AdaDiff: Adaptive Step Selection for Fast Diffusion Models

Hui Zhang[1,2,3] **Zuxuan Wu**[1,2,*], **Zhen Xing**[1,2], **Jie Shao**[3], **Yu-Gang Jiang**[1,2]

[1]Shanghai Key Lab of Intell. Info. Processing, School of CS, Fudan University
[2]Shanghai Collaborative Innovation Center of Intelligent Visual Computing
[3]ByteDance Inc.

Abstract

Diffusion models, as a type of generative model, have achieved impressive results in generating images and videos conditioned on textual conditions. However, the generation process of diffusion models involves denoising dozens of steps to produce photorealistic images/videos, which is computationally expensive. Unlike previous methods that design "one-size-fits-all" approaches for speed up, we argue denoising steps should be sample-specific conditioned on the richness of input texts. To this end, we introduce AdaDiff, a lightweight framework designed to learn instance-specific step usage policies, which are then used by the diffusion model for generation. AdaDiff is optimized using a policy gradient method to maximize a carefully designed reward function, balancing inference time and generation quality. We conduct experiments on three image generation and two video generation benchmarks and demonstrate that our approach achieves similar visual quality compared to the baseline using a fixed 50 denoising steps while reducing inference time by at least 33%, going as high as 40%. Furthermore, our method can be used on top of other acceleration methods to provide further speed benefits. Lastly, qualitative analysis shows that AdaDiff allocates more steps to more informative prompts and fewer steps to simpler prompts.

Introduction

Diffusion models (Podell et al. 2024; OpenAI 2024), as a class of generative models, have made significant strides. These models have the capability to generate specified visual content, including images and videos, based on specific input conditions such as text, semantic maps, representations, and images. For example, models like SDXL (Podell et al. 2024) and Sora (OpenAI 2024) can produce perceptually-convincing or artistic images and videos conditioned on textual descriptions, *a.k.a.* prompts. These diffusion models often contain an iterative denoising process during generation, and more iterations typically indicate better visual quality. However, the improvements come at the cost of increased computational resources, even with the use of state-of-the-art sampling methods (Song, Meng, and Ermon 2021; Lu et al. 2022). Therefore, to strike a balance between quality and inference speed, the number of denoising steps is often empirically set as a fixed value.

Figure 1: **A conceptual overview of our approach.** AdaDiff assigns instance-specific denoising steps based on prompt richness to minimize inference time with minimal quality loss. Images with red borders are produced by AdaDiff.

But do we really need a fixed number of denoising steps for all different prompts? Intuitively, using more steps may lead to higher quality and more detailed content. Nonetheless, in real-world applications, the richness in textual prompts, *i.e.*, the number of objects, and how they relate to each other, vary significantly. For certain easy and coarse-grained prompts, involving only one or a few objects, using fewer steps is already sufficient to generate satisfactory results, and increasing the number of steps may lead to only marginal improvements and does not necessarily produce better results. For complex textual prompts, containing many objects, detailed descriptions, and intricate interactions between objects, a larger number of steps becomes necessary to achieve the desired result. Therefore, the goal of this paper is to develop a dynamic framework for diffusion models by adaptively determining the number of denoising steps needed for generating photorealistic contents conditioned on textual inputs. This is *in contrast yet complimentary* to existing methodologies, which can be categorized into two main types: I) reducing the number of steps via the faster schedulers (Lu et al. 2022) or step distillation (Sauer et al. 2023); II) reducing the computation per step through

*Corresponding author.

model pruning and distillation (Segmend 2023), or by mitigating redundant computation (Ma, Fang, and Wang 2024). While these methods achieve notable acceleration gains, they typically adopt a "one-size-fits-all" strategy, *i.e*. applying the same number of steps without considering the rich complexity of textual prompts.

In light of this, we introduce AdaDiff, an end-to-end framework that aims to achieve efficient diffusion models by learning adaptive step selection in the denoising process based on prompts. For each prompt, deriving a dynamic generation strategy involves: I) determining the required number of steps for generation and II) ensuring high-quality generation even with a relatively smaller number of steps. With this, AdaDiff can allocate more computational resources to more descriptive prompts while using fewer resources for simpler ones. While this approach is highly appealing, learning the dynamic step selection is a non-trivial task, as it involves non-differentiable decision-making processes.

To address this challenge, AdaDiff is built upon a reinforcement learning framework. Specifically, given a prompt, AdaDiff trains a lightweight step selection network to produce a policy for step usage. Subsequently, based on this derived policy, a dynamic sampling process is performed on a pre-trained diffusion model for efficient generation. The step selection network is optimized using a policy gradient method to maximize a meticulously crafted reward function. The primary objective of this reward function is to encourage the generation of high-quality visual content while minimizing computational resources. It is also worth pointing out that the step selection network conditioned on textual inputs is lightweight with negligible computational overhead.

We conduct extensive experiments to evaluate our proposed method, and the results demonstrate that AdaDiff saves between 33% and 40% of inference time compared to the baseline using a fixed denoising step while maintaining similar visual quality across various image and video generation benchmarks (Lin et al. 2014; Schuhmann et al. 2022; Wang et al. 2022; Xu et al. 2016; Wang et al. 2023b). In addition, we demonstrate that our approach can be combined with various existing acceleration paradigms (Sauer et al. 2023; Segmend 2023). Moreover, the learned policy from one dataset can be successfully transferred to another. Finally, through qualitative and quantitative analysis, we show that AdaDiff flexibly allocates fewer steps for less informative prompts and more for informative prompts.

