Diffusion Model Compression for Image-to-Image Translation –Supplemental Document–

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

In this Supplementary Material, we present:

- Implementation details (Sec. S1)
- Discussions on simple editing with shallower depth on IP2P (Sec. S2)
- Additional analyses on the time-step optimization (Sec. S3)
- Additional comparisons with AutoDiffusion (Sec. S4)
- Additional details and results on applications (Sec. S5)
- Discussions on the multi-depth search of IP2P (Sec. S6)
- Profiling on the parameter number and latency for Stable Diffusion (Sec. S7)
- Discussions on a challenge of step distillation in image restoration (Sec. S8)
- Additional latency and computational cost analysis (Sec. S9)
- Time-step optimization using different scheduler (Sec. S10)
- Additional qualitative results of StableSR (Fig. S7)

S1 Implementation Details

For the depth-search, we randomly sample 100 images from the training dataset of each task. In StableSR [23], we set the PSNR threshold at 28 dB, using the ground truth based on 50 iteration results. Regarding IP2P [2], we utilized a combination of CLIP image similarity [17] and directional CLIP similarity [4], which are the same evaluation protocol used in IP2P [2]. The threshold values for these metrics were set at 0.7 and 0.2, respectively. Regarding ControlNet [24], we set the FID [5] threshold at 22.

S2 Simple Editing with Shallower Depth on IP2P

IP2P [2] supports a wide range of image editing operations with various levels of difficulty. This implies that different input text commands to IP2P may require different depth levels, i.e., easy commands that do not alter the image structure may require shallower depths than other commands. Fig. S1 shows examples of such easy and difficult commands. In the figure, the commands "Make it autumn" and "Make it cubism" are easy ones that do not alter image structures, while the other commands shown on the right are difficult ones that change image structures. As shown in the figure, the shallow-depth model with only six depth levels

Fig. S1: Various structure-preserved editing operates even after depth-skip compression at level 6, which uses 15.6% parameters. The shallower the depth-skip, the more complex edits, such as structure-modified editing, tend to fail.

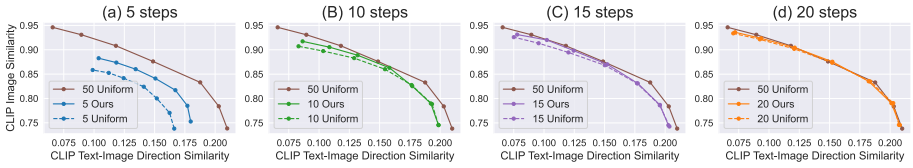

Fig. S2: Quantitative comparison on time-step optimization for InstructPix2Pix [2].

(D6 in the figure), which uses only 15.6% parameters, successfully produces plausible results for the easy commands, but fails for the difficult ones, for which, the deeper-depth model with nine depth levels (D9) succeeds. This result suggests that, by limiting target image editing operations to simple structure-preserving ones, more effective depth-skip compression can be achieved.

S3 Additional Analyses on Time-step Optimization

Fig. S2 presents additional experimental results on our time-step optimization for various iteration numbers, conducted on IP2P [2]. Our time-step optimization achieves superior results compared to uniform sampling in terms of both CLIP image similarity and CLIP text-image direction similarity for small iteration numbers, as demonstrated in Fig. S2 (a). The differences among the uniform sequence, our optimized sequence and 50-step sampling become marginal at 20 steps, as shown in Fig. S2 (d). This suggests that by transferring to simpler downstream tasks, the necessary number of iterations is naturally reduced compared to the conventional 50-step approach.

Table S1: The optimized values for the time-step control parameter γ varies with change of CFG [6] values.

Iteration	CFG_I			
	1.0	1.2	1.4	1.6
5	0.322	0.333	0.377	0.435
10	0.556	0.581	0.617	0.704
15	0.709	0.714	0.833	0.833

Table S2: Time-step optimization results with and without depth-skip compression. The optimized values for the time-step control parameter γ are almost identical between the models with and without depth-skip compression, indicating that the time-step optimization is not affected by the depth-skip compression.

