf{CFG}_{\mathsf{DDIM}}$, $\mathsf{CFG}_{\mathsf{DDPM}}$, and gamma-powered distributions had different variances but the same Gaussian form, so one might wonder whether the distributions differ only by a scale factor in general. This is not the case, as we can see in a different counterexample that reveals greater qualitative differences, in particular a symmetry-breaking behavior of CFG.

In Counterexample 2, the unconditional distribution is a Gaussian mixture with two clusters with equal weights and variances, and means at $\pm\mu$.

$$\begin{aligned} c \in \{0,1\}, \quad p(c=0) &= \frac{1}{2} \\ p_0(x_0|c=0) = \mathcal{N}(-\mu,1), \quad p_0(x_0|c=1) &= \mathcal{N}(\mu,1) \\ p_0(x_0) &= \frac{1}{2}p_0(x_0|c=0) + \frac{1}{2}p_0(x_0|c=1) \end{aligned} \tag{18}$$

If the means are sufficiently separated ($\mu \gg 1$), then the gamma-powered distribution for $\gamma \geq 1$ is approximately equal to the conditional distribution, i.e. $p_{0,\gamma}(x|c) \approx p_0(x|c)$, due to the near-zero-probability valley between the conditional densities (see Appendix C.4). However, for sufficiently

high noise the clusters begin to merge, and $p_{t,\gamma}(x|c) \neq p_t(x|c)$. In particular, $p_{0,\gamma}(x|c)$ is approximately Gaussian with mean $\pm\mu$, but $p_{t,\gamma}(x|c) \neq p_t(x|c)$ is not. Although we cannot solve the CFG ODE and SDE in this case, we can empirically sample the $\mathsf{CFG}_{\mathsf{DDIM}}$ and $\mathsf{CFG}_{\mathsf{DDPM}}$ distributions using an exact denoiser and compare them to the gamma-powered distribution. In particular, we see that neither $\mathsf{CFG}_{\mathsf{DDIM}}$ nor $\mathsf{CFG}_{\mathsf{DDPM}}$ is Gaussian with mean $\pm\mu$, hence neither is a scaled version of the gamma-powered distribution. The results are shown in Figure 2.

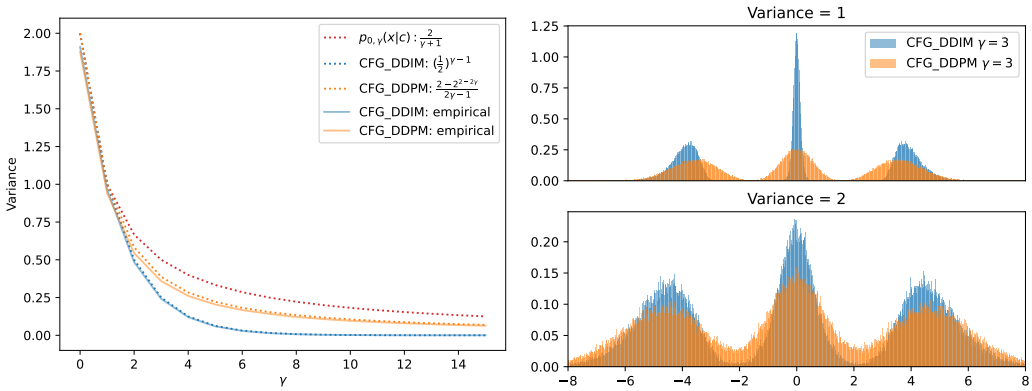

Figure 4: (Left) For Counterexample 1 (section C.1), we plot the empirical and theoretical variance of the gamma-powered, $\mathsf{CFG}_{\mathsf{DDIM}}$, and $\mathsf{CFG}_{\mathsf{DDPM}}$ distributions, over a range of values of γ. The theoretical predictions are given by equations (17) and (16), and the empirical distributions are sampled using an exact denoiser. This verifies the theoretical predictions and illustrates the decreasing variance from $p_{0,\gamma}$ to $\mathsf{CFG}_{\mathsf{DDPM}}$ to $\mathsf{CFG}_{\mathsf{DDIM}}$. (Right) For counterexample 3 (section C.5 with different choices of variance ($\sigma = 1$ and $\sigma = 2$), we compare $\mathsf{CFG}_{\mathsf{DDIM}}$ and $\mathsf{CFG}_{\mathsf{DDPM}}$. Increasing the variance makes the two CFG samplers more similar. Also note that the $\mathsf{CFG}_{\mathsf{DDIM}}$ distribution is symmetric around the center cluster, but asymmetric around the side clusters. This experiment suggests that multiple clusters and greater overlap between classes can help symmetrize and reduce the difference between $\mathsf{CFG}_{\mathsf{DDIM}}$ and $\mathsf{CFG}_{\mathsf{DDPM}}$

