Diffusion Model Compression for Image-to-Image Translation

Geonung Kim, Beomsu Kim, Eunhyeok Park, and Sunghyun Cho

POSTECH
{k2woong92,qjatn0120,eh.park,s.cho}@postech.ac.kr
https://kimgeonung.github.io/id-compression

Abstract. As recent advances in large-scale Text-to-Image (T2I) diffusion models have yielded remarkable high-quality image generation, diverse downstream Image-to-Image (I2I) applications have emerged. Despite the impressive results achieved by these I2I models, their practical utility is hampered by their large model size and the computational burden of the iterative denoising process. In this paper, we propose a novel compression method tailored for diffusion-based I2I models. Based on the observations that the image conditions of I2I models already provide rich information on image structures, and that the time steps with a larger impact tend to be biased, we develop surprisingly simple yet effective approaches for reducing the model size and latency. We validate the effectiveness of our method on three representative I2I tasks: InstructPix2Pix for image editing, StableSR for image restoration, and ControlNet for image-conditional image generation. Our approach achieves satisfactory output quality with 39.2%, 56.4% and 39.2% reduction in model footprint, as well as 81.4%, 68.7% and 31.1% decrease in latency to InstructPix2Pix, StableSR and ControlNet, respectively.

Keywords: Diffusion model compression · Image-to-Image translation

1 Introduction

In the advent of large-scale text-to-image (T2I) diffusion models such as DALL-E [50], Stable Diffusion [53], and Imagen [56], there has been a dramatic improvement in image generation quality. This achievement has consequently opened up new opportunities across diverse applications, including image restoration [35, 69], image composition [16, 41, 58], image editing [6, 20, 51, 67, 70], conditional image synthesis [3, 17, 44, 74, 76–78], panorama generation [5, 80], personalized generation [55], creature generation [52], and even 3D generation [9, 33, 48, 68]. While these applications employing T2I models demonstrate unprecedented high-quality results, the extremely large parameter size combined with an iterative denoising process necessitates substantial computational resources, thus limiting their practicality. For instance, typical restoration networks generate images with fewer than 80 million parameters in a single feedforward pass [7, 8, 71, 72]. Meanwhile, StableSR [69], which utilizes Stable Diffusion [53] for higher-quality

image restoration, requires approximately 916 million parameters and at least 40 times longer latency, which is an unaffordable trade-off in many cases.

Recently, numerous diffusion model compression approaches have been actively explored to reduce the memory and computational requirements of diffusion models. These approaches can be roughly categorized into two topics: reducing the number of denoising iterations [31,37,40,43,57,63,73] and reducing model footprint [14, 27]. By focusing on the innate characteristics of diffusion models, these studies have proposed diverse task-agnostic optimization techniques. However, their compression performance remains insufficient for practical use in downstream tasks, and the potential for more effective compression methods applicable to I2I tasks has yet to be explored at all.

In this work, we introduce a novel approach to reduce both memory footprint and latency of diffusion models for downstream Image-to-Image (I2I) applications. While T2I diffusion models are designed to synthesize images including both their structures and details from random noise, downstream I2I translation tasks, such as image restoration, use input images that provide substantial guidance on the structures of output images. This offers significant potential for more aggressive compression of diffusion models beyond what existing task-agnostic methods offer, yet this potential remains unexplored so far. Therefore, we explore such potential, and present a practical solution for I2I tasks that provides significant benefits in both latency and memory footprint over existing task-agnostic techniques while requiring a minimal cost.

Our approach comprises two surprisingly simple but effective components: depth-skip pruning and time-step optimization for reducing model size and latency, respectively. Regarding depth-skip pruning, we first empirically verify that coarse layers of the denoising U-Net of a diffusion model, which primarily corresponds to coarse-grained features, contribute less to the output of downstream I2I operations. Based on this, depth-skip pruning carefully prune less contributing coarse layers and fine-tune the model to effectively reduce the model size.

The time-step optimization method reduces the latency by searching for a reduced sequence of time steps of denoising iterations. Specifically, the time-step optimization method searches for an optimal time-step sequence that produce high-quality outputs for a given number of time steps. Unfortunately, finding an optimal time-step sequence is a challenging optimization problem as it involves integer variables, a nonlinear objective function, and a huge search space. To overcome this challenge, AutoDiffusion [31] adopts the genetic algorithm, but it costs a huge amount of search time of a few days, and is prone to local minima. Xue et al. [75] mathematically derive a highly simplified approximation of the objective function, which can be optimized efficiently to find an approximate solution. However, due to the simplicity of the approximation, their approach tends to find less optimal solutions.

For effective search for an optimal time-step sequence, our approach is based on the following intuition: depending on the I2I task, the distribution of the time steps with large impacts tend to be biased towards either the beginning or end of the iterative diffusion process as will be further discussed in Sec. 3.3. Based

on this intuition, we propose an extremely simple approach that aggressively reduces the search space to efficiently find an optimal time-step sequence. Despite its simplicity, our experiments show that our approach achieves higher-quality results than previous approaches.

To reduce both memory footprint and latency, we apply depth-skip pruning and time-step optimization sequentially. Our experiments show that the combination of the proposed depth-skip pruning and time-step optimization achieves satisfactory output quality with 60.8% of parameters and 18.6% of latency in InstructPix2Pix (IP2P) [6], 43.6% of parameters and 31.3% of latency in StableSR [69], and 60.8% of parameters and 68.9% of latency in ControlNet [78] with canny-edge image as a condition input, respectively.

2 Related Work

I2I Downstream Tasks based on T2I Diffusion Models Thanks to the rich generative power of large-scale T2I models, transferring their generation capability to downstream I2I tasks have achieved state-of-the-art performance in various domains such as image inpainting [53], depth-conditioned generation [53], image restoration [35, 69], image editing [6], and conditional image synthesis [44, 78]. These downstream methods utilize the entire parameters and the complete denoising process, despite the relative simplicity of their tasks compared to the pure generation task that starts from Gaussian noise without any guidance images. In this paper, we explore the compression potential of these I2I models, taking into account both model footprint and denoising iterations.