Related Work

Diffusion Models. Diffusion models (Ho, Jain, and Abbeel 2020; Dhariwal and Nichol 2021) have emerged as a powerful force in the field of deep generative models, achieving top-notch performance in various applications, spanning image generation (Rombach et al. 2022; Podell et al. 2024; Li et al. 2024a; Chen et al. 2024; Esser et al. 2024; Li et al. 2024b), video generation (Wang et al. 2023a; Blattmann et al. 2023; OpenAI 2024; Bao et al. 2024), and image restoration (Xia et al. 2023; Gao et al. 2023), among others. Notably, in image and video generation, these diffusion models have demonstrated the ability to produce desired results based on diverse input conditions, including text, semantic maps, representations, and images. However, the inherent iterative nature of the diffusion process has led to a substantial demand for computational resources and inference time during the generation process.

Reinforcement Learning in Diffusion Models. Pre-training objectives of generative models often do not align perfectly with human intent. Therefore, some work focuses on fine-tuning generative models through reinforcement learning (Sutton and Barto 2018) to align their outputs with human preferences, using human feedback or carefully designed reward functions. Typically, these models (Xu et al. 2023; Black et al. 2023; Fan et al. 2023; Wallace et al. 2023; Wu et al. 2023) enhance aspects such as text-to-image alignment, aesthetic quality, and human-perceived image quality. Nevertheless, we are the first to leverage reinforcement learning to accelerate image and video generation by learning an instance-specific step usage policy.

Acceleration of Diffusion Models. Recently, effort has been made to accelerate the reverse process of diffusion models. These approaches can be broadly categorized into two paradigms: reducing the number of sampling steps and reducing the computation per step. The first paradigm focus on designing fewer-step samplers that extract subsequences from the original sequence (Song, Meng, and Ermon 2021; Lu et al. 2022; Zhao et al. 2023) or use knowledge distillation, where a student model learns to approximate a teacher model's output in fewer steps (Meng et al. 2023; Sauer et al. 2023; Luo et al. 2023; Lin, Wang, and Yang 2024; Ren et al. 2024). The second paradigm primarily focuses on model pruning and distillation (Kim et al. 2023; Fang, Ma, and Wang 2023; Segmend 2023) or reducing redundant computations during the denoising process (Ma, Fang, and Wang 2024; Wimbauer et al. 2024; Li et al. 2023). The proposed AdaDiff method is *orthogonal* to, yet *complementary* to, the aforementioned acceleration methods. While previous methods reduce the total number of sampling steps, they still rely on a manually set and "one-size-fits-all" denoising step, ignoring the complexity in textual prompts that are used to condition the generation process. Instead, AdaDiff aims to dynamically decide the optimal step that achieves a balance between visual quality and inference time on a per-input basis, which can be combined with existing speed-up methods.

Methodology

AdaDiff reduces computational cost and inference time for diffusion models by learning step usage policies conditioned on prompts. The intuition is to encourage using fewer denoising steps while generating high-quality results. In the following, we will first review the background knowledge of diffusion models. Subsequently, we will delve into the components of AdaDiff and how it facilitates adaptive generation, including image and video generation.

Background on diffusion models

Diffusion modes achieve new state-of-the-art performance in the field of deep generative models inspired by the principles of equilibrium thermodynamics. Specifically, diffusion models involve a forward process where noise is gradually

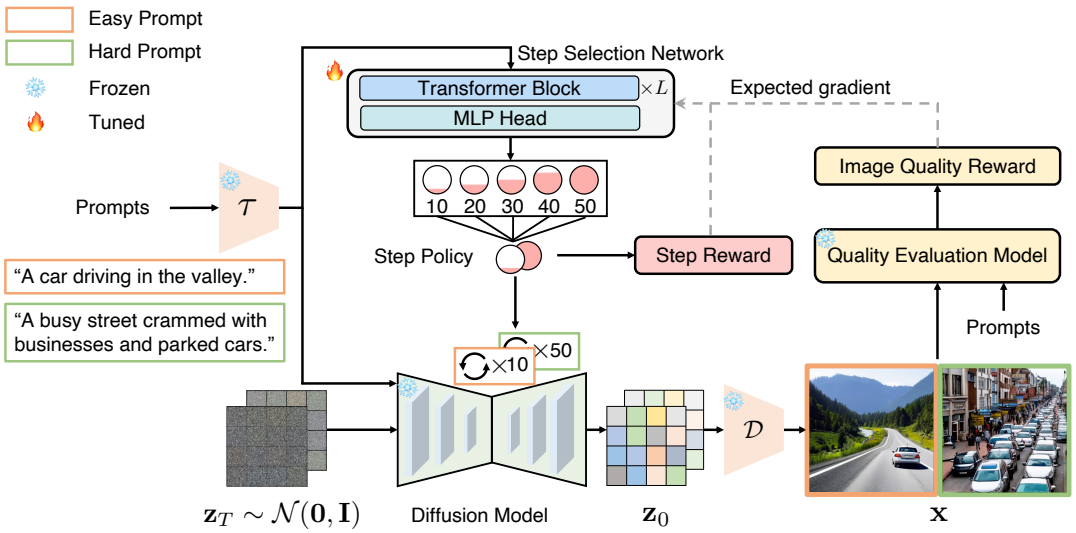

Figure 2: **An overview of AdaDiff**. Given the input prompts, the step selection network learns the information richness of each prompt and derives the corresponding step usage policy. These policies determine the number of steps required for the diffusion model to generate images. Subsequently, the reward function balances the trade-off between speed and image quality.