C.3 Counterexample 1 Detail

Counterexample 1 (equation 15) has

$$p_0(x) \sim \mathcal{N}(0, 2)$$
$$p_0(x|c = 0) \sim \mathcal{N}(0, 1).$$

The γ-powered distribution is

$$\begin{aligned} p_{0,\gamma}(x|c = 0) &= p_0(x|c)^\gamma p_{c=0}(x)^{1-\gamma} \\ &\propto e^{-\frac{\gamma x^2}{2}} e^{-\frac{(1-\gamma)x^2}{4}} = e^{-\frac{(\gamma+1)x^2}{4}} \\ &\sim \mathcal{N}(0, \frac{2}{\gamma+1}). \end{aligned}$$

We consider the simple variance-exploding diffusion defined by the SDE

$$dx = \sqrt{t} dw.$$

The DDIM sampler is a discretization of the reverse ODE

$$\frac{dx}{dt} = -\frac{1}{2} \nabla_x \log p_t(x),$$

and the DDPM sampler is a discretization of the reverse SDE

$$dx = -\nabla_x \log p_{t,\gamma}(x) dt + d\overline{w}.$$

For $\mathsf{CFG}_{\mathsf{DDIM}}$ or $\mathsf{CFG}_{\mathsf{DDPM}}$, we replace the score with CFG score $\nabla_x \log p_{t,\gamma}(x)$.

During training we run the forward process until some time $t = T$, at which point we assume it is fully-noised, so that approximately

$$p_T(x|c = 0) \sim \mathcal{N}(0, T)$$

(in this case the exact distribution $p_T(x|c = 0) \sim \mathcal{N}(0, T + 1)$ so we need to choose $T \gg 1$ to ensure sufficient terminal noise). At inference time we choose an initial sample $x_T \sim \mathcal{N}(0, T)$ and run $\mathsf{CFG}_{\mathsf{DDIM}}$ from $t = T \to 0$ to obtain a final sample x_0.

$\mathsf{CFG}_{\mathsf{DDIM}}$ For Counterexample 1, the $\mathsf{CFG}_{\mathsf{DDIM}}$ ODE has a closed-form solution (derivation in section C.7):

$$\begin{aligned}
\mathsf{CFG}_{\mathsf{DDIM}}: \quad \frac{dx}{dt} &= -\frac{1}{2}\nabla_x \log p_{t,\gamma}(x) \\
&= x_t \left(\frac{\gamma}{2(1+t)} + \frac{(1-\gamma)}{2(2+t)} \right) \\
\implies x_t &= x_T \sqrt{\frac{(t+1)^\gamma (t+2)^{1-\gamma}}{(T+1)^\gamma (T+2)^{1-\gamma}}}.
\end{aligned}$$

That is, for a particular initial sample x_T, $\mathsf{CFG}_{\mathsf{DDIM}}$ produces the sample x_t at time t. Evaluating at $t = 0$ and taking the limit as $T \to \infty$ yields the ideal denoised x_0 sampled by $\mathsf{CFG}_{\mathsf{DDIM}}$ given an initial sample x_T:

$$\begin{aligned}
\widehat{x}_0^{\mathsf{CFG}_{\mathsf{DDIM}}}(x_T) &= x_T \sqrt{\frac{2^{1-\gamma}}{(T+1)^\gamma (T+2)^{1-\gamma}}} \\
&\to x_T \sqrt{\frac{2^{1-\gamma}}{T}} \quad \text{as } T \to \infty.
\end{aligned}$$

To get the denoised distribution obtained by reverse-sampling with $\mathsf{CFG}_{\mathsf{DDIM}}$, we need to average over the distribution of x_T:

$$\mathop{\mathbb{E}}_{x_T \sim \mathcal{N}(0,T)} [\widehat{x}_0^{\mathsf{CFG}_{\mathsf{DDIM}}}(x_T)] = \mathcal{N}(0, T\frac{2^{1-\gamma}}{T}) = \mathcal{N}\left(0, 2^{1-\gamma}\right).$$

which is equation 17 in the main text.