Model Pruning of Diffusion Models Research on model pruning has primarily focused on pruning the architecture of monolithic end-to-end neural networks for image classification such as Convolutional Neural Networks (CNNs) [18, 19, 29,39] and Vision Transformer [10,24,45,65,66,83]. Thus, applying these methods directly to diffusion models is challenging due to the intricate dynamics between the denoising network and time steps inherent in diffusion models. Recently, a couple of works dedicated to diffusion models have been proposed [14, 27]. Diff-Pruning [14] proposes a channel pruning method for diffusion models, which prunes a fixed amount of channels from each layer of the denoising network without considering the impacts of different layers, which can be different for different tasks. BK-SDM [27] analyzes the impact of different network blocks of Stable Diffusion [53], and proposes three different versions of manually pruned models. However, they do not propose an automatic pruning scheme that can be applied to other diffusion models with other metrics. In contrast to these approaches, our depth-skip pruning is designed with a focus on downstream I2I tasks and offers a more principled approach to identifying redundant network blocks based on the quality constraints of a target task.

Acceleration of Diffusion Models For fast sampling, alternative ODE or SDE samplers have been proposed [4, 37, 40, 63, 79], achieving a dramatic reduc-

tion in the number of iterations to fewer than a hundred. For further acceleration, parallel sampling methods [61, 82] have been proposed, but these methods cannot be directly applied to pretrained foundation models. OMS-DPM [36] and T-stitch [46] present model scheduling methods that are applicable only when multiple pretrained diffusion models of varying sizes are available. One notable branch is step distillation techniques [32, 38, 42, 43, 57, 59], which achieve feasible output quality with significantly small step numbers. Despite their effectiveness, they require a substantial amount of training time, approximately in the order of hundreds of V100 GPU days, to distill the pretrained knowledge of an original diffusion model to an accelerated model.

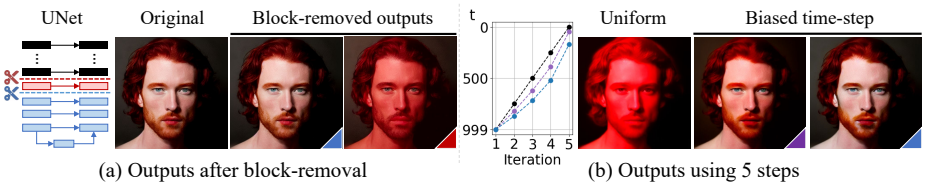
Another line of work is time scheduling approaches such as DDSS [73], AutoDiffusion [31] and Xue et al. [75]. These approaches aim to identify an optimal sequence of time steps within a predetermined number of denoising iterations. Such approaches offer a couple of benefits over the step distillation methods. They are computationally more efficient and simpler to implement, and crucially, preserve the full capabilities of the original model as they do not need to retrain a diffusion model, but allow to use the original model with a reduced time-step sequence. This preservation significantly broadens its applicability to various downstream tasks unlike step distillation approaches. Specifically, step distillation transforms the objective of a diffusion model from progressive denoising into tracking the mean posterior, which can impair the functionality of Classifier-Free Guidance (CFG) [23] that is a vital control parameter in some applications such as IP2P [6]. It is also incompatible with certain downstream tasks that depend on the original denoising process of diffusion models [35, 44, 69, 78]. Our time-step optimization scheme also employs the time scheduling approach to support various downstream tasks.

3 Methods

In this section, we first briefly review the T2I diffusion model and transferred I2I model in Sec. 3.1. We then introduce our depth-skip pruning for effectively reducing the model size in Sec. 3.2, and our time-step optimization method to find an optimal time-step sequence in Sec. 3.3. To reduce both model size and latency, depth-skip pruning and time-step optimization can be performed in any sequence, as the order has a negligible impact, as shown in the Supplemental Document (Tab. S2). In our experiments, we initially perform depth-skip pruning, followed by time-step optimization.

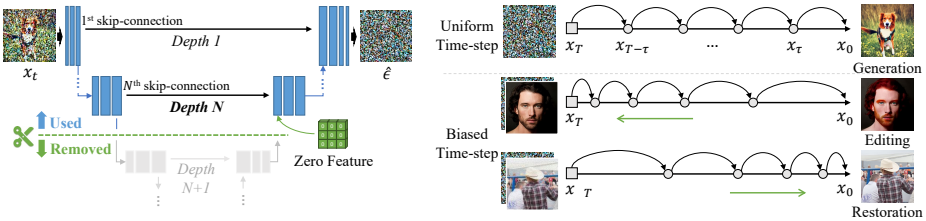
3.1 Diffusion Models

Diffusion models [22, 62, 64] are a class of generative models designed to convert Gaussian noise into a desired sample through iterative refinement using a denoising process guided by a neural network. In T2I diffusion models, a noise prediction network ϵ_θ is employed to estimate the noise component within the



(a) Outputs after block-removal (b) Outputs using 5 steps

Fig. 1: Motivations of our approach. (a) Even after removing the network layers beneath a certain depth, IP2P [6], a downstream I2I model, still produces a plausible result. (b) By focusing on earlier time steps, a feasible output can be obtained using only five denoising steps.



(a) Depth-skip pruning (b) Time-step optimization

Fig. 2: (a) Depth-skip pruning eliminates all layers deeper than a certain depth level, effectively reducing the model size. (b) Given a fixed number of time steps, our time-step optimization finds differently biased time step sequences for different I2I tasks.

input image x_t conditioned on time step t and text prompt $\mathcal{P}$. For downstream I2I tasks utilizing T2I models, the model is retrained using a loss function:

$$\mathcal{L} = ||\epsilon_\theta(x_t, c_I, \mathcal{P}, t) - \epsilon||^2, \tag{1}$$

where c_I represents an additional input image. To accommodate the additional input, these approaches either fine-tune the diffusion model with minor modifications to the input network block [6, 53] or train a feature injection network while keeping the diffusion parameters fixed [44, 69, 78].

3.2 Depth-Skip Pruning

Our depth-skip pruning approach assumes that the denoising network of the target diffusion model is based on the U-Net [54] architecture. In image generation tasks, the coarse layers of the U-Net are primarily responsible for creating the image structure [25, 26, 28]. However, in I2I tasks, the image structure is already provided as input, such as a low-resolution image in image restoration task. As a result, we hypothesize that the deeper layers of the U-Net [54] in the I2I model have a reduced impact on the final output.

