Adaptive Guidance: Training-free Acceleration of Conditional Diffusion Models

Angela Castillo[1*], Jonas Kohler[2*], Juan C. Pérez[2,3*], Juan Pablo Pérez[1], Albert Pumarola[2], Bernard Ghanem[3], Pablo Arbeláez[1], Ali Thabet[2]

[1]Center for Research and Formation in Artificial Intelligence, Universidad de los Andes
[2]GenAI, Meta
[3]King Abdullah University of Science and Technology (KAUST)

Abstract

This paper presents a comprehensive study on the role of Classifier-Free Guidance (CFG) in text-conditioned diffusion models from the perspective of inference efficiency. In particular, we relax the default choice of applying CFG in all diffusion steps and instead propose to search for more efficient guidance policies. We formulate the discovery of such policies in the framework of differentiable neural architecture search. Our findings suggest that, as denoising progresses, the updates produced by CFG become increasingly aligned with simple conditional steps, which renders CFG's additional neural network evaluation redundant, especially in the second half of the denoising process. Building upon this insight, we propose "Adaptive Guidance" (AG), an efficient variant of CFG that adaptively omits network evaluations when the denoising process displays convergence. Our experiments demonstrate that AG preserves CFG's image quality while reducing computation by 25%. Thus, AG constitutes a plug-and-play alternative to Guidance Distillation, achieving 50% of the speed-ups of the latter, while being training-free and retaining the capacity to handle negative prompts. We conclude by uncovering further redundancies of CFG in the first half of the diffusion process, showing that entire neural network evaluations can be replaced by simple affine transformations of past score estimates[1].

Introduction

Diffusion Models (DMs) (Ho, Jain, and Abbeel 2020; Song et al. 2020) exhibit outstanding generative capacities across domains such as images (Rombach et al. 2022), video (Ho et al. 2022a), audio (Kong et al. 2020), human pose estimation (Castillo et al. 2023), and even cosmological simulations (Schanz, List, and Hahn 2023). DMs generate data by sampling noise and iteratively denoising it with a neural network. The sequential nature of this denoising operation makes sampling from DMs a slow and expensive process. In particular, the time required to sample from a given DM is a function of *(i)* the latency of each denoising iteration, and *(ii)* the number of denoising steps.

Many practical applications target "conditional generation", where DMs synthesize samples conditioned on specific criteria such as a class, a text, or an image (Nichol et al. 2021). DMs perform conditional generation by replacing unconditional denoising steps with conditional ones, in which the neural network processes both the input and the condition. While conditional denoising steps provide competitive results, Ho *et al.* proposed Classifier-Free Guidance (CFG) (Ho and Salimans 2022) as a technique to enhance sample quality. CFG enriches the conditional denoising process by leveraging implicit priors of the diffusion model itself. Despite its simplicity, CFG significantly improves sample quality in tasks such as text-to-image generation (Nichol et al. 2021; Peebles and Xie 2023; Dai et al. 2023), image editing (Brooks, Holynski, and Efros 2023; Meng et al. 2021; Sheynin et al. 2023), and text-to-3D (Poole et al. 2022; Lin et al. 2023). Yet, the benefits of CFG come at the cost of *duplicating* the Number of Function Evaluations (NFEs), since each denoising iteration requires evaluating the neural network both conditionally *and* unconditionally. Adding to the problem, neural networks used in practice for DMs exceed the parallelization capacity of production-grade GPUs. For instance, with `bfloat16` precision and a batch size of 1, an EMU-768 model requires 1,553 ms on an A100 GPU without CFG and nearly doubles to 2,865 ms with CFG. This fact prevents simultaneous computation of the conditional and unconditional function evaluations.

In this paper, we improve the efficiency of text-to-image DMs that use CFG. Our analysis reveals that not all denoising steps contribute equally to image quality, suggesting that the traditional policy of applying CFG in all steps is sub-optimal. Instead, we search for policies offering more desirable trade-offs between quality and NFEs by employing techniques from differentiable neural architecture search (Liu, Simonyan, and Yang 2018). Our search uncovers that unnecessary function evaluations occur in the latter part of the denoising process. based on this we propose "Adaptive Guidance" (AG), an adaptive version of CFG that maintains image quality while reducing NFEs by 25%. See Fig. 1 for AG's generation quality.

Compared to the popular alternative of Guidance Distillation (GD) (Meng et al. 2023), AG is easy to implement, training-free, and preserves the capacity to handle negative prompts and image conditioning, which are useful capabilities for editing tasks. Finally, we propose LINEARAG, a fast version of AG that estimates updates required by AG as a

[*]These authors contributed equally.

[1]Find more at bcv-uniandes.github.io/adaptiveguidance-wp/

Figure 1: **Accelerating Guided Diffusion Models with Adaptive Guidance.** We present Adaptive Guidance (AG), an efficient variant of CFG that reduces the NFEs by 25% without compromising image quality. AG is a training-free, plug-and-play alternative to Guidance Distillation (Meng et al. 2023) that achieves 50% of its speed-ups while offering the ability to handle dynamic negative prompts. As shown above, our approach (left) replicates the baseline one-to-one and furthermore outperforms a naïve reduction of diffusion steps (right).

linear combination of past iterates. LINEARAG provides further reductions in computation at the cost of imperceptible losses in sample quality.

In summary, our contributions are threefold:

1. We show that efficient guidance policies for sampling from Diffusion Models can be found via gradient-based neural architecture search.
2. We propose "Adaptive Guidance", a plug-and-play alternative to Guidance Distillation that achieves 50% of the speed-ups of the latter, while retaining the ability to handle dynamic negative prompts and image conditioning.
3. We discover that regularities across diffusion paths enable replacing certain network evaluations in CFG with affine transformations of past iterates. This observation enables further runtime reductions and constitutes an interesting starting point for future research.