added to the input and a reverse process that learns to recover the desired noise-free data from noisy data. In the forward process, the posterior probability of the diffusion image x_t at time step t has a closed form:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{1}$$

where $\bar{\alpha}_t = \prod_{i=0}^{t} \alpha_i = \prod_{i=0}^{t}(1-\beta_i)$ and $\beta_i \in (0,1)$ represents the noise variance schedule. Once the diffusion model $\epsilon_\theta(x_t)$ is trained, during the reverse process, traditional diffusion models like DDPM (Ho, Jain, and Abbeel 2020) denoise $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ step by step for a total of T steps. One can also use a discrete-time DDIM (Song, Meng, and Ermon 2021) sampler to speed up the sampling process. Initially, the total number of sampling steps S is predetermined. Sampling updates are performed at every $\lceil T/S \rceil$ steps according to the plan, reducing the original T steps to a new sampling plan that consists of a subset of S diffusion steps $\hat{T} = \{\eta_1, \ldots, \eta_S\}$. DDIM sampler significantly reduces the number of sampling steps and is widely employed in various generative tasks.

Latent Diffusion Models (LDMs). LDMs employ an approach where the diffusion process operates in the latent space rather than the pixel space. This reduces the training cost and improves the inference speed. It uses the pre-trained encoder $\mathcal{E}$ and decoder $\mathcal{D}$ to encode the pixel-space image into a low-dimensional latent and decode the latent back into the image, respectively. In addition, LDMs incorporate flexible conditional information, such as text conditions and semantic maps, through a cross-attention mechanism to guide the visual generation process. For instance, Stable Diffusion and ModelScopeT2V have been influential approaches for image and video generation conditioned on texts. They utilize a DDIM sampler during generation, with a default of 50 steps for all prompts. In this paper, AdaDiff aims to enhance the inference speed of LDMs by implementing dynamic step selection on a per-input basis.

Adaptive step selection for image generation

In image generation, AdaDiff learns a step usage policy conditioned on the text description to reduce the denoising step of Stable Diffusion such that steps vary for different prompts. To this end, AdaDiff builds upon a lightweight selection network trained to determine the total number of steps $\mathbf{t}$ of the DDIM sampler used in the reverse process of Stable Diffusion, as shown in 2. The selection network makes non-differentiable categorical decisions, *i.e.*, steps to be used.

We design N distinct schedulers for the DDIM sampler, each corresponding to a specific total number of sampling steps. In this paper, unless specified otherwise, we set $N = 10$, which corresponds to the set of step values $\mathcal{S} = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. The state space is defined as the input prompts, and actions in the model involve categorizing them in these discrete action spaces. Then, a carefully designed reward function balances the quality of the generated images with the computational cost. Formally, given a prompt $\mathbf{p}$, Stable Diffusion first uses a text encoder $\boldsymbol{\tau}$ to extract the text features, denoted as $\mathbf{c} = \boldsymbol{\tau}(\mathbf{p})$. Following this, the step selection network f_s, parameterized by $\mathbf{w}$, learns the informativeness of $\mathbf{c}$ using self-attention mechanisms and then further maps it to $\mathbf{s} \in \mathbb{R}^N$ through a Multi-Layer Perceptron (MLP):

$$\mathbf{s} = f_s(\mathbf{c}; \mathbf{w}), \tag{2}$$

where each entity in $\mathbf{s}$ indicates the probability score of choosing this step. We then define a step selection policy $\pi^f(\mathbf{u} \mid \mathbf{p})$ with a N-dimensional Categorical Distribution. Here, $\mathbf{u}$ is a one-hot vector of length N, denoted as $\mathbf{u} \in \{0,1\}^N$, and $\mathbf{u}_j = 1$ indicates that the step $\mathbf{t}$ with index j in $\mathcal{S}$ is selected. During training, $\mathbf{u}$ is sampled from the corresponding policy, and during testing, a greedy approach is employed.

So far, the total number of steps $\mathbf{t}$ for DDIM sampling is determined on a per-prompt $\mathbf{p}$ basis. Then, Stable Diffusion

starts with a latent $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, fuses the text-condition features $\mathbf{c}$ through cross-attention, and further generates the desired latent $\mathbf{z}_0$ through $\mathbf{t}$ denoising steps. Finally, the decoder $\mathcal{D}$ decodes the latent into a pixel-space generated image $\mathbf{x}$. This process is formalized as:

$$\mathbf{x} = \mathcal{D}(LDM^{\mathbf{p}\to\mathbf{t}}_{sampler}(\mathbf{z}_T, \mathbf{c})). \tag{3}$$

Recall that the primary objective of AdaDiff is to generate high-quality images in a smaller number of sampling steps, hence, it is crucial to design a suitable reward function to evaluate these actions efficiently. The reward function consists primarily of two parts: an image quality reward and a step reward, balancing quality and inference time. Firstly, to assess image quality, we leverage a quality evaluation model f_q specifically designed to evaluate the quality of generated images (Xu et al. 2023), abbreviated as the IQS model. This model assesses image quality in two dimensions: image-text alignment and perceptual fidelity. Image-text alignment means that the generated image should match the user-provided text conditions, while perceptual fidelity means that the generated image should be faithful to the shape and characteristics of the object rather than being generated chaotically. Typically, the higher the IQS score $f_q(\mathbf{x})$, the higher the quality of the generated image $\mathbf{x}$. Thus, in this paper, we design the image quality reward as $\mathcal{Q}(\mathbf{u}) = f_q(\mathbf{x})$.