To validate this assertion, we conduct an experiment in which we gradually remove the deepest network layers of IP2P [6] and evaluate the output quality. Fig. 1(a) presents the image editing results obtained by the IP2P models after removing the four and five deepest layers of the U-Net, which are indicated in

Table 1: Comparison between the single- and muti-depth search schemes on StableSR [69]. The PSNR values are measured against the outputs obtained without depth-skip pruning.

Baseline (Single depth)				(a) Fix param. & Min time (ΔPSNR < 0.2 dB)	(b) Fix time & Max quality (ΔTime < 1%)		(c) Fix quality & Min time (ΔPSNR < 0.2 dB)	
Depth	PSNR	Time(%)	Param(%)	ΔTime(%)	ΔPSNR	ΔParam(%)	ΔTime(%)	ΔParam(%)
11	32.86	89.91	79.17	-3.46	+0.02	+9.47	-3.46	+0.00
10	32.34	85.70	69.70	-2.62	+0.34	+9.47	-4.59	+9.47
9	31.39	82.78	60.77	-4.27	+1.08	+18.40	-5.72	+8.93
8	28.65	72.59	43.58	-2.23	+1.53	+26.13	-6.29	+26.13

blue and red, respectively. Note that the models are not retrained after removing their layers. As the result shows, even if we simply remove deep layers beneath a certain depth, the output quality remains comparable to that of the original model. This trend persists across different tasks, which indicates that the deep layers have little impact on the output quality. Inspired by this, we introduce a depth-skip pruning which effectively reduces the model size.

The key idea of the proposed method is to *skip* certain network blocks located beyond a predetermined *depth* of the skip-connection of the UNet. For example, in a depth-8 model, denoted as $D8$, the network components beyond the 8th skip-connection are bypassed, as shown in Fig. 2(a). It is noteworthy that coarse-level network blocks typically have numerous channels, leading to substantial memory consumption. Depth-skip pruning removes these bulky blocks and allows us to effectively reduce the memory footprint while minimizing performance loss.

Algorithm 1: Depth-search

Input: input image c_I, prompt $\mathcal{P}$, maximum depth d_{max}, metric function $\mathcal{M}$
Output: Optimal depth d
$d \leftarrow d_{max}$
$x_T \sim \mathcal{N}(0, I)$
repeat
  $d \leftarrow d - 1$
  $x \leftarrow \text{Sampler}_{DDIM}(x_T, c_I, \mathcal{P}; d)$
  $m \leftarrow \mathcal{M}(x)$
until OverThreshold(m);

Our depth-skip pruning consists of two steps: depth-search and fine-tuning. In the depth-search step, we identify the target depth level for pruning by performing depth-skip from the deepest level upwards, using a predefined metric and threshold, until the quality threshold is met, as described in Alg. 1. Then, we fine-tune the pruned model to enhance its quality further.

Single- *vs.* multi-depth search Since we apply the same target depth level for all time steps (single-depth search), one might argue that our approach overly restricts the search space, potentially missing out on additional performance gains that could be achieved by using different depth levels for different time steps (multi-depth search). Here, we demonstrate that our single-depth search can find a near-optimal solution in a highly efficient manner compared to the multi-depth search.

To prove this, we compare the performances of our single-depth search scheme and the multi-depth search scheme. For the multi-depth search, we find its optimal solutions using exhaustive search. To mitigate the search overhead in this analysis, we confine the depth-skip levels from 7 to 12, and use 10 denoising

iterations, and use the same depth levels for every two time steps, i.e., two consecutive time steps use the same depth level.

While the primary goal of depth-skip pruning is to reduce the model size, it also affects the latency and output quality. As a result, the multi-depth search may potentially find a solution that is more optimal in either quality or latency while having the same model size as the solution of the single depth-search approach. Similarly, it may also find a superior solution in terms of model size or latency while having the same quality. Thus, we analyze the performances in all the three aspects for a comprehensive analysis.

Optimizing quality or latency while fixing model size. As diffusion models use a single denoising UNet for all time steps, the pruned model size is determined by the deepest depth across all time steps. Having this in mind, to achieve the highest output quality while maintaining the same model size as the single-depth search, the solution of the multi-depth search must be the same as that of the single-depth search because maximizing the output quality necessitates utilizing as many network blocks as possible for all time steps. On the other hand, we may find a solution with a smaller latency and the same model size using the multi-depth search if we accept a certain amount of quality degradation. Nonetheless, the gain is small as shown in Tab. 1(a). For the quality degradation of 0.2 dB, the gain in latency obtained by the multi-depth search is at most 4.27%.

Optimizing model size or quality while fixing latency. Another possibility is to use the multi-depth search to find a solution with a smaller model size or higher quality and the same latency as the result of the single-depth search. However, finding a solution with a smaller model size while fixing the latency neither makes sensor nor is possible, as a smaller model inevitably leads to lower output quality and a smaller latency. We may find a solution with a higher quality while fixing the latency using the multi-depth search. However, in this case, the solution requires a significantly larger model size compared to that of our approach as shown in Tab. 1(b).

Optimizing model size or latency while fixing quality. We may use the multi-depth search to find a solution with a smaller model size and the same quality as the single-depth search. However, again, this is impossible because a smaller model size inevitably causes lower output quality. Finally, we may find a solution with a smaller latency and the same quality using the multi-depth search, but such a solution requires a substantially larger model size as shown in Tab. 1(c).

This analysis indicates that our single-depth search scheme identifies solutions that are as effective as those found by multi-depth search, but with a considerably reduced search overhead.

3.3 Time-step Optimization

Our time-step optimization scheme is inspired by the following observation: earlier time steps are primarily involved in the generation of overall image structures incorporating the text prompt [2], while later time steps are mainly responsible for synthesizing image details [11,12]. Based on this observation, we hypothesize

that later time steps are more influential in image restoration tasks like StableSR [69], whereas earlier time steps play a greater role in image editing like IP2P [6]. To validate this, we conduct IP2P [6] with only five steps at different intervals. In Fig. 1(b), the black dashed line represents IP2P [6] generation using a uniform time sequence, while the purple and blue dashed lines represent non-uniform sequences with a focus on early time steps. As depicted in the figure, prioritizing earlier time steps yields viable results using only five steps. Although the impact of early and later time steps depends on tasks, we empirically observe similar phenomena in other I2I tasks. Inspired by this observation, we design our time-step optimization method to find a biased sequence of time steps for each task, as illustrated in Fig. 2(b).