Thereby, our findings provide insights into the efficiency of the conditional denoising process that contribute to more practical deployment of text-conditioned diffusion models.

Related Work

Fast Inference with Diffusion Models

Diffusion Models (DMs) (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021) achieve density estimation and sampling by modeling a reversible transport map T that pushes forward a tractable base distribution p_b (usually a standard Gaussian) to a target distribution $p_*(\mathbf{x})$, *i.e.*, $T\#p_b = p_*(\mathbf{x})$. In contrast to traditional measure transport approaches (*e.g.*, (Dinh, Sohl-Dickstein, and Bengio 2016; Kingma and Dhariwal 2018; Chen et al. 2018)), DMs do not parameterize T explicitly but rather learn it implicitly from the reverse direction of a gradual noising process. DMs, hence, enjoy the benefit of the transport T being learnable without the need for simulation. However, DMs also suffer from having high inference costs due to the iterative nature of the sampling process.

Thus, a significant body of work aims to accelerate and optimize sampling from DMs. One angle of attack is improving the solvers used for integrating the differential equations that drive the measure transport process. For example, methods based on exponential integrators (Lu et al. 2022a,b), higher-order solvers (Zhao et al. 2023; Karras et al. 2022; Zhang and Chen 2022) or model-specific bespoke solvers (Shaul et al. 2023; Zheng et al. 2023; Wizadwongsa and Suwajanakorn 2023; Zheng et al. 2024) have been proposed. Orthogonal to these efforts, alternative strategies include parallelizing sampling via fixed-point iterations (Shih et al. 2023), reducing the size of denoising networks (Peebles and Xie 2023; Yang et al. 2023; Li et al. 2023b), and decreasing the diffusion's latent space (Gu et al. 2022; Ho et al. 2022b; Rombach et al. 2022; Rampas et al. 2022). Recent research has also focused on reformulating the diffusion process to reduce curvature in both forward (noising) (Albergo, Boffi, and Vanden-Eijnden 2023; Lipman et al. 2022) and backward (denoising) trajectories (Liu, Gong, and Liu 2022; Pooladian et al. 2023; Lee, Kim, and Ye 2023; Karras et al. 2022), which allows for larger solver steps even when employing lower-order solvers. Along these lines, (Salimans and Ho 2022) suggests progressively reducing diffusion steps through distillation.

AutoDiffusion (Li et al. 2023a) is conceptually similar to our work as it uses a neural architecture search-inspired algorithm to speed up pre-trained DMs. In contrast, our approach utilizes a more efficient gradient-based search rather than an evolutionary one, and we optimize per-step guidance options, whereas AutoDiffusion focuses on the time schedule and network architecture.

Conditioning Diffusion Paths

For both image generation and editing, the most challenging and practical cases involve some form of conditioning. Inspired by the success of class-conditioning in GANs (*e.g.*, (Odena, Olah, and Shlens 2017)), (Dhariwal

(a) $\bar{\gamma} = \infty$ (40 NFEs) (b) $\bar{\gamma} = 0.993$ (32 NFEs) (c) $\bar{\gamma} = 0.991$ (30 NFEs) (d) Zoom of left image

(e) $steps = 20$ (40 NFEs) (f) $steps = 16$ (32 NFEs) (g) $steps = 15$ (30 NFEs) (h) Zoom of left image

Figure 2: **Adaptive Guidance (AG) *vs.* Classifier-Free Guidance (CFG) for various Number of Function Evaluations (NFEs).** For AG, we fix the number of denoising iterations and reduce the NFEs that use CFG by increasing the threshold $\bar{\gamma}$ (top). CFG reduces NFEs by directly decreasing the total number of denoising steps (bottom). Vertically-aligned samples require the same NFEs. Notably, AG closely replicates the baseline, while when CFG uses fewer steps it introduces undesirable artifacts (for instance, see the helmet's glass).

and Nichol 2021) enhances the estimates of the diffusion probability path $p_t(\mathbf{x}|\mathbf{c})$ with the gradient of an image classifier $p_\theta(\mathbf{c}|\mathbf{x})$. Similarly, (Nichol et al. 2021) uses CLIP guidance for text-to-image generation with DMs. Yet, both approaches are prone to adversarial outcomes (*i.e.*, degenerate solutions) and struggle with the domain shift between the noisy images of the diffusion sampling process and the clean images on which the guidance models are trained.

In their seminal work, Ho *et al.* (Ho and Salimans 2022) show that the diffusion process can be successfully conditioned in a "classifier-free" manner by leveraging implicit priors of the DM itself. Toward this end, Ho *et al.* jointly train a network to predict both unconditional and conditional scores. During generation, the two scores are combined, giving rise to the technique known as Classifier-Free Guidance (CFG), to pinpoint samples with high conditional probability using the inverted DM as an implicit classifier. Despite the inherent imperfections of implicit classifiers, especially when the model's representation of the data distribution is not perfect, CFG's effectiveness remains evident in practice. Unfortunately, by design, CFG requires two function evaluations per step instead of one. While simultaneous inference of conditional and unconditional scores is common practice, CFG nearly doubles the sampling latency of state-of-the-art models that saturate the parallelization capacities of GPUs.

Guidance Distillation (GD) (Meng et al. 2023) performs distillation to mitigate the need for an additional unconditional forward pass. However, GD requires re-training as well as re-evaluation, both of which are resource intensive. For instance, to achieve comparable performance to CFG, applying GD on EMU-768 requires around 10k iterations with batch size of over 32, which amounts to roughly 4 days on an A100 GPU. Moreover, this technique cannot handle dynamic negative prompts, which are an important asset for responsible AI. Furthermore, GD is also incompatible with compositional guidance (Liu et al. 2022), which is used, for instance, in text-to-3D generation (Poole et al. 2022). Finally, it is unclear how to generalize GD to multimodal conditioning, such as the one employed in image editing (Brooks, Holynski, and Efros 2023; Sheynin et al. 2023).