	5 step		10 step		15 step	
Depth-skip	-	✓	-	✓	-	✓
InstructPix2Pix [2]	0.322	0.322	0.556	0.556	0.709	0.714
StableSR [23]	1.900	1.900	1.600	1.600	1.610	1.610

Impact of CFG Classifier Free Guidance (CFG) [6] is a method that is widely used in numerous conditional diffusion models to control the influence of conditions on image generation. CFG provides a single strength parameter for each condition to control its influence. To obtain a high-quality result that a user wants, they often run diffusion models multiple times with different values for the CFG strength parameters. IP2P [2], which is one of our target applications, also utilizes CFG to control the strength of prompt or image conditions.

In this section, we additionally present an analysis on the impact of CFG on the time-step optimization. Specifically, we conduct the time-step optimization of IP2P [2] with different CFG strength parameter values for the image condition, and with different iteration numbers. Tab. S1 shows the result where CFG_I indicates the CFG strength parameter for the image condition. As the result reveals, changing the CFG parameter leads to significantly different values for the time-step control parameter γ, which indicate different optimal time step sequences, for all iteration numbers.

This result also validates the practicality of our computationally-efficient time-step optimization method. To achieve high-quality results for different CFG parameter values requires to perform time-step optimization multiple times, which can be excessively time-consuming given the wide range of CFG parameter values. Nevertheless, our time-step optimization can significantly reduce the computational overhead compared to AutoDiffusion [8], which is the previous state-of-the-art method, as our method performs at least 62 times faster than AutoDiffusion, as shown in Table 4 in the main paper.

Impact of Depth-skip on Time-step optimization We analyze the impact of depth-skip on the time-step optimization. To this end, we prepare two diffusion models: one is an original full model, and the other is a depth-skipped version obtained using our depth-skip compression. We then perform time-step

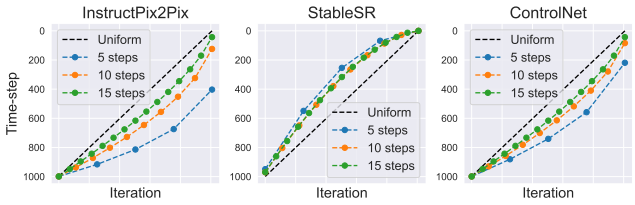

Fig. S3: Visualization for our optimized time-steps

Table S3: Quantitative comparison with AutoDiffusion for ControlNet [24].

Step	Method	FID↓	CLIP-Score↑	CLIP-a ↑
5	AutoDiffusion [8]	28.77	30.35	5.81
5	Ours	**28.26**	**30.38**	**5.85**
10	AutoDiffusion [8]	24.21	30.39	5.98
10	Ours	**23.29**	**30.40**	**6.00**

optimization on both models, and compare the results. Tab. S2 shows the time-step optimization results, where the optimal values for the time-step control parameter γ are not affected by depth skip. This result indicates that the time-step optimization and depth-skip compression are independent to each other, allowing them to be performed parallel.

Optimized time-step visualization Fig. S3 visualizes the results of our optimized time-steps. Consistent with our intuition, the results show that later time steps are crucial in StableSR, whereas early time steps are more important in other tasks.

S4 Comparisons on Time-step Optimization with AutoDiffusion

In this section, we present additional quantitative comparisons on time-step optimization with AutoDiffusion [8] using IP2P [2], StableSR [23] and ControlNet [24]. Regarding IP2P, we set the CFG strength parameters for the image and text conditions to 1.0 and 7.5, respectively, and optimize its time steps using our time-step optimization approach and AutoDiffusion for five iterations. We then evaluate the time-step optimization results using the CLIP image similarity [17] and directional CLIP similarity [4]. Tab. S4 shows a comparison between our result and the result of AutoDiffusion. As the table shows, our method achieves higher CLIP image similarity and directional CLIP similarity scores within a much shorter search time, proving the effectiveness of our approach.

For comparison on StableSR [23], we optimize the time steps of StableSR for 10 iterations using our time-step optimization approach and AutoDiffusion [8]. We then evaluate their results using the DIV2K validation dataset [1]. Tab. S5 shows that our approach achieves comparable results in terms of FID [5], PSNR, and LPIPS [25] within a significantly shorter search time.