Specifically, the time-step optimization aims to find an optimal sequence of time steps for a given number of time steps. To design an effective and efficient parameterization for finding a biased sequence, we exploit the gamma curve formulation, which is defined as:

$$F_t(\gamma, n) = T \cdot t^{\gamma}, \quad \gamma > 0,$$
$$t = 0, \frac{1}{n-1}, \frac{2}{n-1}, \cdots, 1 \tag{2}$$

where T represents the last time step, γ is a parameter to control the shape of the gamma curve, and n is the number of iterations. If $\gamma > 1$, the generation process concentrates on the early time steps of generation, while if $0 < \gamma < 1$, it focuses on the later time steps. Then, our optimization problem becomes to find an optimal γ that produces the output closest to the original sampling results using a small n.

Algorithm 2: Time-step optimization

Input: step size η, metric function $\mathcal{M}$, signum function sgn, small value ϵ, GT iteration N, target iteration n,
Output: Optimal time-step $F_t(p^s_{prev}, n)$
$p \leftarrow 1$, $m \leftarrow \infty$, $x_T \sim \mathcal{N}(0, I)$
$x_{uni} \leftarrow \text{Sampler}_{DDIM}(x_T, c_I, \mathcal{P}, F_t(p, n))$
$x_{pos} \leftarrow \text{Sampler}_{DDIM}(x_T, c_I, \mathcal{P}, F_t(p+\epsilon, n))$
$x_{neg} \leftarrow$ $\text{Sampler}_{DDIM}(x_T, c_I, \mathcal{P}, F_t((p+\epsilon)^{-1}, n))$
$s \leftarrow \text{sgn}(\mathcal{M}(x_{uni}, x_{neg}) - \mathcal{M}(x_{uni}, x_{pos}))$
$x^* \leftarrow \text{Sampler}_{DDIM}(x_T, c_I, \mathcal{P}, F_t(p, N))$
repeat
  $m_{prev} \leftarrow m$, $p_{prev} \leftarrow p$
  $p \leftarrow p + \eta$
  $x \leftarrow \text{Sampler}_{DDIM}(x_T, c_I, \mathcal{P}, F_t(p^s, n))$
  $m \leftarrow \mathcal{M}(x^*, x)$
until $m > m_{prev}$;

While this simple strategy already provides comparable or outperforming results to previous state-of-the-art approaches [31,75], it is still limited due to the fixed nature of the first and last time steps. To further improve the performance, we introduce a scale-down mechanism for the gamma curve. Specifically, we scale down the gamma curve toward T when $\gamma < 1$ proportional to decrease of γ value, and vice versa. The formal definition is as follows:

$$F_t(\gamma, n) = T \cdot t'^{\gamma}, \quad t' = \frac{T \cdot t - t_l}{t_u - t_l}$$
$$(t_l, t_u) = \begin{cases} (0, T + \alpha(\gamma - 1)), & \gamma \geq 1 \\ (\alpha(1 - 1/\gamma), T), & \gamma < 1 \end{cases} \tag{3}$$

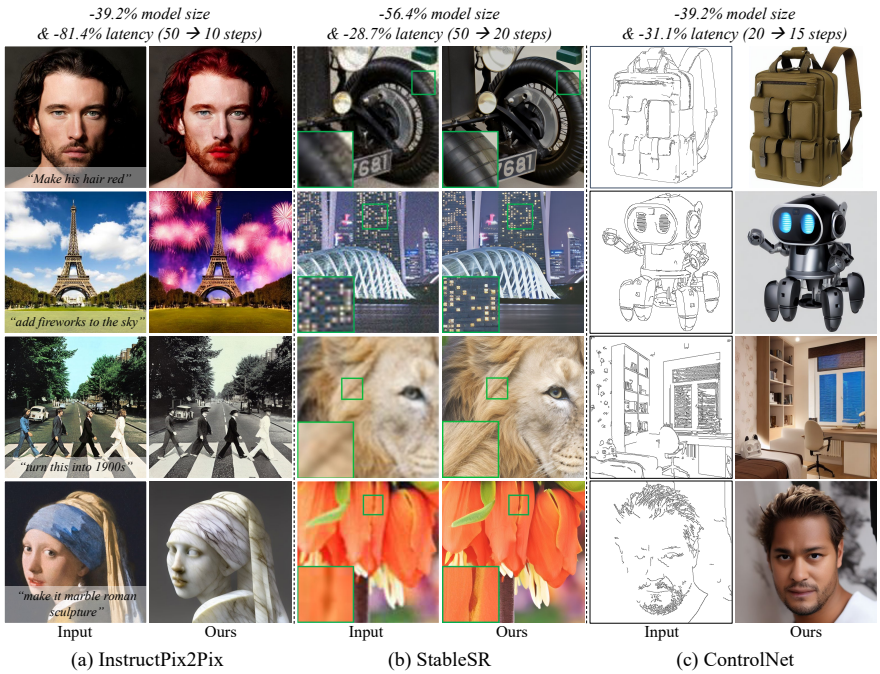


Fig. 3: Qualitative examples of our depth-skip pruning and time-step optimization on IP2P [6], StableSR [69], and ControlNet [78].

where α is a coefficient for scale strength. This adjustment allows for greater flexibility and potentially more effective optimization of the time steps.

The search process consists of two stages. Firstly, we determine whether γ increases or decreases by evaluating which direction yields better outputs. Then, we perform a greedy search by progressively increasing or decreasing the value of γ until no further improvement in quality is observed, as described in Alg. 2. Our time-step optimization is designed to be highly efficient in terms of computational cost by aggressively limiting the search space to one dimension. Despite this simplification, as demonstrated in our experiments (Sec. 4.3), it achieves higher-quality results than existing state-of-the-art methods like AutoDiffusion [31], while being at least 62 times faster.

4 Experiments

In this section, we validate the effectiveness of our compression method by applying it to IP2P [6] for image editing, StableSR [69] for image restoration, and ControlNet [78] for image-conditioned image generation. In the case of ControlNet, we use canny edge maps as input condition in our experiments. The baseline step numbers used are the same as originally specified in their respective applications: 50 steps for IP2P [6] and StableSR [69], and 20 steps for ControlNet [78]. When discussing the model size and latency, we only consider those of the dif-

Table 2: Latency and the number of parameters including VAE, text encoder, and adapter network. "U-Net+" indicates the union of the U-Net and additional adapter network. The compression includes both depth-skip pruning and time-step optimization. The unit of latency is seconds.