In this work, we propose AG, a plug-and-play technique that achieves 50% to 75% of GD speed-ups while maintaining equal sample quality and addressing the aforementioned problems. AG supports image editing, negative prompts, requires no training, and precisely replicates baseline outputs, such that no re-evaluation is needed.

Neural Architecture Search

Neural architecture search automates neural network design by conceptualizing the network as a Directed Acyclic Graph (DAG), with layers as nodes (Zoph and Le 2016; Zoph et al. 2018; Pham et al. 2018; Liu et al. 2018; Brock et al. 2017). Our review focuses on differentiable methods (Liu, Simonyan, and Yang 2018; Li et al. 2020, 2022; Wu et al. 2019). The DARTS framework (Liu, Simonyan, and Yang 2018) is central to our work, enabling differentiable and efficient architecture search through continuous relaxation of layer representation (see Tab. 3 in (Liu, Simonyan, and Yang 2018)). We extend the analogies between neural network design and the denoising process by unrolling the synthesis process across time, treating each denoising step as a distinct node in the DAG, which allows us to apply DARTS conceptualization directly to search for the optimal guidance at each node.

Methodology

Background on Diffusion Models

As introduced before, diffusion models generate images by reversing a pre-defined noising process. In particular, when the noising process is an Ornstein-Uhlenbeck process, the continuous-time limit of the forward Stochastic Differ-

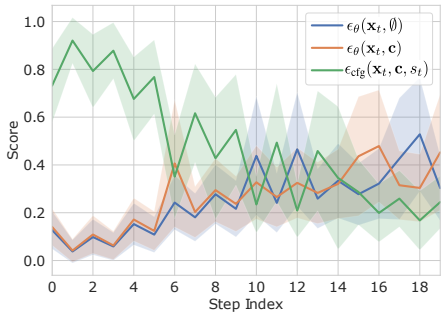

Figure 3: **Search results over guidance policies.** These results suggest that CFG is most important at the beginning of denoising, and then its importance decreases over time.

ential Equation (SDE) reads as $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\, dt + g(t)\, d\mathbf{w}$, where $f(\mathbf{x}, t) : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued drift coefficient, $g(t) : \mathbb{R} \to \mathbb{R}$ is the diffusion coefficient of $\mathbf{x}(t)$ and $\mathbf{w}$ is the standard Brownian motion. Anderson's Theorem (Anderson 1982) states that, under mild assumptions, such SDE satisfies the reverse-time process

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + g(t)\, d\bar{\mathbf{w}}, \quad (1)$$

where $\bar{\mathbf{w}}$ is the reverse-time Brownian motion. As shown in (Hyvärinen and Dayan 2005; Song et al. 2020), the marginal transport map can be learned (in expectation) by maximum likelihood estimation of the perturbation kernel of individually diffused data samples $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}|\mathbf{x}_0)$ in a simulation-free manner. This map is commonly learnt by optimizing the parameters θ of a time-conditioned neural network, yielding score estimates $\epsilon_\theta(\mathbf{x}_t, t)$. For brevity's sake, we omit the conditioning of ϵ on t going forward.

As shown in (Song et al. 2020), the SDE in Eq. (1) has a deterministic counterpart (*i.e.*, an ODE) that enjoys equivalent marginal probability densities:

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt. \quad (2)$$

When using few (and large) discretization steps, solving Eq. (2) generally yields better results than solving its stochastic counterpart (Karras et al. 2022).

Conditional generation with diffusion models. The diffusion framework can be extended to allow for conditional generation by learning the score $\log p_t(\mathbf{x}|\mathbf{c})$, where $\mathbf{c}$ is, for example, a class or text condition. Current state-of-the-art models for conditional generation employ "Classifier-Free Guidance" (CFG) (Ho and Salimans 2022), a technique in which both the conditional and unconditional scores are linearly combined to denoise the sample. In particular, CFG proposes to follow the score estimate given by

$$\epsilon_{\text{cfg}}(\mathbf{x}_t, \mathbf{c}, s) = \epsilon_\theta(\mathbf{x}_t, \emptyset) + s \cdot (\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) - \epsilon_\theta(\mathbf{x}_t, \emptyset)), \quad (3)$$

where $\emptyset$ is the unconditional prompt token, and $s > 1$ indicates the guidance strength. While this new score may not directly reflect the gradient of a classifier's log-likelihood, it is inspired by the gradient of an implicit classifier $p'(\mathbf{c}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{c})/p(\mathbf{x})$. As a result, $\nabla_{\mathbf{x}} \log p(\mathbf{c}|\mathbf{x}) \propto \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{c}) - \nabla_{\mathbf{x}} \log p(\mathbf{x})$ and hence $\epsilon_\lambda(\mathbf{x}_t, \mathbf{c}) \propto \epsilon(\mathbf{x}_t, \emptyset) + s \cdot \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{c})$. In that sense, CFG shifts probability mass toward data where an implicit classifier $p'(\mathbf{c}|\mathbf{x})$ assigns a high likelihood to the condition $\mathbf{c}$.

Notably, evaluating Eq. (3) introduces an extra function evaluation compared to unguided sampling, which may almost double the latency. Therefore, we search for efficient ways of guiding the denoising process, aiming at reducing NFEs while retaining the benefits of CFG. Next, we discuss these approaches.