$\mathsf{CFG}_{\mathsf{DDPM}}$ $\mathsf{CFG}_{\mathsf{DDPM}}$ also has a closed-form solution (derived in section C.7):

$$\begin{aligned}
dx &= -\nabla_x \log p_{t,\gamma}(x) dt + d\overline{w} \\
&= x \left(\frac{\gamma}{(1+t)} + \frac{(1-\gamma)}{(2+t)} \right) dt + d\overline{w}
\end{aligned}$$

$$\implies x(t) = x_T \frac{(1+t)^\gamma (2+t)^{1-\gamma}}{(1+T)^\gamma (2+T)^{1-\gamma}} + (1+t)^\gamma (2+t)^{1-\gamma} \sqrt{\frac{1}{2\gamma - 1}} \sqrt{\left(\frac{t+1}{t+2}\right)^{1-2\gamma} - \left(\frac{T+1}{T+2}\right)^{1-2\gamma}} \xi.$$

Similar to the $\mathsf{CFG}_{\mathsf{DDIM}}$ argument, we can obtain the final denoised distribution as follows:

$$\begin{aligned}
\widehat{x}_0^{\mathsf{CFG}_{\mathsf{DDPM}}}(x_T) &= x_T \frac{2^{1-\gamma}}{(1+T)^\gamma (2+T)^{1-\gamma}} + 2^{1-\gamma} \sqrt{\frac{1}{2\gamma - 1}} \sqrt{2^{2\gamma - 1} - \left(\frac{T+1}{T+2}\right)^{1-2\gamma}} \xi \\
&\to x_T \frac{2^{1-\gamma}}{T} + \sqrt{\frac{2 - 2^{2-2\gamma}}{2\gamma - 1}} \xi \quad \text{as } T \to \infty \\
\implies \mathop{\mathbb{E}}_{x_T \sim \mathcal{N}(0,T)} [\widehat{x}_0^{\mathsf{CFG}_{\mathsf{DDPM}}}(x_T)] &= \mathcal{N}\left(0, T\left(\frac{2^{1-\gamma}}{T}\right)^2 + \frac{2 - 2^{2-2\gamma}}{2\gamma - 1}\right) \\
&\to \mathcal{N}\left(0, \frac{2 - 2^{2-2\gamma}}{2\gamma - 1}\right),
\end{aligned}$$

which is equation 16 in the main text, and for $\gamma \gg 1$ becomes approximately

$$\mathop{\mathbb{E}}_{x_T \sim \mathcal{N}(0,T)} [\widehat{x}_0^{\mathsf{CFG}_{\mathsf{DDPM}}}(x_T)] \approx \mathcal{N}\left(0, \frac{2}{2\gamma - 1}\right).$$

In Figure 4, we confirm results (16, 17) empirically.

C.4 Counterexample 2

Counterexample 2 (15) is a Gaussian mixture with equal weights and variances.

$$
\begin{aligned}
c \in \{0,1\}, \quad p(c=0) &= \frac{1}{2} \\
p_0(x_0|c) \sim \mathcal{N}(\mu^{(c)}, 1), \quad \mu^{(0)} &= -\mu, \quad \mu^{(1)} = \mu \\
p_0(x_0) &\sim \frac{1}{2} p_0(x_0|c=0) + \frac{1}{2} p_0(x_0|c=1).
\end{aligned}
$$

We noted in the main text that if μ is sufficiently large enough that the clusters are approximately disjoint, and $\gamma \geq 1$, then $p_{0,\gamma}(x|c) \approx p_0(x|c)$. To see this note that

$$
\begin{aligned}
p_0(x_0) &\approx \frac{1}{2} p_0(x_0|0) \mathbb{1}_{x>0} + \frac{1}{2} p_0(x_0|1) \mathbb{1}_{x>0} \\
p_{0,\gamma}(x|c) &\propto p_0(x|c)^{\gamma} p_0(x)^{1-\gamma} \\
&= p_0(x) \left(\frac{p_0(x|c)}{p_0(x)} \right)^{\gamma} \\
&\propto p_0(x) \left(\mathbb{1}_{\text{sign}(x)=\mu^{(c)}} \right)^{\gamma} \\
&\approx p_0(x|c) \quad \text{for } \gamma \geq 1.
\end{aligned}
$$

However, $p_{t,\gamma}(x|c) \neq p_t(x|c)$ since the noisy distributions do overlap/interact.