	Latency (U-Net+/Total/Iteration)			Parameter (U-Net+/Total)		
	Original	Compressed	Total Reduction	Original	Compressed	Total Reduction
IP2P [6]	6.31/6.54/50	1.17/1.40/10	78.6%	859M/1066M	522M/729M	31.6%
StableSR [69]	2.81/2.94/50	0.88/1.01/20	65.7%	969M/1176M	452M/658M	44.0%
ControlNet [78]	1.57/1.75/20	1.08/1.26/15	28.0%	1220M/1427M	883M/1090M	23.6%

fusion U-Net, excluding the text-encoder [49], auto-encoder [13], and additional adapter networks [78] unless specified otherwise. We refer the readers to Sec. S1 in the Supplemental Document for more details.

4.1 Qualitative and Quantitative Comparisons

Fig. 3 shows results of the models of different tasks compressed by the proposed depth-skip pruning and time-step optimization. Compared to the original models of IP2P [6], StableSR [69], and ControlNet [78], our compressed models use only 60.8%, 43.6%, and 60.8% of the parameters, respectively, and their latencies are reduced to 18.6%, 31.3%, and 68.9%, respectively. Tab. 2 reports a quantitative comparison of the latencies and model sizes of the original models and their compressed results. In this comparison, we also report the total sizes and latencies including those of the VAE, text encoder and adapter networks. As the results show, despite the much smaller model sizes and latencies, the compressed models successfully produce visually pleasing results, clearly indicating that our method effectively reduces both model size and latency while preserving the original generative power required for each task.

4.2 Evaluation of Depth-skip Pruning

We compare the proposed depth-skip pruning with state-of-the-art pruning methods tailored for diffusion models: Diff-pruning [14] and BK-SDM [27]. For fine-tuning, we followed the original training strategy except for ControlNet [78], as the LAION [60] dataset used for ControlNet is no longer publicly available. Instead, we used the COCO [34] dataset. The optimal depths searched by our depth-search algorithm are $D9$ (60.8% of parameters) in IP2P [6] and ControlNet [78], and $D8$ (43.6% of parameters) in StableSR [69], respectively. We refer to Sec. S2 in the Supplemental Document for experimental results involving extreme depth-skip pruning, such as $D6$ (15.6% of parameters) models.

Qualitative comparison. Fig. 4 shows a qualitative comparison of depth-skip pruning with and without fine-tuning, and the previous pruning methods. Fig. 4(a) shows the results of the original models for each task without any pruning. Compared to the original results, the previous methods often produce semantically incorrect results with artifacts despite their larger model sizes than ours.

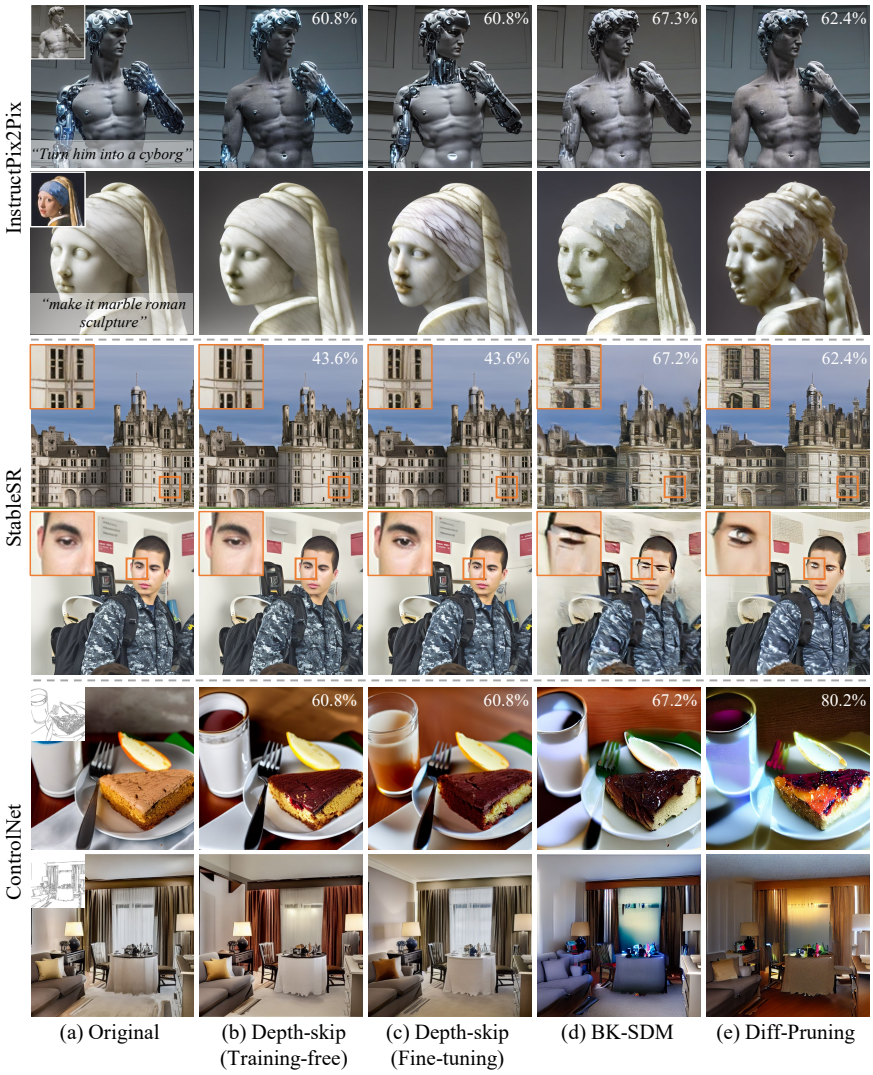


Fig. 4: Comparison of the depth-skip pruning and previous pruning methods. The number on the top-right side in each image denotes the pruned model size.

On the other hand, our depth-skip pruning shows more visually pleasing results compared to the previous methods, even with a smaller model size and without fine-tuning. Note that the results of our depth-skip pruning are not exactly the same as the original results due to the pruned layers and fine-tuning. Nevertheless, our methods produce semantically correct results, showing that the generative capabilities of the original models are well preserved.