Gradient Search along Diffusion Dynamics

Assuming a pre-trained diffusion model $\Phi : X \times C \to X$, where $X = \mathbb{R}^{H \times W \times C}$ is the latent space and C represents the condition space, with $\mathbf{c} \in C$ denoting a condition (*e.g.*, a text prompt), we initialize $\mathbf{x}_T \sim p_b$, where p_b is a Gaussian distribution. By setting a condition $\mathbf{c} \in C$ and a time-schedule $\tau = \{T, T-1, ..., 0\}$, the diffusion model generates a sequence of latent codes

$$\{\mathbf{x}_t\}_{t=0}^{T} \text{ s.t. } \mathbf{x}_T \sim p_b,\ \mathbf{x}_{t-1} = \Phi(\text{solver}(\bar{\mathbf{x}}_t)), \quad (4)$$

where "solver" represents an ODE solver for Eq. (2). The model Φ operates uses CFG as in Eq. (3), with $\bar{\mathbf{x}}_t = \epsilon_{\text{cfg}}(\mathbf{x}_t, \mathbf{c}, s)$, where s remains constant over time. While this setup is standard in many diffusion models (Rombach et al. 2022; Dai et al. 2023; Saharia et al. 2022; Nichol et al. 2021), alternative options for $\bar{\mathbf{x}}_t$ exist, each with its associated computational costs:

(i)	Unconditional score:	$\epsilon_\theta(\mathbf{x}_t, \emptyset)$	(1 NFE)
(ii)	Conditional score:	$\epsilon_\theta(\mathbf{x}_t, \mathbf{c})$	(1 NFE)
(iii)	CFG score:	$\epsilon_{\text{cfg}}(\mathbf{x}_t, \mathbf{c}, s_t)$	(2 NFEs)

Here, ϵ_θ represents a neural network with fixed weights θ, and s_t is no longer constant in time. We denote by f_t the particular step *choice* at time t with $f_t \in \mathcal{F}_t = \{\epsilon_\theta(\mathbf{x}_t, \emptyset), \epsilon_\theta(\mathbf{x}_t, \mathbf{c}), \epsilon_{\text{cfg}}(\mathbf{x}_t, \mathbf{c}, s_t)\}$.

The search space for the denoising process is $\mathcal{S} = \prod_{t=0}^{T} \mathcal{F}_t$, representing all possible sequences of choices $\zeta = (f_0, f_1, \ldots, f_T)$, which we refer to as *policies*. Here, $\prod$ denotes the Cartesian product over sets. As $s_t \in \mathbb{R}$, $\mathcal{S}$ is unbounded, but to simplify and generalize policies, we constrain s_t to belong to a finite set $\mathcal{S} = \{s^1, ..., s^k\}$. Thus, there are $|\mathcal{S}| = |\prod_{t=0}^{T} \mathcal{F}_t| = (2+k)^{T+1}$ different policies.

To enable backpropagation, we relax the discrete search to a continuous one. Each set of choices $\mathcal{F}_t$ is associated with a trainable vector $\boldsymbol{\alpha}_t \in \mathbb{R}^{k+2}$, and we obtain the solver's input using a softmax weighting of options, *i.e.*,

$$\bar{\mathbf{x}}_t := \text{softmax}\,(\boldsymbol{\alpha})^\intercal\, \mathcal{F}_t. \quad (5)$$

The score matrix $\boldsymbol{\alpha} := [\boldsymbol{\alpha}_T^\intercal, ..., \boldsymbol{\alpha}_0^\intercal]$ represents a multinomial distribution over options, from which concrete policies ζ can be sampled.

We define a differentiable objective to guide the search for efficient policies, aiming to find a policy ζ that closely replicates Φ, as measured by a differentiable distance metric $d(\mathbf{x}_0, \mathbf{x}'_0)$. Thus, we optimize:

$$\boldsymbol{\alpha}^* = \text{argmin}_{\boldsymbol{\alpha}} \left[d(\mathbf{x}_0, \mathbf{x}'_0(\zeta(\boldsymbol{\alpha})) + \lambda g(\zeta(\boldsymbol{\alpha}))\right], \quad (6)$$

where $\lambda > 0$ and $g(\zeta(\boldsymbol{\alpha}))$ regularizes the sum of scores weighted by the per-choice costs. We initialize $\boldsymbol{\alpha}$ randomly, update it iteratively by sampling $\mathbf{x}_T \sim \mathcal{N}(0, I)$, and backpropagate through a student model Φ' (which mimics Φ but

(a) Individual denoising iterates.

(b) Element-wise difference between consecutive iterates.

Figure 4: **Denoising process.** Point-wise differences reveal scene organization from the start.

uses the soft alphas from Eq. (5)) to obtain $\mathbf{x}'_0(\zeta(\boldsymbol{\alpha}))$. We elaborate on this search in the Suppl.

Adaptive Guidance

Building on the insights gained from this search, we observed that conditional and unconditional updates become increasingly correlated over time. This fact suggests an intuitive way to save NFEs by stopping CFG computation when this correlation is high. This observation led us to develop a more efficient approach to guidance, which we term "Adaptive Guidance" (AG), a principled technique to decrease sampling cost while maintaining high image quality. AG leverages the correlation to reduce the number of NFEs by strategically switching from CFG updates to conditional updates when the correlation, measured by γ_t (Eq. (8)), exceeds a certain threshold $\bar{\gamma}$. Consequently, AG produces simplified policies, such as

$$\zeta_{\text{AG}} = [\epsilon_{\text{cfg}}(\mathbf{x}_T, \mathbf{c}), ..., \epsilon_{\text{cfg}}(\mathbf{x}_t, \mathbf{c}), \epsilon_\theta(\mathbf{x}_{t-1}, \mathbf{c}), ..., \epsilon_\theta(\mathbf{x}_0, \mathbf{c})], \tag{7}$$

where the truncation timestep t is a function of $\bar{\gamma}$, the starting seed $\mathbf{x}_T$ and the conditioning $\mathbf{c}$. More details in the Suppl.