We don't have complete closed-form solutions for this problem like we did for Counterexample 1. We have the solution for conditional DDIM for the basic VE process $dx = dw$ (using the results from the previous section):

$$
\begin{aligned}
\text{DDIM on } p_t(x|c)\text{:} \ \frac{dx}{dt} &= -\frac{1}{2} \nabla_x \log p_t(x|c) \\
&= -\frac{1}{2(1+t)} (\mu^{(c)} - x_t) \\
\implies x(t) &= \mu^{(c)} + (x_T - \mu^{(c)}) \sqrt{\frac{1+t}{1+T}},
\end{aligned}
$$

but otherwise have to rely on empirical results. We do however have access to the ideal conditional and unconditional denoisers via the scores (Appendix C.8):

$$
\begin{aligned}
\nabla_x \log p_t(x|c) &= -\frac{1}{2(1+t)} (\mu^{(c)} - x_t) \\
\nabla_x \log p_t(x) &= \frac{\nabla_x p_t(x)}{p_t(x)} = \frac{\frac{1}{2} \sum_{c=0,1} \nabla_x p_t(x|c)}{p_t(x)}.
\end{aligned}
$$

C.5 Counterexample 3

We consider a 3-cluster problem to investigate why $\mathsf{CFG}_{\mathsf{DDIM}}$ and $\mathsf{CFG}_{\mathsf{DDPM}}$ often appear similar in practice despite being different in theory. Counterexample 3 (15) is a Gaussian mixture with equal weights and variances. We vary the variance to investigate its effect on CFG.

$$
\begin{aligned}
c \in \{0,1,2\}, \quad p(c) &= \frac{1}{3} \quad \forall c \\
p_0(x_0|c) \sim \mathcal{N}(\mu^{(c)}, \sigma), \quad \mu^{(0)} &= -3, \quad \mu^{(1)} = 0, \quad \mu^{(2)} = 3 \\
p_0(x_0) &\sim \frac{1}{3} p_0(x_0|c=0) + \frac{1}{3} p_0(x_0|c=1) + \frac{1}{3} p_0(x_0|c=2).
\end{aligned}
$$

We run $\mathsf{CFG}_{\mathsf{DDIM}}$ and $\mathsf{CFG}_{\mathsf{DDPM}}$ with $\gamma = 3$, for $\sigma = 1$ and $\sigma = 2$. Results are shown in Figure 4.

C.6 Generalization Example 4

We consider a multi-cluster problem to explore the impact of guidance on generalization:

$$\begin{aligned} p_0(x) &\sim \mathcal{N}(0, 10) \\ p_0(x|c=0) &\sim \sum_i w_i \mathcal{N}(\mu_i, \sigma) \\ \mu &= (-3, -2.5, -2, -1.5, -1, -0.5, 0, 0.5, 1, 1.5, 2, 2.5) \\ w_i &= 0.0476 \quad \forall i \neq 6; \quad w_6 = 0.476 \\ \sigma &= 0.1 \end{aligned} \tag{19}$$

Note that the unconditional distribution is wide enough to be essentially uniform within the numerical support of the conditional distribution. The conditional distribution is a GMM with evenly spaced clusters of equal variance, and all equal weights, except for a “dominant" cluster in the middle with higher weight. The results are shown in Figure 3.

C.7 Closed-form ODE/SDE solutions

First, we want to solve equations of the general form $\frac{dx}{dt} = -a(t)x + b(t)$, which will encompass the ODEs and SDEs of interest to us. All we need for the ODEs is the special $b(t) = a(t)c$, which is easier.