For the step reward, we define it as $\mathcal{O}(\mathbf{u}) = 1 - \frac{\mathbf{t}}{S_{max}}$, which represents the normalized steps saved relative to the maximum steps in $\mathcal{S}$. Finally, the overall reward function is formalized as follows:

$$R(\mathbf{u}) = \begin{cases} \mathcal{O}(\mathbf{u}) + \lambda\mathcal{Q}(\mathbf{u}) & \text{for high quality image} \\ -\gamma & \text{else} \end{cases} \tag{4}$$

where λ is a hyperparameter that controls the effect of image quality reward $\mathcal{Q}(\mathbf{u})$ and γ is the penalty imposed on the reward function when the generated image quality is low. Instead of a straightforward comparison between the image quality score $f_q(\mathbf{x})$ and a predefined threshold (*e.g.*, 0) to discern whether $\mathbf{x}$ is a high-quality image, we design the determination as a relative manner. Specifically, for a given prompt, we individually generate an image for each step in the step set $\mathcal{S}$ and we consider the image quality score to be high if it ranks top k among these ten images (we empirically set k to 3). At this point, the step selection network can be optimized to maximize the expected reward:

$$\max_{\mathbf{w}} \mathcal{L} = \mathbb{E}_{\mathbf{u}\sim\pi_f} R(\mathbf{u}). \tag{5}$$

In this paper, we use the policy gradient method (Sutton and Barto 2018) to learn the parameters $\mathbf{w}$ for the step selection network. The expected gradient can be derived as follows:

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbb{E}\left[R(\mathbf{u})\nabla_{\mathbf{w}} \log \pi^f(\mathbf{u} \mid \mathbf{p})\right], \tag{6}$$

which is further approximated with Monte-Carlo sampling using mini-batches:

$$\nabla_{\mathbf{w}}\mathcal{L} \approx \frac{1}{B}\sum_{i=1}^{B}\left[R\left(\mathbf{u}_i\right)\nabla_{\mathbf{w}} \log \pi^f\left(\mathbf{u}_i \mid \mathbf{p}_i\right)\right]. \tag{7}$$

where B is the total number of prompts in the mini-batch. The gradient is then propagated back to train the step selection network using the Adam optimizer.

Following the aforementioned training process, the selection network learns the step usage policy that strikes a balance between inference time and generation quality. During the inference phase, for different prompts, the maximum probability score in $\mathbf{s}$ is used to determine the number of generation steps, enabling dynamic inference.

Adaptive step selection for video generation

In addition to image generation, the proposed step selection strategy can also be applied to video diffusion models, such as ModelScopeT2V, as the prompts used to guide video generation also have varying levels of richness. The overall implementation paradigm is similar to Figure 2. To assess the quality of the generated videos $\mathcal{V}$, we individually score each frame using the IQS model and subsequently calculate the average of all frames to obtain the video quality reward. This procedure can be formalized as $\mathcal{Q}(u) = \frac{1}{F}\sum_{i=1}^{F} f_q(\mathcal{V}_i)$. Here, F is the number of frames in the generated video. Then, we calculate the step reward by $\mathcal{O}(\mathbf{u}) = 1 - \frac{\mathbf{t}}{S_{max}}$, and the overall reward is obtained from Eq.(4).

Experiments

Experimental Details

Datasets. To evaluate the effectiveness and generalizability of our approach, we conduct extensive experiments on three image datasets: MS COCO 2017 (Lin et al. 2014), Laion-COCO (Schuhmann et al. 2022), DiffusionDB (Wang et al. 2022), and two video datasets: MSR-VTT (Xu et al. 2016) and InternVid (Wang et al. 2023b). In MS COCO 2017, our training set consists of $118,287$ textual descriptions, and all $25,014$ text-image pairs from the validation set are employed for testing. Regarding Laion-COCO, we randomly select $200K$ textual descriptions for training and $20K$ text-image pairs for testing. The partitioning of the training and testing sets for DiffusionDB follows the same paradigm as Laion-COCO. The training sets for MSR-VTT and InternVid consist of $6,651$ and $24,911$ text descriptions, respectively. The test set for MSR-VTT comprises $2,870$ text-video pairs.

Evaluation Metrics. Following previous work, we assess the quality of generated images or videos using several metrics, including FID (Heusel et al. 2017), IS (Salimans et al. 2016), CLIP Score (Radford et al. 2021), NIQE (Mittal, Soundararajan, and Bovik 2012), and the recently introduced Image Quality Score (IQS) (Xu et al. 2023). Additionally, we use the average denoising steps and time per image or video generation to measure the inference speed.

Implementation Details. We design the step selection network as a lightweight architecture consisting of three self-attention layers and a multi-layer perceptron. For image generation, we use SD-v2.1-base and SDXL-Turbo to generate 512×512 images, and SDXL-v1.0 for 1024×1024 images. For video generation, we use ModelScopeT2V to generate 16-frame videos at a resolution of 256×256. The parameters and computational cost of the step selection network are 25.71M and 1.93 GFLOPs, respectively, which are negligible

compared to SD-v2.1-base's 865M and the 35,140 GFLOPs required to generate an image in 50 steps. We train the step selection network for 200 epochs with a batch size of 256 and use the Adam optimizer with an initial learning rate of 10^{-5}. The training cost ranges from 16 to 80 A100 GPU hours for the adaptive step policy applied to different base models.

Main Results

Performance on Image and Video Generation. To validate the effectiveness and general applicability of AdaDiff, we compare it with the following baseline methods:

- *One-size-fits-all*: The base model uses a fixed number of sampling steps for different prompts. We primarily chose two step counts: the default number (typically 50) and a count close to the speed of AdaDiff.
- *Random*: Given the step usage policies produced by AdaDiff, we generate random step policies to validate the effectiveness of learned policies.
- *Heuristic*: We propose a heuristic step usage policy, allocating more generation steps for prompts containing more words. For example, for words <8, we allocate 10 steps; for $8 \leq$ words <10, we allocate 20 steps, and so on.
- *Perplexity*: We further propose a step usage strategy, which allocates more steps for prompts with higher perplexity (Jelinek et al. 1977). For example, for perplexity <20, we allocate 10 steps; for $20 \leq$ perplexity <40, we allocate 20 steps, and so on.