We highlight that ζ_{AG} is independent of the specific time schedule τ (Eq. (4)) or solver used for sampling, making it adaptable to a wide range of diffusion models and arbitrary quantities of diffusion steps.

Replacing NFEs with Affine Transformations

Guided by these insights, we found that in the latter stages of denoising CFG updates can be replaced with conditional steps. However, during the first half of the denoising process, guidance remains crucial. As demonstrated in Fig. 9a, prematurely replacing CFG updates with conditional steps —such as reducing the number of guidance steps to just five, followed by 15 conditional steps— can significantly degrade image quality. At the same time, the smooth alignment of conditional and unconditional steps over time, along with the consistent cosine similarities shown in Fig. 5, suggest a high regularity in diffusion paths. Intrigued by this observation, we probe for linear patterns in the diffusion path.

Experiments

Experimental Setup. We optimize guidance policies for text-to-image generation using the Stable Diffusion architecture (Rombach et al. 2022), referred to as **LDM-512**. We train LDM-512 from scratch on a commissioned dataset, consisting of 900M parameters and generating 512×512 resolution images from a $4 \times 64 \times 64$ latent space. Training involves 10k noise-image pairs from CC3M (Sharma et al. 2018) with $T = 20$ DPM++ (Lu et al. 2022a) steps and a guidance strength of $s = 7.5$. CC3M is used because the linguistic richness and contextual variety of its prompts ensure meaningful and realistic image synthesis. The search space $\mathcal{S}$ includes $k = 3$ guidance strengths: unconditional, conditional, and scaled $\epsilon_{\text{cfg}}(\mathbf{x}_t, \mathbf{c}, a \cdot 7.5)$ with $a \in \{1/2, 1, 2\}$. We optimize Eq. (6) using the Lion optimizer (Chen et al. 2023) for 5 epochs. Evaluation metrics are computed on 1k real-world user prompts from the OUI dataset (Dai et al. 2023), which represents diverse, practical user inputs. The search took approximately 1.5 days on a Quadro RTX 8000.

Model	GPU	AG (30 NFEs)	CFG (40 NFEs)	AG Gain
EMU 9.5b	H100	3251 ± 26	3822 ± 31	15%
EMU 2.7b	A100	2634 ± 8	3184 ± 6	17%
SD XL 2.5b	V100	4584 ± 8	5876 ± 10	22%
SD XL 2.5b	RTX 8000	4932 ± 5	6339 ± 6	22%
SD 1.5 0.8b	GTX 1080Ti	4396 ± 27	5542 ± 30	21%

Table 1: Runtime for 20 denoising steps in ms (mean $\pm$ std., 30 runs). All models run in half-precision in PyTorch 2. Times exclude text-conditioning and VAE-decoding, which induce a $< 5\%$ overhead.

We demonstrate that our findings generalize by validating the efficacy of AG on the larger **EMU-768** model (Dai et al. 2023), with 2.7B parameters and 768×768 resolution, as well as on the **SDXL-1024** (Rombach et al. 2022) model at 1024×1024 resolution. We also include results on a latent flow matching model (**FM-256**) (Liu, Gong, and Liu 2022) trained on a smaller 256×256 dataset of sticker-like images. The following section presents our search results.

Gradient Search

Our search identifies three main guidance options: conditional, unconditional, and CFG with $s = 7.5$. This outcome aligns with expectations, as following the entire diffusion path is essential for accurate replication. The score distribution, shown in Fig. 3, reveals a clear pattern: CFG is crucial in the early stages of denoising but less so later. We suggest that text-conditioning is vital for establishing semantic structures early on, while later steps refine local details and high-frequency features (see Fig. 4).

Interestingly, this generative structure is mirrored in the diffusion process itself. The cosine similarity γ_t between conditional and unconditional predictions, $\epsilon(\mathbf{x}_t, \mathbf{c})$ and $\epsilon(\mathbf{x}_t, \emptyset)$, steadily increases until they nearly align. As shown in Fig. 5, γ_t reaches near-perfect alignment by the end of the diffusion process, a pattern observed across different models and resolutions. We empirically observe that

$$\lim_{t \to 0} \left[\gamma_t := \frac{\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) \cdot \epsilon_\theta(\mathbf{x}_t, \emptyset)}{\|\epsilon_\theta(\mathbf{x}_t, \mathbf{c})\| \|\epsilon_\theta(\mathbf{x}_t, \emptyset)\|} \right] = 1. \tag{8}$$

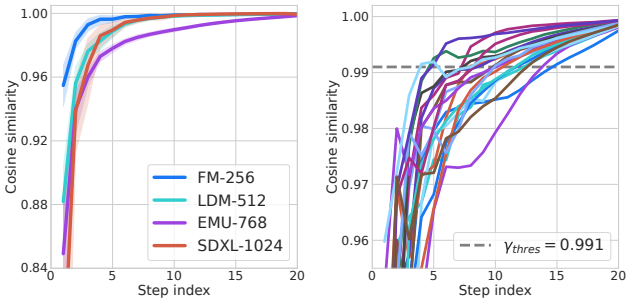

Figure 5: **Cosine similarities over time.** Left: Average cosine similarity γ and 99% confidence interval over $1,000$ OUI prompts for FM, EMU, LDM and SDXL. Right: Zoom to y-values in $[0.955, 1.0]$ for 21 EMU samples.