Quantitative comparison. For quantitative comparison, we first compare the performances of the previous approaches and ours on StableSR [69]. To this end, we use the super-resolution validataion dataset of StableSR, which is generated

Table 3: Quantitative comparisons of the depth-skip pruning and other pruning methods for StableSR [69] and ControlNet [78].

Model	Steps	FID↓	PSNR↑	LPIPS↓	Parameter
Diff-pruning [14]	50	38.70	21.20	0.483	578M (62.4%)
BK-SDM [27]	50	64.46	21.45	0.492	615M (67.2%)
Depth-skip ($D9$) without fine-tuning	50	**28.55**	21.54	**0.441**	557M (60.8%)
Depth-skip ($D8$) without fine-tuning	50	40.25	21.40	0.467	400M (43.6%)
Depth-skip ($D8$)	50	30.15	21.48	0.449	400M (43.6%)
Depth-skip ($D8$) + Time-step optimization	20	32.31	**21.82**	0.457	400M (43.6%)
Original	50	27.70	21.51	0.437	917M (100%)

(a) StableSR [69]

Model	Steps	FID ↓	CLIP-Score↑	CLIP-a↑	Parameter
Diff-pruning [14]	20	28.52	28.91	5.01	687M (80.2%)
BK-SDM [27]	20	29.66	29.08	4.87	576M (67.2%)
Depth-skip ($D9$) without fine-tuning	20	21.18	30.29	**5.93**	521M (60.8%)
Depth-skip ($D9$)	20	**17.64**	**30.46**	5.88	521M (60.8%)
Depth-skip ($D9$) + Time-step optimization	15	18.65	30.37	5.87	521M (60.8%)
Original fine-tuned on the COCO dataset	20	17.52	30.57	5.91	857M (100%)
Original	20	19.88	30.42	6.10	857M (100%)

(b) ControlNet [78]

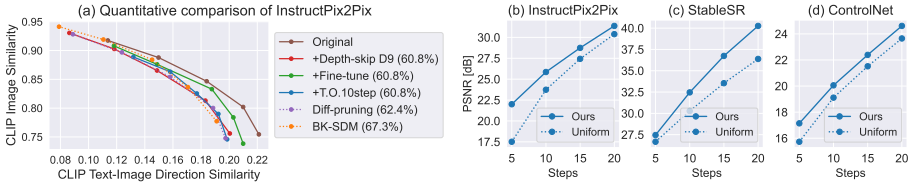


Fig. 5: (a) Quantitative comparison of the depth-skip pruning and the other methods on IP2P [6]. "T.O." denotes the time-step optimization. (b-d) Comparison between the time-step optimization and uniform sampling.

from the DIV2K dataset [1]. Tab. 3(a) shows the quantitative comparison. We measure FID [21], PSNR and LPIPS [81] scores for evaluation. As the table shows, $D9$ without fine-tuning and $D8$, both of which are our depth-skip pruning results, achieve the best FID [21], PSNR and LPIPS [81] scores close to the scores of the original model, significantly outperforming Diff-pruning [14] and BK-SDM [27] even though they have smaller model sizes and $D9$ does not use fine-tuning. The table also shows the effect of the fine-tuning step in the depth-skip pruning. By comparing $D8$ before and after fine-tuning, it is evident that fine-tuning significantly enhances performance, enabling $D8$ to match the original model's performance with less than half of the original model's size. Finally, although time-step optimization results in a minor quality degradation due to the reduction of iterations by more than half, it still surpasses the performance of the previous methods.

We also conduct a quantitative evaluation on ControlNet [78]. For evaluation, we use the COCO validation set [34] for ControlNet. For the input text prompts required for the ControlNet models, we generate text prompts using BLIP [30]. Tab. 3(b) shows the quantitative comparison. As mentioned earlier, we use the COCO dataset for fine-tuning instead of the LAION dataset [60]. Thus, we

also compare the result of the original model fine-tuned on the COCO dataset. Similar to the results in Tab. 3(b), despite their smaller model sizes, our results achieve the best scores in all quality metrics regardless of fine-tuning and time-step optimization.

Finally, we conduct a quantitative comparison on IP2P [6]. For evaluation, we follow the protocol of IP2P [6]. Specifically, we measure the CLIP image similarity scores [49] and CLIP text-image direction similarity scores [15]. As IP2P allows the control of the editing strength using the CFG [23] parameter, we plot the scores for different image CFG parameter values ranging from 1.0 to 1.8. Fig. 5(a) shows the quantitative comparison. The solid lines in this figure display the quantitative results where the depth-skip, fine-tuning and time-step optimization with 10 steps are successively applied, and the dotted lines show the results of previous pruning method. Our depth-skip pruning without fine-tuning shows comparable results to other pruning methods. After fine-tuning, which recovers the quality degradation of model pruning, our pruning method outperforms the other methods by a large margin.

4.3 Evaluation of Time-step Optimization

We evaluate the performance of the proposed time-step optimization. For evaluation of the proposed method, we apply time-step optimization to the original models without applying depth-skip pruning. Also, we randomly sample 100 images from the training dataset for the search process, and employ a bias coefficient of $\alpha = 30$.

Fig. 6 shows a qualitative comparison between the results of our time-step optimization and the uniform sampling strategy that samples evenly distributed time steps. For all the tasks, our approach produces superior results than the uniform sampling strategy. Specifically, in the case of IP2P [6] and ControlNet [78], our method produces results that are similar to the results of the original models even with only five steps. On the other hand, the outputs of the uniform sampling scheme quickly degrade as the number of time steps decreases. In the case of StableSR [69], our results show accurately restored high-frequency details, while the uniform sampling fails to restore such details.

Fig. 5(b-d) visualizes PSNR values of the time-step optimization and uniform sampling strategies with respect to different numbers of iterations. The PSNR values of the outputs from the optimized time steps are measured against the results of the original models with 50 iterations using DDIM [63] deterministic process. Also, we measure the metric based on a random selection of 1,000 images from the validation dataset for each task. Across all the iteration numbers, our time-step optimization consistently yields higher PSNRs for all the tasks.