COCO	Speed		Image Quality				
	Step↓	Time↓	IQS↑	CLIP↑	IS↑	FID↓	NIQE↓
SD-v2.1	50	2.24	0.419	0.314	37.48	22.13	3.75
SD-v2.1	28	1.28	0.362	0.312	37.35	22.50	3.88
Random	30.03	1.41	0.354	0.313	36.85	22.50	3.88
Heuristic	29.75	1.40	0.369	0.313	36.72	22.73	3.96
Perplexity	31.79	1.48	0.368	0.313	37.33	22.77	3.86
AdaDiff	28.61	1.35 (↑39.7%)	**0.412**	**0.314**	**37.60**	**21.92**	**3.76**

Table 1: Comparison of AdaDiff on image generation.

Table 1 offers a detailed analysis of AdaDiff's performance on the COCO-2017 benchmarks, with results averaged over five independent runs. AdaDiff assigns average sampling step counts of 28.61. Compared to the fixed 50-step SD, AdaDiff achieves similar performance across five image quality metrics while providing a 39.7% speed increase. At comparable speeds, AdaDiff significantly surpasses the fixed 28-step SD in image quality. Furthermore, the step usage policy learned by AdaDiff shows advantages over the random policy and manually crafted policies like heuristic and perplexity-based approaches. These findings validate that AdaDiff efficiently generates instance-specific step usage policies, allocating different sampling step counts per prompt to optimize speed with minimal loss in image quality.

Table 2 confirms the effectiveness of AdaDiff for video generation tasks. Compared to ModelScope's fixed 50-step outputs, AdaDiff not only enhances video quality but also reduces the generation time per video by 35.8% (13.6 vs 21.2). When compared to the fixed 31-step, random, heuristic, and perplexity-based policies, AdaDiff uses similar computational resources but significantly enhances video quality across all metrics. These results demonstrate the efficacy of AdaDiff in video generation and its potential to be applied across various text-conditioned diffusion models.

MSR-VTT	Speed		Video Quality				
	Step↓	Time↓	IQS↑	CLIP↑	IS↑	FID↓	NIQE↓
ModelScope	50	21.2	-0.518	0.293	18.79	44.85	6.37
ModelScope	31	13.1	-0.678	0.293	18.42	46.09	6.52
Random	29.98	13.5	-0.723	0.293	18.22	47.41	6.75
Heuristic	28.19	13.2	-0.709	0.293	18.32	46.45	6.59
Perplexity	28.86	13.3	-0.685	0.294	18.39	46.01	6.65
AdaDiff	31.14	13.6 (↑35.8%)	**-0.532**	**0.294**	**18.71**	**45.02**	**6.42**

Table 2: Comparison of AdaDiff on video generation.

COCO	Speed		Image Quality			
	Step↓	Time↓	IQS↑	IS↑	FID↓	NIQE↓
SDXL-Euler	50	5.59	0.692	36.43	24.13	3.83
Reduce the number of steps						
SDXL-DPMSolver	29	3.24	0.634	35.52	24.49	3.95
SDXL-Lightning	8	0.92	0.652	36.14	27.75	3.85
Reduce the computation per step						
SSD-1B	50	3.60	0.588	35.03	28.99	3.95
DeepCache(N=2)	50	3.21	0.636	34.99	24.76	4.07
AdaDiff-SDXL	29.16	3.31 (↑40.8%)	**0.678**	**36.16**	**24.31**	3.88

Table 3: Comparison with other acceleration methods

Compared with other acceleration methods. In Table 3, we compare AdaDiff with two other acceleration paradigms, ensuring a fair comparison by using the same base model and benchmark—SDXL and the COCO-2017 5k validation set. In comparison with the first paradigm, which reduces the number of steps, AdaDiff outperforms the faster DPMSolver sampler in quality at a similar speed. Compared to step distillation methods like SDXL-Lightning, our method achieves better performance with significantly lower training costs (80 vs 1000+ A100 GPU hours). Although AdaDiff uses more average steps, it can be used on top of step distillation methods, which we discuss later. In the comparison with the second paradigm, which focuses on reducing the computation per step, AdaDiff achieves better image quality at similar speeds compared to methods based on model pruning and distillation like SSD-1B, as well as those that save computational efforts such as DeepCache.

Extension AdaDiff to other acceleration methods. Although previous methods achieve notable acceleration gains, they still adopt a "one-size-fits-all" strategy, *i.e.* applying the same number of steps regardless of the complexity of the prompts. In light of this, AdaDiff can be used on top of these methods to achieve further acceleration. Firstly, we apply AdaDiff to the paradigm of reducing the number of steps, *i.e.* SDXL-Turbo. As shown in Table 4, compared to the fixed 5-step results, AdaDiff achieves comparable performance with a 52.7% speed up. Compared to the fixed 2-step, AdaDiff

COCO	Speed		Image Quality				
	Step↓	Time↓	IQS↑	CLIP↑	IS↑	FID↓	NIQE↓
SDXL-Turbo	5	0.55	0.801	0.315	43.71	26.12	4.01
SDXL-Turbo	2	0.24	0.754	0.313	43.37	28.42	4.17
Random	3.01	0.35	0.761	0.313	43.39	27.27	4.13
Heuristic	2.97	0.33	0.767	0.313	43.31	27.17	4.11
Perplexity	3.18	0.38	0.772	0.314	43.42	26.80	4.11
AdaDiff	2.19	0.26 (↑52.7%)	**0.791**	**0.314**	**43.68**	**26.71**	**4.06**

Table 4: Extension AdaDiff to SDXL-Turbo.

clearly provides better image quality at similar speeds. Additionally, we apply AdaDiff to the paradigm of reducing the computation per step, such as SSD-1B. As shown in Table 5, AdaDiff achieves 41.1% speed up compared to fixed 50-steps and outperformed the fixed 30-step, random, heuristic, and perplexity policies at similar speeds. These results confirm that AdaDiff is orthogonal to, yet complementary to, previous acceleration methods, offering broad applicability.