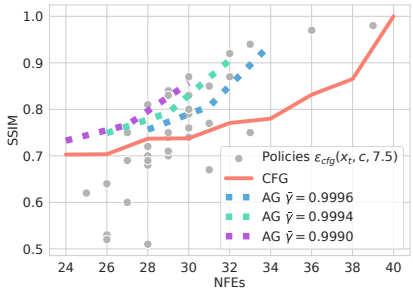

Figure 6: **Search results.** SSIM of various policies (dots) vs. 20-step CFG baseline on LDM-512. AG results (dashed lines) and CFG with naïve step reduction (solid line) are shown. AG outperforms naïve step reduction, closely matching most searched policies while being simpler and scalable.

In light of this finding, we argue that an adaptive method for guidance can reduce network evaluations by stopping guidance precisely when the conditional and unconditional update steps have approximately converged, such that guidance no longer introduces directional changes. As illustrated in Fig. 4, the semantic content dictated by the prompt is setup in the first half of diffusion process, which suggests an opportunity to cease CFG guidance early, thus saving NFEs toward the latter stage of the generative process.

Adaptive Guidance

Figure 5 shows a consistent trend of increasing cosine similarities across different models and resolutions, including FM-256, EMU-768, and the larger SDXL-1024, demonstrating the generalization capacity of our findings.

Quantitative evaluation. In our comparison of AG and CFG, we evaluate their ability to reconstruct a set of 1k target images generated from a baseline model using 20 CFG steps (40 NFEs). Running this experiment on LDM-512, as shown in Fig. 6, AG consistently outperforms CFG in replicating target images with higher accuracy across the entire regime considered (from 22 to 40 NFEs). These results, also generalize to the much larger EMU-768 model.

EMU-756	SSIM ↑	Win ↑	Lose ↓	NFEs ↓
CFG	-	502	498	40
AG $\bar{\gamma} = 0.991$	0.91 ± 0.03	498	502	29.6 ± 1.3

Table 2: **Evaluation results.** Comparison of AG ($\bar{\gamma} = 0.991$, approximately 30 NFEs) and the 20 step (*i.e.*, 40 NFEs) CFG baseline. We report average SSIM and majority voting of five trained human evaluators (each voting on 1k images generated from OUI prompts). These results show that AG achieves equal visual quality despite using 25% fewer NFEs.

Moreover, as demonstrated in Tab. 1, AG offers significant efficiency gains in terms of wall-clock time. While CFG requires two NFEs per step, resulting in 40 NFEs when $T = 20$, GD reduces this to one NFE per step, saving 20 NFEs. However, it necessitates re-training and does not support negative prompts. Alternatively, AG achieves 50% of the GD savings (reducing 10 NFEs) by dropping guidance after the first half of steps, achieving similar efficiency with greater flexibility. This efficiency is particularly advantageous for large models, which can saturate the parallelization capacity of even production-grade GPUs (A/H100), as well as for smaller GPUs commonly used in academic communities, where NFE savings translate almost 1:1 to latency reduction (RTX, GTX, and V100). An example of the performance gap between GD and AG is provided in the Suppl.

Qualitative evaluation. Figure 2 depicts samples generated with AG for different $\bar{\gamma}$ values. Our results suggest that up to 50% of the diffusion steps can be performed without CFG at no cost to image quality. Moreover, Figs. 1 and 2 showcase samples where AG outperforms the naïve alternative of reducing the total number of diffusion steps.

Human evaluation. We validated AG's image quality through a human evaluation with five trained annotators comparing images generated by CFG (40 NFEs) and AG with $\bar{\gamma} = 0.991$, achieving 25% fewer NFEs (examples in Fig. 7 and in the Suppl.). Using 1k prompts from the OUI dataset, results in Tab. 2 showed no significant preference (AG: 498 votes, CFG: 502 votes), with a mean vote difference of -0.047 (SD $= 2.543$) and a p-value of 0.603 using a two-sided Wilcoxon Signed-Rank Test. These findings indicate *no significant difference in visual appeal* between the images generated by the two models ($p > 0.05$), which suggest that AG maintains comparable image quality to CFG, despite its greater efficiency.

Image editing. To highlight our method's generalization, we showcase results in image editing. Text-to-image models are widely studied for generating novel images and instruction-based editing (*e.g.*, (Meng et al. 2021; Zhang et al. 2023; Brooks, Holynski, and Efros 2023; Sheynin et al. 2023)). InstructPix2Pix (Brooks, Holynski, and Efros 2023) extends CFG to enable conditioning on images and text, modifying the score estimate as

$$\begin{aligned}\epsilon_{\text{pix2pix}}(\mathbf{x}_t, \mathbf{c}, \mathbf{I}) = &\epsilon_\theta(\mathbf{x}_t, \emptyset, \emptyset) \\ &+ s_c \cdot (\epsilon_\theta(\mathbf{x}_t, \mathbf{c}, \mathbf{I}) - \epsilon_\theta(\mathbf{x}_t, \emptyset, \mathbf{I})) \\ &+ s_T \cdot (\epsilon_\theta(\mathbf{x}_t, \emptyset, \mathbf{I}) - \epsilon_\theta(\mathbf{x}_t, \emptyset, \emptyset)).\end{aligned} \tag{9}$$

(a) CFG (40NFEs) (b) AG (30NFEs) (c) CFG (40NFEs) (d) AG (29NFEs)

Figure 7: **Human evaluation examples.** An exemplary sample for win and lose of AG with $\bar{\gamma} = 0.991$ *vs.* CFG. Baseline CFG tends to produce higher frequencies, which can be for better or worse. More examples in the Suppl.

However, (Brooks, Holynski, and Efros 2023) introduces two important implications: *(i)* a diffusion step now requires 3 NFEs instead of 2, and *(ii)* GD cannot be applied as part of the "unconditional" update due to dynamic changes in $\mathbf{I}$ (akin to negative prompts).