Tab. 4 compares the output qualities and search times of our time-step optimization with those of previous state-of-the-art time scheduling approaches for diffusion models: AutoDiffusion [31] and Xue et al.'s method [75]. As Xue et al.'s method is based on a highly simplified mathematical approximation for efficient time scheduling, it takes only a few seconds for five time steps, and less than a minute for 10 time steps. Nevertheless, due to its approximation, its output

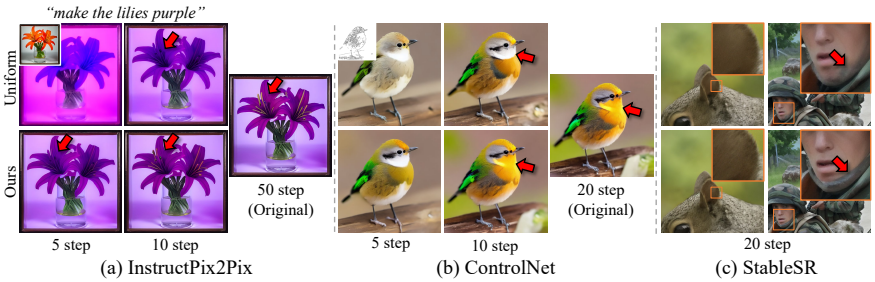


Fig. 6: Comparison between the time-step optimization and uniform sampling.

Table 4: Quantitative comparison of time-step optimization with previous methods.

		InstructPix2Pix [6]			StableSR [69]			ControlNet [78]		
	# Steps	Ours	AutoDiff.	Xue et al.	Ours	AutoDiff.	Xue et al.	Ours	AutoDiff.	Xue et al.
PSNR	5	**22.00**	20.64	14.35	**27.46**	26.58	25.83	**17.14**	16.83	13.89
(dB)	10	**25.86**	24.79	20.62	**32.46**	29.03	27.71	**20.05**	19.23	16.23
Search	5	38.7m	40.5h	3.2s	9.5m	16.3h	3.2s	14.1m	31.2h	3.2s
time	10	30.9m	75.1h	53.1s	11.1m	27.1h	53.1s	18.0m	65.1h	53.1s

quality is the lowest among the compared methods. On the contrary, AutoDiffusion takes tens of hours to search for optimal time steps due to its reliance on the genetic algorithm. Despite the lengthy search duration, it still lags behind our method in output quality, as the genetic algorithm tends to get trapped in local minima. In contrast, thanks to its constrained yet effective search space, our method only requires 10 to 40 minutes and consistently delivers superior output quality for all the cases. More analyses and details can be found in Sec. S3 and S4 in the Supplemental Document.

5 Conclusion

In this paper, we introduced a novel compression method for downstream I2I diffusion models, which consists of depth-skip pruning and time-step optimization for reducing the memory footprint and latency, respectively. Despite their simplicity, our experiments show that they significantly outperform previous state-of-the-art task-agnostic pruning and time scheduling approaches.

Limitation & Future work. Our depth-skip pruning assumes that the denoising network has a U-Net [54]-based architecture. Therefore, the pruning method would be unavailable to other diffusion models with different network architectures, such as transformers [47]. Developing a compression method applicable to various network architectures could be a promising future direction.

Acknowledgements. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH), No.2024-00457882, AI Research Hub Project). This work was also partly supported by Samsung Research Funding Center (SRFC-IT1801-52).

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (July 2017)
2. Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., et al.: ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. arXiv preprint arXiv:2211.01324 (2022)
3. Bansal, A., Chu, H.M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., Goldstein, T.: Universal guidance for diffusion models. In: CVPR. pp. 843–852 (2023)
4. Bao, F., Li, C., Zhu, J., Zhang, B.: Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. arXiv preprint arXiv:2201.06503 (2022)
5. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: Fusing diffusion paths for controlled image generation. International Conference on Machine Learning (2023)
6. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR. pp. 18392–18402 (2023)
7. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: European Conference on Computer Vision. pp. 17–33. Springer (2022)
8. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: Hinet: Half instance normalization network for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 182–192 (2021)
9. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. ICCV (2023)
10. Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., Wang, Z.: Chasing sparsity in vision transformers: An end-to-end exploration. Advances in Neural Information Processing Systems **34**, 19974–19988 (2021)
11. Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., Yoon, S.: Perception prioritized training of diffusion models. In: CVPR. pp. 11472–11481 (2022)
12. Deja, K., Kuzina, A., Trzcinski, T., Tomczak, J.: On analyzing generative and denoising capabilities of diffusion-based deep generative models **35**, 26218–26229 (2022)
13. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
14. Fang, G., Ma, X., Wang, X.: Structural pruning for diffusion models. Advances in neural information processing systems **36** (2024)
15. Gal, R., Patashnik, O., Maron, H., Bermano, A.H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. ACM Transactions on Graphics (TOG) **41**(4), 1–13 (2022)
16. Hachnochi, R., Zhao, M., Orzech, N., Gal, R., Mahdavi-Amiri, A., Cohen-Or, D., Bermano, A.H.: Cross-domain compositing with pretrained diffusion models. arXiv preprint arXiv:2302.10167 (2023)
17. Ham, C., Hays, J., Lu, J., Singh, K.K., Zhang, Z., Hinz, T.: Modulating pretrained diffusion models for multimodal image synthesis. SIGGRAPH Conference Proceedings (2023)
18. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4340–4349 (2019)

19. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE international conference on computer vision. pp. 1389–1397 (2017)
20. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score. In: ICCV. pp. 2328–2337 (2023)
21. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
22. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. vol. 33, pp. 6840–6851 (2020)
23. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
24. Jia, D., Han, K., Wang, Y., Tang, Y., Guo, J., Zhang, C., Tao, D.: Efficient vision transformers via fine-grained manifold distillation. arXiv e-prints pp. arXiv–2107 (2021)
25. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR. pp. 4401–4410 (2019)
26. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR. pp. 8110–8119 (2020)
27. Kim, B.K., Song, H.K., Castells, T., Choi, S.: Bk-sdm: Architecturally compressed stable diffusion for efficient text-to-image generation. ICML Workshop on Efficient Systems for Foundation Models (ES-FoMo) (2023), https://openreview.net/forum?id=bOVydU0XKC
28. Kim, G., Kang, K., Kim, S., Lee, H., Kim, S., Kim, J., Baek, S.H., Cho, S.: Bigcolor: colorization using a generative color prior for natural images. In: ECCV. pp. 350–366. Springer (2022)
29. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016)
30. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International conference on machine learning. pp. 12888–12900. PMLR (2022)
31. Li, L., Li, H., Zheng, X., Wu, J., Xiao, X., Wang, R., Zheng, M., Pan, X., Chao, F., Ji, R.: Autodiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In: ICCV. pp. 7105–7114 (2023)
32. Li, Y., Wang, H., Jin, Q., Hu, J., Chemerys, P., Fu, Y., Wang, Y., Tulyakov, S., Ren, J.: Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. Advances in Neural Information Processing Systems **36** (2024)
33. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: CVPR. pp. 300–309 (2023)
34. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
35. Lin, X., He, J., Chen, Z., Lyu, Z., Fei, B., Dai, B., Ouyang, W., Qiao, Y., Dong, C.: Diffbir: Towards blind image restoration with generative diffusion prior. arXiv preprint arXiv:2308.15070 (2023)
36. Liu, E., Ning, X., Lin, Z., Yang, H., Wang, Y.: Oms-dpm: Optimizing the model schedule for diffusion probabilistic models. arXiv preprint arXiv:2306.08860 (2023)
37. Liu, L., Ren, Y., Lin, Z., Zhao, Z.: Pseudo numerical methods for diffusion models on manifolds (2022)

38. Liu, X., Zhang, X., Ma, J., Peng, J., et al.: Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In: The Twelfth International Conference on Learning Representations (2023)
39. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE international conference on computer vision. pp. 2736–2744 (2017)
40. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps **35**, 5775–5787 (2022)
41. Lu, S., Liu, Y., Kong, A.W.K.: Tf-icon: Diffusion-based training-free cross-domain image composition. In: ICCV. pp. 2294–2305 (2023)
42. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
43. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: CVPR. pp. 14297–14306 (2023)
44. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
45. Pan, B., Panda, R., Feris, R.S., Oliva, A.J.: Interpretability-aware redundancy reduction for vision transformers (Jun 22 2023), uS Patent App. 17/559,053
46. Pan, Z., Zhuang, B., Huang, D.A., Nie, W., Yu, Z., Xiao, C., Cai, J., Anandkumar, A.: T-stitch: Accelerating sampling in pre-trained diffusion models with trajectory stitching. arXiv preprint arXiv:2402.14167 (2024)
47. Peebles, W., Xie, S.: Scalable diffusion models with transformers. arXiv preprint arXiv:2212.09748 (2022)
48. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. ICLR (2023)
49. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
50. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning. pp. 8821–8831. PMLR (2021)
51. Ravi, H., Kelkar, S., Harikumar, M., Kale, A.: Preditor: Text guided image editing with diffusion prior. arXiv preprint arXiv:2302.07979 (2023)
52. Richardson, E., Goldberg, K., Alaluf, Y., Cohen-Or, D.: Conceptlab: Creative generation using diffusion prior constraints. arXiv preprint arXiv:2308.02669 (2023)
53. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
54. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR **abs/1505.04597** (2015), `http://arxiv.org/abs/1505.04597`
55. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: CVPR. pp. 22500–22510 (2023)
56. 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.: Photorealistic text-to-image diffusion models with deep language understanding **35**, 36479–36494 (2022)
57. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: ICLR (2022)

58. Sarukkai, V., Li, L., Ma, A., Ré, C., Fatahalian, K.: Collage diffusion. arXiv preprint arXiv:2303.00262 (2023)
59. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. arXiv preprint arXiv:2311.17042 (2023)
60. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022)
61. Shih, A., Belkhale, S., Ermon, S., Sadigh, D., Anari, N.: Parallel sampling of diffusion models. Advances in Neural Information Processing Systems **36** (2024)
62. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
63. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2021)
64. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. ICLR (2021)
65. Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., Tao, D.: Patch slimming for efficient vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12165–12174 (2022)
66. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
67. Tumanyan, N., Geyer, M., Bagon, S., Dekel, T.: Plug-and-play diffusion features for text-driven image-to-image translation. In: CVPR. pp. 1921–1930 (2023)
68. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In: CVPR. pp. 12619–12629 (2023)
69. Wang, J., Yue, Z., Zhou, S., Chan, K.C., Loy, C.C.: Exploiting diffusion prior for real-world image super-resolution. arXiv preprint arXiv:2305.07015 (2023)
70. Wang, Q., Zhang, B., Birsak, M., Wonka, P.: Mdp: A generalized framework for text-guided image editing by manipulating the diffusion path. arXiv preprint arXiv:2303.16765 (2023)
71. Wang, X., Li, Y., Zhang, H., Shan, Y.: Towards real-world blind face restoration with generative facial prior. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9168–9178 (2021)
72. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17683–17693 (2022)
73. Watson, D., Chan, W., Ho, J., Norouzi, M.: Learning fast samplers for diffusion models by differentiating through sample quality. In: ICLR (2021)
74. Xie, J., Li, Y., Huang, Y., Liu, H., Zhang, W., Zheng, Y., Shou, M.Z.: Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In: ICCV. pp. 7452–7461 (2023)
75. Xue, S., Liu, Z., Chen, F., Zhang, S., Hu, T., Xie, E., Li, Z.: Accelerating diffusion sampling with optimized time steps. arXiv preprint arXiv:2402.17376 (2024)
76. Yang, B., Luo, Y., Chen, Z., Wang, G., Liang, X., Lin, L.: Law-diffusion: Complex scene generation by diffusion with layouts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22669–22679 (2023)
77. Yu, J., Wang, Y., Zhao, C., Ghanem, B., Zhang, J.: Freedom: Training-free energy-guided conditional diffusion model. ICCV (2023)

78. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: CVPR. pp. 3836–3847 (2023)
79. Zhang, Q., Chen, Y.: Fast sampling of diffusion models with exponential integrator. arXiv preprint arXiv:2204.13902 (2022)
80. Zhang, Q., Song, J., Huang, X., Chen, Y., Liu, M.Y.: Diffcollage: Parallel generation of large content with diffusion models. CVPR (2023)
81. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
82. Zheng, H., Nie, W., Vahdat, A., Azizzadenesheli, K., Anandkumar, A.: Fast sampling of diffusion models via operator learning. In: International Conference on Machine Learning. pp. 42390–42402. PMLR (2023)
83. Zhu, M., Tang, Y., Han, K.: Vision transformer pruning. arXiv preprint arXiv:2104.08500 (2021)