COCO	Speed		Image Quality				
	Step↓	Time↓	IQS↑	CLIP↑	IS↑	FID↓	NIQE↓
SSD-1B	50	3.60	0.588	0.336	35.03	28.99	3.95
SSD-1B	29	2.09	0.537	0.334	34.18	29.31	4.01
Random	30.09	2.23	0.522	0.334	34.16	29.29	4.12
Heuristic	29.69	2.18	0.531	0.335	34.21	29.33	4.08
Perplexity	30.11	2.26	0.538	0.335	34.25	29.28	4.06
AdaDiff	29.16	2.12 (↑41.1%)	**0.571**	**0.336**	**34.77**	**28.83**	**3.99**

Table 5: Extension AdaDiff to SSD-1B.

	Speed		Image / Video Quality				
	Step↓	Time↓	IQS↑	CLIP↑	IS↑	FID↓	NIQE↓
COCO 2017 → Laion-COCO							
SD	50	2.27	0.350	0.319	30.71	22.08	4.58
AdaDiff	30.50	1.38 (↑39.2%)	0.341	0.320	30.61	22.15	4.62
InternVid → MSR-VTT							
ModelScope	50	21.20	-0.518	0.293	18.79	44.85	6.37
AdaDiff	32.23	14.03 (↑33.8%)	-0.521	0.292	18.76	45.01	6.45

Table 6: Validation on zero-shot adaptive generation.

Extension to other datasets. We also evaluate AdaDiff's ability to generalize its learned step selection policy from one dataset to another, which we refer to as zero-shot generation performance. For image generation, we evaluate the step usage policy derived from COCO 2017 on Laion-COCO, while for video generation, we use the policy of InternVid for validation on MSR-VTT. As shown in Table 6, the dynamic strategy saves 39.2% and 33.8% of generation time on the two datasets separately compared to the fixed 50-step generation while maintaining comparable generation quality. These results demonstrate the transferability of step selection strategies trained on large-scale data.

Analyses of learned policies. To better understand the learned policy of AdaDiff, we investigate the relationship between step selection and prompt richness through Figure 3. We evaluate prompt richness in terms of the number of words and objects, assuming an overall increase in information richness as both grow. We observe that as the number of words in the prompt increases, the generated results may include more detailed descriptions, spatial relationships, attribute definitions, *etc*. Consequently, AdaDiff allocates more denoising steps for these richness prompts. Besides, as the number of objects in the prompt increases, the results involve a larger amount of details and interactions among the objects. As a result, AdaDiff also assigns more steps for these prompts.

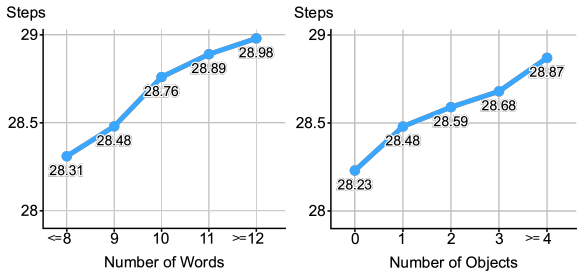

Figure 3: Analyze the learned step policy based on the number of words and objects in the prompts. AdaDiff tends to assign more steps to more informative prompts.

Qualitative Results. We further qualitatively analyze our approach as shown in Figure 4. We specifically examine five distinct scenarios {indoor, food, animal, outdoor, and sports}, and explore the impact of the richness of the prompts on the step usage policy within the same scenario. Our observations reveal that in instances where the prompt is straightforward (easy, Figure 4), typically involving only one or a few objects, AdaDiff assigns 10 to 20 steps to obtain satisfactory generated results. For prompts characterized by an increased number of objects or the inclusion of detailed descriptions (medium, Figure 4), such as "multiple pizzas" or "ankle-deep", AdaDiff allocates a higher number of steps, typically around 30. In the case of challenging prompts (hard, Figure 4), which often involve numerous objects, intricate interactions between them, and diverse detailed descriptions, AdaDiff allocates 40 to 50 steps to achieve satisfactory generation.

Discussion

Reward function. The core idea of the reward function is to generate images via fewer steps while maintaining quality. Table 7 showcases various designs of the reward function:

- *Focus only on step savings*: The step selection strategy tends to use fewer steps but ignores quality degradation.
- *Integrating step savings with various image quality metrics*: Incorporating image quality metrics such as CLIP, NIQE, and IQS scores into the reward function enhances the step selection strategy's sensitivity to image quality in varying degrees. IQS score considers both perceptual fidelity and text-image alignment, significantly improving the strategy's effectiveness. In contrast, focusing solely on text-image alignment (CLIP score) or perceptual fidelity (NIQE score) may lead to quality decline due to the incomplete assessment perspective.