These effects hinder fast image editing, where users may wish to test instructions quickly. Notably, as in text conditioning, the terms in Eq. (9) converge over time. Hence, Fig. 8 shows that AG can reduce NFEs without notable quality loss. In the images we show above, AG employs only ten (instead of 20) steps of InstructPix2Pix ($\epsilon_{\text{pix2pix}}(\mathbf{x}_t, \mathbf{c}, \mathbf{I})$), thereby saving $1/3$ of the total NFEs.

Replacing NFEs with Affine Transformations

We find that unconditional network evaluations $\epsilon(\mathbf{x}_t, \emptyset)$ can be accurately estimated with high accuracy via affine transformations of network evaluations of previous iterations. The unconditional score, being prompt-independent, is more regular and easier to learn. To compute the parameters of these affine transformations, we generate a small dataset of 200 images from EMU-768 and store the intermediate iterations. Subsequently, we model an unconditional step given at any $t < T$ as a linear combination of the previous iterations in the diffusion chain as

$$\hat{\epsilon}(\mathbf{x}_t, \emptyset) = \sum_{i=T}^{t} \beta_i^{\mathbf{c}} \epsilon_\theta(\mathbf{x}_i, \mathbf{c}) + \sum_{i=T}^{t+1} \beta_i^{\emptyset} \epsilon_\theta(\mathbf{x}_i, \emptyset), \qquad (10)$$

where $\beta_i^{\mathbf{c}}$ and $\beta_i^{\emptyset}$ are *scalars*. We learn these Linear Regression (LR) coefficients for each step by solving a simple Ordinary Least Squares problem on the 200 trajectories. Together with the time required for generating the dataset, we obtain LR coefficients for all steps in under 20 minutes. During sampling, computing $\hat{\epsilon}(\mathbf{x}_t, \emptyset)$ is thus essentially for free.

Perhaps surprisingly, we find that this estimator effectively predicts unconditional steps. Of course, for any unconditional score replaced by an LR predictor, the following denoising step will no longer have ground truth past information and errors accumulate auto-regressively. Yet, interleaving CFG steps with "estimated" CFG steps (where the step $\epsilon_\theta(\mathbf{x}_t, \emptyset)$ is replaced by its linear estimator $\hat{\epsilon}(\mathbf{x}_t, \emptyset)$), reduces the rate of error accumulation. We term this strategy LINEARAG. When $T = 20$, LINEARAG performs ten steps, alternating between CFG (2 NFEs) and LR-based CFG (1 NFE), followed by ten LR-based CFG steps. This policy results in 25 NFEs, which accounts for 75% of GD savings.

(a) Input image (b) CFG-based editing (60NFEs) (c) AG-based editing (40NFEs)

Figure 8: **Image editing.** Instruction-based editing with EMU Edit (Sheynin et al. 2023), based on InstructPix2Pix. Original image (left), classic CFG editing (Eq. (9)), and AG editing, which achieves equal quality with $1/3$ fewer NFEs.

(a) AG $\bar{\gamma} = 0.975$ (25 NFEs) (b) naïve CFG (25 NFEs) (c) LINEARAG (25 NFEs)

Figure 9: **Replacing CFG in the initial diffusion steps.** Three methods to reduce NFEs in early diffusion steps. LINEARAG samples show sharper details, and vivid colors.

Figure 9 shows that LINEARAG significantly improves image quality over AG at very low $\bar{\gamma}$. This demonstrates that the LR effectively recognizes patterns along the path, outperforming the naïve approach of alternating CFG and conditional steps in the first half, followed by $T/2$ conditional steps. Additionally, LINEARAG can handle negative prompts to some extent (see examples in the Supplementary). LR parameters, optimized for empty negative prompts, are applied in every even iteration of denoising, while odd iterations use the U-Net conditioned on a negative prompt.

Conclusions

This work identifies computational redundancies in CFG during denoising. Building on these insights, we introduce Adaptive Guidance (AG), a general and efficient plug-and-play CFG variant that closely replicates a baseline model while achieving 50% of GD speed-ups.

Unlike GD, AG requires no training, is easy to implement, and supports negative prompts and image editing. We also introduce LINEARAG, a faster CFG variant that saves up to 75% GD computations using affine transformations instead of full network evaluations. It highlights the potential of leveraging diffusion path smoothness for efficient inference, although it does not exactly replicate the baseline.

References

Albergo, M. S.; Boffi, N. M.; and Vanden-Eijnden, E. 2023. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*.

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

Brock, A.; Lim, T.; Ritchie, J. M.; and Weston, N. 2017. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*.

Brooks, T.; Holynski, A.; and Efros, A. A. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18392–18402.

Castillo, A.; Escobar, M.; Jeanneret, G.; Pumarola, A.; Arbeláez, P.; Thabet, A.; and Sanakoyeu, A. 2023. BoDiffusion: Diffusing Sparse Observations for Full-Body Human Motion Synthesis. *arXiv preprint arXiv:2304.11118*.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chen, X.; Liang, C.; Huang, D.; Real, E.; Wang, K.; Liu, Y.; Pham, H.; Dong, X.; Luong, T.; Hsieh, C.-J.; et al. 2023. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*.

Dai, X.; Hou, J.; Ma, C.-Y.; Tsai, S.; Wang, J.; Wang, R.; Zhang, P.; Vandenhende, S.; Wang, X.; Dubey, A.; et al. 2023. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*.

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

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B. 2022. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10696–10706.

Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; et al. 2022a. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

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

Ho, J.; Saharia, C.; Chan, W.; Fleet, D. J.; Norouzi, M.; and Salimans, T. 2022b. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1): 2249–2281.