The main results are

$$\begin{aligned} \frac{dx}{dt} &= a(t)(c - x) \\ \implies x(t) &= c + (x_T - c)e^{A(T)-A(t)} \\ \text{where } A(t) &= \int a(t)dt \end{aligned} \tag{20}$$

and

$$\begin{aligned} \frac{dx}{dt} &= -a(t)x + b(t) \\ \implies x(t) &= e^{-A(t)}(B(t) - B(T)) + x_T e^{A(T)-A(t)} \\ \text{where } A(t) &= \int a(t)dt, \quad B(t) = \int e^{A(t)} b(t)dt. \end{aligned} \tag{21}$$

First let's consider the special case $b(t) = a(t)c$, which is easier. We can solve it (formally) by separable equations:

$$\begin{aligned} \frac{dx}{dt} &= a(t)(c - x) \\ \implies \int \frac{1}{c-x} dx &= \int a(t)dt = A(t) \\ \implies -\log(c - x) &= A(t) + C \\ \implies c - x &= e^{-A(t)-C} \\ \implies x(t) &= c + Ce^{-A(t)}. \end{aligned} \tag{22}$$

Next we need to apply initial conditions to get the right constants. Remembering that we are actually sampling backward in time from initialization x_T, we can solve for the constant C as follows, to obtain result (20):

$$\begin{aligned} x_T &= c + Ce^{-A(T)} \\ \implies C &= e^{A(T)}(x_T - c) \\ \implies x(t) &= c + (x_T - c)e^{A(T)-A(t)}. \end{aligned}$$

We will apply this result to $\mathsf{CFG}_{\mathsf{DDIM}}$ shortly, but for now we note that for a VE diffusion $dx = \sqrt{t}dw$ on a Gaussian data distribution $p_0(x) \sim \mathcal{N}(\mu, \sigma)$ the above result implies the exact DDIM dynamics:

$$\begin{aligned}
p_t(x) &\sim \mathcal{N}(\mu, \sigma^2 + t) \\
\text{DDIM on } p_t(x)\text{: } \frac{dx}{dt} &= -\frac{1}{2}\nabla_x \log p_t(x) \\
&= -\frac{1}{2(\sigma^2+t)}(\mu - x) \\
A(t) &= -\frac{1}{2}\log(\sigma^2 + t) \\
\implies x_t &= \mu + (x_T - \mu)e^{A(T)-A(t)} \\
&= \mu + (x_T - \mu)\sqrt{\frac{\sigma^2+t}{\sigma^2+T}}.
\end{aligned}$$

(which makes sense since $x_{t=T} = x_T$ and $\frac{\sqrt{\sigma^2}}{\sqrt{\sigma^2+T}} \approx 0 \implies x_{t=0} \approx \mu$).

Now let's return to the general problem with arbitrary $b(t)$ (we need this for the SDEs). We can use an integrating factor to get a formal solution:

$$\begin{aligned}
\frac{dx}{dt} &= -a(t)x + b(t) \\
\text{Integrating factor: } e^{A(t)}, &\quad A(t) = \int a(t)dt \\
\frac{d}{dt}(x(t)e^{A(t)}) &= (x'(t) + a(t)x(t))\, e^{A(t)} \\
&= b(t)e^{A(t)} \\
\implies e^{A(t)}x(t) &= \int e^{A(t)}b(t)dt + C \\
\implies x(t) &= e^{-A(t)}\int e^{A(t)}b(t)dt + Ce^{-A(t)}.
\end{aligned} \tag{23}$$

Note that if $b(t) = a(t)c$ this reduces to (22):

$$\begin{aligned}
\int e^{-A(t)}e^{A(t)}b(t)dt = ce^{-A(t)}\int a(t)e^{A(t)}dt &= c \\
\implies x(t) &= c + Ce^{-A(t)}.
\end{aligned}$$

Again, we need to apply boundary conditions to get the constant, and remember that we are actually sampling backward in time from initialization x_T to obtain result (21):

$$\begin{aligned}
\frac{dx}{dt} &= -a(t)x + b(t) \\
x_T &= e^{-A(T)}B(T) + Ce^{-A(T)}, \quad B(t) := \int e^{A(t)}b(t)dt \\
\implies C &= e^{A(T)}x_T - B(T) \\
\implies x(t) &= e^{-A(t)}B(t) + (e^{A(T)}x_T - B(T))e^{-A(t)} \\
&= e^{-A(t)}(B(t) - B(T)) + x_T e^{A(T)-A(t)}.
\end{aligned}$$