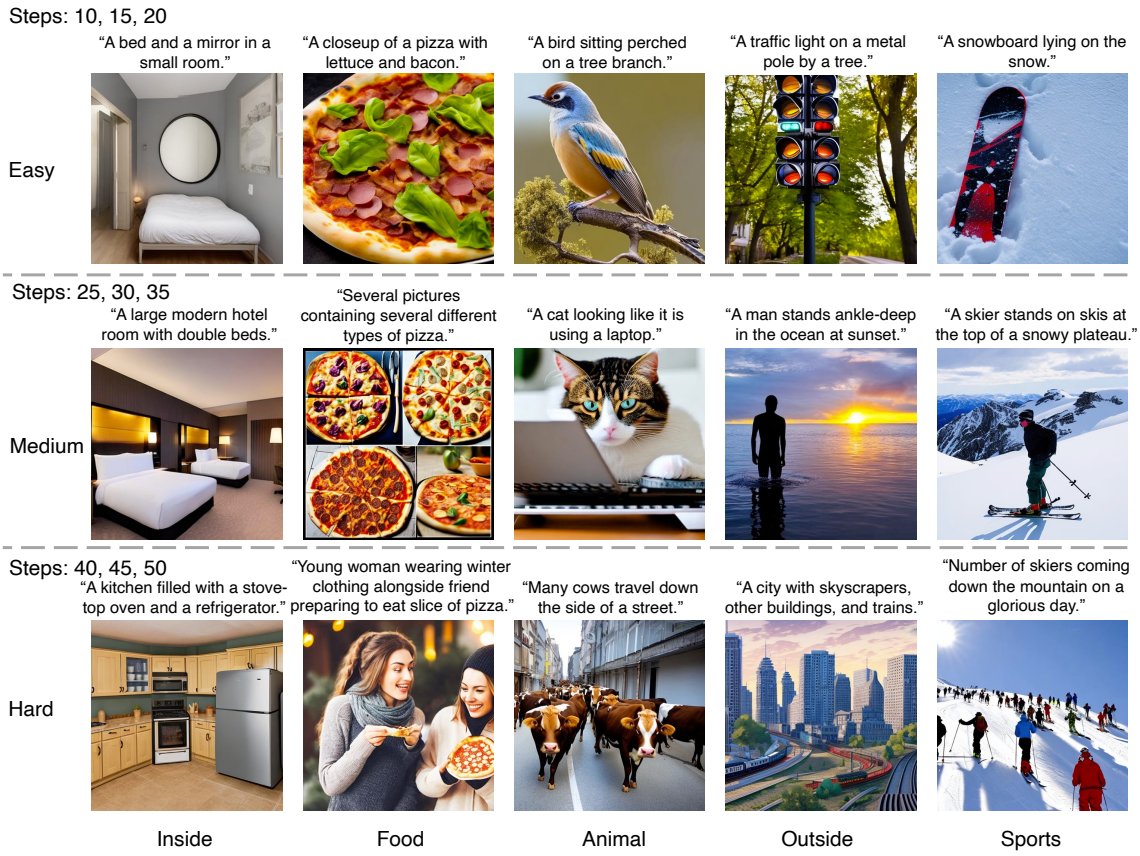

Figure 4: **Qualitative Results.** AdaDiff implements an instance-specific dynamic generation based on the prompt complexity.

- *Different criteria for determining whether an image is high quality*: The relative approach ranks images generated by different steps using IQS scores and defines the top-k images as high quality. In contrast, the absolute approach defines images as high quality if their IQS score exceeds a manually set threshold (*e.g.* 0). The findings in Table 7 indicate that the relative approach learns a more efficient step policy.

Reward function						Performance			
Step	CLIP	NIQE	IQS	Top-k	Threshold	Step↓	IQS↑	IS↑	NIQE↓
✓				✓		18.12	0.335	36.58	3.96
✓	✓			✓		18.86	0.334	36.53	3.97
✓		✓		✓		29.62	0.392	37.11	3.79
✓			✓	✓		28.61	**0.412**	**37.60**	**3.76**
✓			✓		✓	25.23	0.377	37.25	3.91

Table 7: Comparisons of different reward functions.

Different trade-offs between speed and quality. The hyperparameters top-k and image reward weight λ modulate different speed-quality trade-offs, as shown in Figure 5. Smaller values of k and larger values of λ enforce higher standards for image quality, resulting in generated images with higher visual quality and better prompt following, accompanied by an increase in the average number of steps.

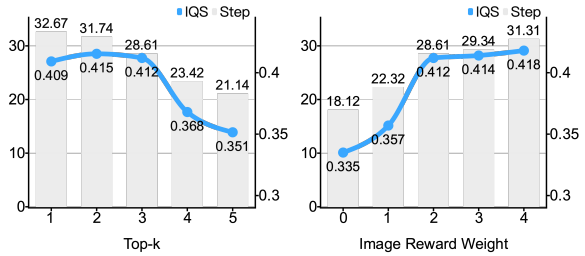

Figure 5: The speed and quality trade-offs modulated by top-k and image reward weight.

Conclusion

In this paper, we introduced AdaDiff, a method that derives adaptive step usage policies tailored to each prompt, facilitating efficient image and video generation. More precisely, a step selection network is trained using policy gradient methods to generate these policies, striking a balance between generation quality and the reduction of overall computational costs. Extensive experiments validated the capability of AdaDiff to generate strong step usage policies on per-input bias, providing compelling qualitative and quantitative evidence. Additionally, AdaDiff can be used on top of other acceleration methods to provide further speed benefits.

Acknowledgements

This project was supported by the National Natural Science Foundation of China under Grant No. 2021ZD0112805.

References

Bao, F.; Xiang, C.; Yue, G.; He, G.; Zhu, H.; Zheng, K.; Zhao, M.; Liu, S.; Wang, Y.; and Zhu, J. 2024. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*.