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

Hyvärinen, A.; and Dayan, P. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35: 26565–26577.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31.

Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; and Catanzaro, B. 2020. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Lee, S.; Kim, B.; and Ye, J. C. 2023. Minimizing trajectory curvature of ode-based generative models. *arXiv preprint arXiv:2301.12003*.

Li, G.; Qian, G.; Delgadillo, I. C.; Muller, M.; Thabet, A.; and Ghanem, B. 2020. Sgas: Sequential greedy architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1620–1630.

Li, G.; Xu, M.; Giancola, S.; Thabet, A.; and Ghanem, B. 2022. LC-NAS: Latency constrained neural architecture search for point cloud networks. In *2022 International Conference on 3D Vision (3DV)*, 1–11. IEEE.

Li, L.; Li, H.; Zheng, X.; Wu, J.; Xiao, X.; Wang, R.; Zheng, M.; Pan, X.; Chao, F.; and Ji, R. 2023a. AutoDiffusion: Training-Free Optimization of Time Steps and Architectures for Automated Diffusion Model Acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7105–7114.

Li, Y.; Wang, H.; Jin, Q.; Hu, J.; Chemerys, P.; Fu, Y.; Wang, Y.; Tulyakov, S.; and Ren, J. 2023b. SnapFusion: Text-to-Image Diffusion Model on Mobile Devices within Two Seconds. *arXiv preprint arXiv:2306.00980*.

Lin, C.-H.; Gao, J.; Tang, L.; Takikawa, T.; Zeng, X.; Huang, X.; Kreis, K.; Fidler, S.; Liu, M.-Y.; and Lin, T.-Y. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 300–309.

Lipman, Y.; Chen, R. T.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, C.; Zoph, B.; Neumann, M.; Shlens, J.; Hua, W.; Li, L.-J.; Fei-Fei, L.; Yuille, A.; Huang, J.; and Murphy, K. 2018. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 19–34.

Liu, H.; Simonyan, K.; and Yang, Y. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Liu, N.; Li, S.; Du, Y.; Torralba, A.; and Tenenbaum, J. B. 2022. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, 423–439. Springer.

Liu, X.; Gong, C.; and Liu, Q. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022a. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35: 5775–5787.

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

Meng, C.; He, Y.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S. 2021. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*.

Meng, C.; Rombach, R.; Gao, R.; Kingma, D.; Ermon, S.; Ho, J.; and Salimans, T. 2023. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14297–14306.

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

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 8162–8171. PMLR.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, 2642–2651. PMLR.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Pham, H.; Guan, M. Y.; Zoph, B.; Le, Q. V.; and Dean, J. 2018. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*.

Pooladian, A.-A.; Ben-Hamu, H.; Domingo-Enrich, C.; Amos, B.; Lipman, Y.; and Chen, R. 2023. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Rampas, D.; Pernias, P.; Zhong, E.; and Aubreville, M. 2022. Fast text-conditional discrete denoising on vector-quantized latent spaces. *arXiv preprint arXiv:2211.07292*.

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

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

Salimans, T.; and Ho, J. 2022. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*.

Schanz, A.; List, F.; and Hahn, O. 2023. Stochastic Super-resolution of Cosmological Simulations with Denoising Diffusion Models. *arXiv preprint arXiv:2310.06929*.

Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of ACL*.

Shaul, N.; Perez, J.; Chen, R. T.; Thabet, A.; Pumarola, A.; and Lipman, Y. 2023. Bespoke Solvers for Generative Flow Models. *arXiv preprint arXiv:2310.19075*.

Sheynin, S.; Polyak, A.; Singer, U.; Kirstain, Y.; Zohar, A.; Ashual, O.; Parikh, D.; and Taigman, Y. 2023. Emu Edit: Precise Image Editing via Recognition and Generation Tasks. *arXiv preprint arXiv:2311.10089*.

Shih, A.; Belkhale, S.; Ermon, S.; Sadigh, D.; and Anari, N. 2023. Parallel Sampling of Diffusion Models. *arXiv preprint arXiv:2305.16317*.

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

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Wizadwongsa, S.; and Suwajanakorn, S. 2023. Accelerating Guided Diffusion Sampling with Splitting Numerical Methods. In *The Eleventh International Conference on Learning Representations*.

Wu, B.; Dai, X.; Zhang, P.; Wang, Y.; Sun, F.; Wu, Y.; Tian, Y.; Vajda, P.; Jia, Y.; and Keutzer, K. 2019. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10734–10742.

Yang, X.; Zhou, D.; Feng, J.; and Wang, X. 2023. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22552–22562.

Zhang, K.; Mo, L.; Chen, W.; Sun, H.; and Su, Y. 2023. MagicBrush: A Manually Annotated Dataset for Instruction-Guided Image Editing. *arXiv preprint arXiv:2306.10012*.

Zhang, Q.; and Chen, Y. 2022. Fast Sampling of Diffusion Models with Exponential Integrator. In *The Eleventh International Conference on Learning Representations*.

Zhao, W.; Bai, L.; Rao, Y.; Zhou, J.; and Lu, J. 2023. UniPC: A Unified Predictor-Corrector Framework for Fast Sampling of Diffusion Models. *arXiv preprint arXiv:2302.04867*.

Zheng, K.; Lu, C.; Chen, J.; and Zhu, J. 2023. DPM-Solver-v3: Improved Diffusion ODE Solver with Empirical Model Statistics. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zheng, K.; Lu, C.; Chen, J.; and Zhu, J. 2024. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36.

Zoph, B.; and Le, Q. 2016. Neural Architecture Search with Reinforcement Learning. In *International Conference on Learning Representations*.

Zoph, B.; Vasudevan, V.; Shlens, J.; and Le, Q. V. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8697–8710.