Note that for $b(t) = a(t)c$ this reduces (20):

$$\begin{aligned}
b(t) = a(t)c &\implies B(t) = ce^{A(t)} \\
\implies x(t) &= -ce^{-A(t)}(e^{A(t)} - e^{A(T)}) + x_T e^{A(T)-A(t)} \\
&= c + (x_T - c)e^{A(T)-A(t)}.
\end{aligned}$$

Counterexample 1 solutions To solve the $\mathsf{CFG}_{\mathsf{DDIM}}$ ODE for Counterexample 1 (Equation 15) we apply result (20):

$$
\begin{aligned}
\frac{dx}{dt} &= a(t)(c-x) \implies x(t) = c + (x_T - c)e^{A(T)-A(t)} \\
a(t) &= -\frac{\gamma}{2(1+t)} - \frac{(1-\gamma)}{2(2+t)}, \quad c = 0 \\
A(t) &= -\frac{1}{2}\int \frac{\gamma}{(1+t)} + \frac{(1-\gamma)}{(2+t)} dt \\
&= -\frac{1}{2}(\gamma \log(t+1) + (\gamma-1)\log(t+2)) \\
\implies x_t &= x_T \sqrt{\frac{(t+1)^\gamma (t+2)^{1-\gamma}}{(T+1)^\gamma (T+2)^{1-\gamma}}}.
\end{aligned}
$$

To solve the $\mathsf{CFG}_{\mathsf{DDPM}}$ SDE for Counterexample 1 (Equation 15), we first apply (21) to the SDE with $b(t) = -\xi(t)$:

$$
\begin{aligned}
\frac{dx}{dt} &= -a(t)x - \xi(t), \quad \langle \xi(t) \rangle = 0, \quad \langle \xi(t), \xi(t') \rangle = \delta(t-t') \\
\implies x(t) &= x_T e^{A(T)-A(t)} + e^{-A(t)}(B(t) - B(T)), \quad A(t) = \int a(t)dt, \quad B(t) = -\int e^{A(t)}\xi(t)dt \\
&= x_T e^{A(T)-A(t)} + e^{-A(t)} \sqrt{\int_t^T e^{2A(t)} dt} \xi.
\end{aligned}
$$

Now, plugging in the DDPM drift term we find that

$$
\begin{aligned}
a(t) &= -\frac{\gamma}{(1+t)} - \frac{(1-\gamma)}{(2+t)} \\
A(t) &= -\gamma \log(1+t) - (1-\gamma)\log(2+t) \\
e^{A(t)} &= (1+t)^{-\gamma}(2+t)^{-1+\gamma} \\
\int e^{2A(t)} dt &= \int (1+t)^{-2\gamma}(2+t)^{-2+2\gamma} dt \\
&= -\frac{1}{2\gamma - 1}\left(\frac{t+1}{t+2}\right)^{1-2\gamma} \\
x(t) &= x_T e^{A(T)-A(t)} + e^{-A(t)} \sqrt{\int_t^T e^{2A(t)} dt} \xi \\
&= x_T \frac{(1+t)^\gamma (2+t)^{1-\gamma}}{(1+T)^\gamma (2+T)^{1-\gamma}} + (1+t)^\gamma (2+t)^{1-\gamma} \sqrt{\frac{1}{2\gamma-1}} \sqrt{\left(\frac{t+1}{t+2}\right)^{1-2\gamma} - \left(\frac{T+1}{T+2}\right)^{1-2\gamma}} \xi.
\end{aligned}
$$

C.8 Exact Denoiser for GMM

For the experiments in Figure 2, we used an exact denoiser, for which we require exact conditional and unconditional scores. Exact scores are available for any GMM as follows. This is well-known (e.g. Karras et al. [7]) but repeated here for convenience.

$$
\begin{aligned}
p(x) = \sum w_i \phi(x; \mu_i, \sigma_i), \quad \text{where} \quad \phi(x; \mu, \sigma^2) &:= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\
\implies \nabla \log p(x) &= \frac{\nabla p(x)}{p(x)} \\
&= \frac{\sum w_i \nabla \phi(\mu_i, \sigma_i)}{\sum w_i \phi(\mu_i, \sigma_i)} \\
&= -\frac{\sum w_i \left(\frac{x-\mu_i}{\sigma_i^2}\right) \phi(x; \mu_i, \sigma_i^2)}{\sum w_i \phi(\mu_i, \sigma_i)}.
\end{aligned}
$$