Black, K.; Janner, M.; Du, Y.; Kostrikov, I.; and Levine, S. 2023. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*.

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*.

Chen, J.; Yu, J.; Ge, C.; Yao, L.; Xie, E.; Wu, Y.; Wang, Z.; Kwok, J.; Luo, P.; Lu, H.; et al. 2024. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. In *ICLR*.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. In *NeurIPS*.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*.

Fan, Y.; Watkins, O.; Du, Y.; Liu, H.; Ryu, M.; Boutilier, C.; Abbeel, P.; Ghavamzadeh, M.; Lee, K.; and Lee, K. 2023. DPOK: Reinforcement Learning for Fine-tuning Text-to-Image Diffusion Models. *arXiv preprint arXiv:2305.16381*.

Fang, G.; Ma, X.; and Wang, X. 2023. Structural pruning for diffusion models. In *NeurIPS*.

Gao, S.; Liu, X.; Zeng, B.; Xu, S.; Li, Y.; Luo, X.; Liu, J.; Zhen, X.; and Zhang, B. 2023. Implicit diffusion models for continuous super-resolution. In *CVPR*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.

Jelinek, F.; Mercer, R. L.; Bahl, L. R.; and Baker, J. K. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*.

Kim, B.-K.; Song, H.-K.; Castells, T.; and Choi, S. 2023. BK-SDM: Architecturally Compressed Stable Diffusion for Efficient Text-to-Image Generation. *ICML Workshop*.

Li, D.; Kamko, A.; Akhgari, E.; Sabet, A.; Xu, L.; and Doshi, S. 2024a. Playground v2. 5: Three Insights towards Enhancing Aesthetic Quality in Text-to-Image Generation. *arXiv preprint arXiv:2402.17245*.

Li, S.; Hu, T.; Khan, F. S.; Li, L.; Yang, S.; Wang, Y.; Cheng, M.-M.; and Yang, J. 2023. Faster diffusion: Rethinking the role of unet encoder in diffusion models. *arXiv preprint arXiv:2312.09608*.

Li, Z.; Zhang, J.; Lin, Q.; Xiong, J.; Long, Y.; Deng, X.; Zhang, Y.; Liu, X.; Huang, M.; Xiao, Z.; et al. 2024b. Hunyuan-DiT: A Powerful Multi-Resolution Diffusion Transformer with Fine-Grained Chinese Understanding. *arXiv preprint arXiv:2405.08748*.

Lin, S.; Wang, A.; and Yang, X. 2024. SDXL-Lightning: Progressive Adversarial Diffusion Distillation. *arXiv preprint arXiv:2402.13929*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*.

Luo, S.; Tan, Y.; Huang, L.; Li, J.; and Zhao, H. 2023. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*.

Ma, X.; Fang, G.; and Wang, X. 2024. Deepcache: Accelerating diffusion models for free. In *CVPR*.

Meng, C.; Rombach, R.; Gao, R.; Kingma, D.; Ermon, S.; Ho, J.; and Salimans, T. 2023. On distillation of guided diffusion models. In *CVPR*.

Mittal, A.; Soundararajan, R.; and Bovik, A. C. 2012. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*.

OpenAI. 2024. Video generation models as world simulators. https://openai.com/sora.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2024. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Ren, Y.; Xia, X.; Lu, Y.; Zhang, J.; Wu, J.; Xie, P.; Wang, X.; and Xiao, X. 2024. Hyper-SD: Trajectory Segmented Consistency Model for Efficient Image Synthesis. *arXiv preprint arXiv:2404.13686*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *NeurIPS*.

Sauer, A.; Lorenz, D.; Blattmann, A.; and Rombach, R. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*.

Schuhmann, C.; Köpf, A.; Vencu, R.; Coombes, T.; and Beaumont, R. 2022. LaionCOCO. https://laion.ai/blog/laion-coco/.

Segmend. 2023. Segmind Stable Diffusion 1B. https://https://huggingface.co/segmind/SSD-1B.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *ICLR*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Wallace, B.; Gokul, A.; Ermon, S.; and Naik, N. 2023. End-to-End Diffusion Latent Optimization Improves Classifier Guidance. In *ICCV*.

Wang, J.; Yuan, H.; Chen, D.; Zhang, Y.; Wang, X.; and Zhang, S. 2023a. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571*.

Wang, Y.; He, Y.; Li, Y.; Li, K.; Yu, J.; Ma, X.; Chen, X.; Wang, Y.; Luo, P.; Liu, Z.; et al. 2023b. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*.

Wang, Z. J.; Montoya, E.; Munechika, D.; Yang, H.; Hoover, B.; and Chau, D. H. 2022. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*.

Wimbauer, F.; Wu, B.; Schoenfeld, E.; Dai, X.; Hou, J.; He, Z.; Sanakoyeu, A.; Zhang, P.; Tsai, S.; Kohler, J.; et al. 2024. Cache me if you can: Accelerating diffusion models through block caching. In *CVPR*.

Wu, X.; Sun, K.; Zhu, F.; Zhao, R.; and Li, H. 2023. Human Preference Score: Better Aligning Text-to-Image Models with Human Preference. In *ICCV*.

Xia, B.; Zhang, Y.; Wang, S.; Wang, Y.; Wu, X.; Tian, Y.; Yang, W.; and Van Gool, L. 2023. Diffir: Efficient diffusion model for image restoration. In *ICCV*.

Xu, J.; Liu, X.; Wu, Y.; Tong, Y.; Li, Q.; Ding, M.; Tang, J.; and Dong, Y. 2023. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *NeurIPS*.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*.

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*.