D PCG SDE

We want to show that the SDE limit of Algorithm 1 with $K = 1$ is

$$dx = \Delta\mathsf{DDIM}(x, t) + \Delta\mathsf{LD}_\mathsf{G}(x, t, \gamma).$$

To see this, note that a single iteration of Algorithm 1 with $K = 1$ expands to

$$
\begin{aligned}
x_t &= x_{t+\Delta t} \underbrace{-\frac{1}{2}\beta_t(x_{t+\Delta t} - \nabla \log p_{t+\Delta t}(x_{t+\Delta t}|c))\Delta t}_{\text{DDIM step on } p_{t+\Delta t}(x+\Delta t|c)} + \underbrace{\frac{\beta_t \Delta t}{2} \nabla \log p_{t,\gamma}(x_t|c) + \sqrt{\beta_t \Delta t}\mathcal{N}(0, I_d)}_{\text{Langevin dynamics on } p_{t,\gamma}(x|c)} \\
\implies dx &= \lim_{\Delta t \to 0} x_t - x_{t+\Delta t} = \underbrace{-\frac{1}{2}\beta_t(x_t - \nabla \log p_t(x_t|c))dt}_{\Delta\mathsf{DDIM}(x,t)} + \underbrace{\frac{1}{2}\beta_t \nabla \log p_{t,\gamma}(x_t|c)dt + \sqrt{\beta_t}d\overline{w}}_{\Delta\mathsf{LD}_\mathsf{G}(x,t,\gamma)}.
\end{aligned}
$$

This concludes the proof.

A subtle point in the argument above is that $\Delta\mathsf{LD}_\mathsf{G}(x, t, \gamma)$ represents the result of the Langevin step in the PCG corrector update, rather than the differential of an SDE. In Algorithm 1, t remains constant during the LD iteration, and so the SDE corresponding to the LD iteration is

$$dx = \frac{1}{2}\beta_t \nabla \log p_{t,\gamma}(x_t|c)ds + \sqrt{\beta_t}d\overline{w}, \tag{24}$$

where s is an LD time-axis that is distinct from the denoising time t, which is fixed during the LD iteration. Thus $\Delta\mathsf{LD}_\mathsf{G}(x, t, \gamma)$ is not the differential of (24) (the difference is dt vs ds). However, when we take an LD step of length dt as required for the PCG corrector, the result is

$$\int_0^{dt} -\frac{\beta_t}{2}\nabla \log p_{t,\gamma} ds + \sqrt{\beta_t}d\overline{w} = -\frac{\beta_t}{2}\nabla \log p_{t,\gamma} dt + \sqrt{\beta_t}d\overline{w} = \Delta\mathsf{LD}_\mathsf{G}(x, t, \gamma),$$

so $\Delta\mathsf{LD}_\mathsf{G}(x, t, \gamma)$ represents the result of the PCG corrector update in the limit as $\Delta t \to 0$.

E Stable Diffusion Examples

We include several examples running predictor-corrector guidance on Stable Diffusion XL [16]. These serve primarily to sanity-check our theory, not as a suggestion for practice.

For all experiments, we use $\mathsf{PCG}_\mathsf{DDIM}$ as implemented explicitly in Algorithm 2[2]. Note that PCG offers a more flexible design space than standard CFG; e.g. we can run multiple corrector steps for each denoising step to improve the quality of samples (controlled by parameter K in Algorithm 2).

CFG vs. PCG. Figure 1 illustrates the equivalence of Theorem 3: we compare $\mathsf{CFG}_\mathsf{DDPM}$ with guidance γ to $\mathsf{PCG}_\mathsf{DDIM}$ with exponent $\gamma' := (2\gamma - 1)$. We run $\mathsf{CFG}_\mathsf{DDPM}$ with 200 denoising steps, and $\mathsf{PCG}_\mathsf{DDIM}$ with 100 denoising steps and $K = 1$ Langevin corrector step per denoising step. Corresponding samples appear to have qualitatively similar guidance strengths, consistent with our theory.

[2]Note that Algorithm 1 and 2 have slightly different DDIM steps, but this just corresponds to two different discretizations of the same process. Algorithm 1 uses the first-order Euler–Maruyama discretization known as "reverse SDE" [24], which is convenient for our mathematical analysis. Algorithm 2 uses the original DDIM discretization [22], equivalent to a more sophisticated integrator [9], which is more common in practice.

Figure 5: **Effect of Guidance and Correction.** Each grid shows SDXL samples using $\mathsf{PCG}_{\mathsf{DDIM}}$, as the guidance strength γ and Langevin iterations K are varied. Left: "photograph of a dog drinking coffee with his friends". Right: "a tree reflected in the hood of a blue car". (Zoom in to view).

Figure 6: **Effect of Langevin Dynamics**. PCG generations with $\gamma = 1$ (no guidance) fixed and number of Langevin steps K varied. The prompt is "photograph of a panda eating pizza". Increasing the number of Langevin steps can qualitatively improve image quality, even without guidance.

Effects of Guidance and Corrector. In Figure 5 we show samples from $\mathsf{PCG}_{\mathsf{DDIM}}$, varying the guidance strength and Langevin iterations (i.e. parameters γ and K respectively in Algorithm 2). We also include standard $\mathsf{CFG}_{\mathsf{DDIM}}$ samples for comparison. All samples used 1000 denoising steps for the base predictor. Overall, we observed that increasing Langevin steps tends to improve the overall image quality, while increasing guidance strength tends to improve prompt adherence. In particular, sufficiently many Langevin steps can sometimes yield high-quality conditional samples, even *without* any guidance ($\gamma = 1$); see Figure 6 in the Appendix for another such example. This is consistent with the observations of Song et al. [24] on unguided predictor-corrector methods. It is also related to the findings of Du et al. [3] on MCMC methods: Du et al. [3] similarly use an annealed Langevin dynamics with reverse-diffusion annealing, although they focus on general compositions of distributions rather than the specific gamma-powered distribution of CFG.

Notice that in Figure 5, increasing the number of Langevin steps appears to also increase the "effective" guidance strength. This is because the dynamics does not fully mix: one Langevin step ($K = 1$) does not suffice to fully converge the intermediate distributions to $p_{t,\gamma}$.

F Additional Samples

G Algorithms

Algorithm 2: $\mathsf{PCG}_{\mathsf{DDIM}}$, explicit

Input: Conditioning c, guidance weight $\gamma \geq 0$
Constants: $\{\alpha_t\}, \{\overline{\alpha}_t\}, \{\beta_t\}$ from Ho et al. [5]

$x_1 \sim \mathcal{N}(0, I)$
for $(t = 1 - \Delta t;\ t \geq 0;\ t \leftarrow t - \Delta t)$ **do**
  $\varepsilon, \varepsilon_c := \mathsf{NoisePredictionModel}(x_{t+\Delta t}, c)$
  $\widehat{x}_0 := (x_{t+\Delta t} - \sqrt{1 - \overline{\alpha}_{t+\Delta t}}\varepsilon_c)/\sqrt{\overline{\alpha}_{t+\Delta t}}$
  $x_t := \sqrt{\overline{\alpha}_t}\widehat{x}_0 + \sqrt{1 - \overline{\alpha}_t}\varepsilon_c$ ▷ DDIM step on $p_t(x|c)$
  for $k = 1, \ldots K$ **do**
    $x_t \leftarrow x_t - \frac{\beta_t}{2\sqrt{1-\overline{\alpha}_t}}\left((1-\gamma)\varepsilon + \gamma\varepsilon_c\right) + \sqrt{\beta_t}\eta$ ▷ Langevin dynamics on $p_{t,\gamma}(x|c)$
  end
end
return x_0

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Claims are acccurate.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations section included.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [Yes]

Justification: We include proofs.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: Experimental setup described, using open-source models.

   Guidelines:

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

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

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

Answer: [No]

Justification: Paper is mainly theoretical.

Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: Details provided.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: No hypothesis is formally tested.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

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

8. **Experiments Compute Resources**

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

   Answer: [No]

   Justification: Mainly theoretical paper.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We have reviewed the Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: The paper is mainly theoretical; societal impact is unclear.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: No code or models released.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Code assets (eg SDXL) are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

Answer: [NA]

Justification: No new assets.

Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: No human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: No human subjects.

    Guidelines:

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