Table S4: Comparison with AutoDiffusion for InstructPix2Pix [2] at 5 steps.

Method	$CLIP_I$	$CLIP_D$	Search Time
Ours	**0.767**	**0.190**	**38.7 min**
AutoDiffusion	0.765	0.188	40.5 hour

Table S5: Comparison with AutoDiffusion for StableSR [23] at 10 steps.

Method	FID	PSNR	LPIPS	Search Time
Ours	**34.956**	22.250	**0.461**	**11.1 min**
AutoDiffusion	36.496	**22.347**	0.466	27.1 hour

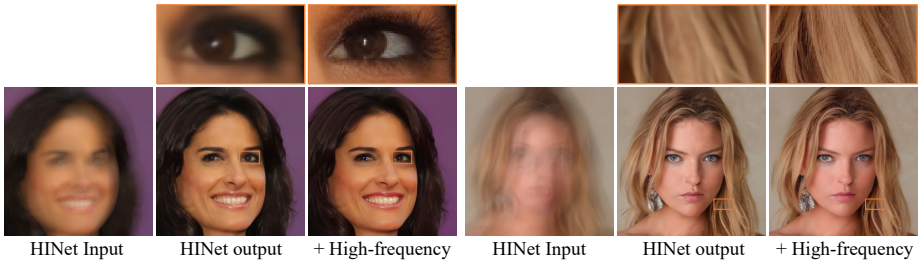

Fig. S4: Qualitative results on high-frequency synthesis application.

Regarding ControlNet [23], we conduct comparisons for 5 and 10 iterations. We then evaluate the results using the 5K COCO [10] validation dataset. Tab. S3 shows that our approach outperforms for all criteria.

S5 Additional Applications

To train the high-frequency synthesis model in Sec. 4.5 in the main paper, we use paired images consisting of a restored image and its ground-truth. To collect the restored images, we employ a pretrained HINet [3], trained for deblurring using the CelebA dataset [13]. To accommodate the additional input image, we implement minor modifications to the input network layer, same as IP2P [2]. To train the image inpainting model, we synthesize paired images where each image pair consists of a masked-image and its ground-truth image using the OpenImage dataset [7]. The text embedding is fixed as an empty text. Fig. S4 and Fig. S5 shows additional qualitative examples of the high-frequency synthesis and image inpainting models, respectively.

S6 Analysis on Multi-depth Search on IP2P

We demonstrated the efficiency of our single-depth search over multi-depth search on StableSR [23] in Sec. 3.3 in the main paper. This section presents a similar analysis on IP2P [2], which is detailed in Tab. S6. we use the CLIP image similarity [17] and directional CLIP similarity [4] for quality assessment. When the model size is constrained, the multi-depth search solution achieves marginal latency improvement over ours, while maintaining comparable quality, as shown in Tab. S6 (a). When the latency is constrained, optimizing the

Fig. S5: Qualitative results on image inpainting application.

Table S6: Comparisons between single and multi-depth search applied to InstructPix2Pix [2]. The percentages at "Time" and "Param", which indicates parameter, are proportion to requirements in full model. This results suggest that our single-depth approach efficiently searches the near-optimal point compared to a multi-depth search strategy.

	(a) Fix memory & Min time (ΔCLIP < 0.005)	(b) Fix time & Max CLIP$_I$ (ΔTime $< 1\%$)		(c) Fix time & Max CLIP$_D$ (ΔTime $< 1\%$)		(d) Fix quality & Min time (ΔCLIP < 0.005)	
Depth	ΔTime(%)	ΔCLIP$_I$	ΔParam(%)	ΔCLIP$_D$	ΔParam(%)	ΔTime(%)	ΔParam(%)
10	-7.46	+0.04	+18.73	+0.01	+18.73	-7.46	+0.00
9	-2.82	+0.03	+27.90	+0.01	+18.54	-6.30	+18.54
8	-6.64	+0.04	+36.01	+0.07	+26.64	-10.46	+17.47

quality using the multi-depth search leads to a significant increase in the parameter number, as shown in Tab. S6 (b) and (c). When the quality is constrained, minimizing the latency also results in a trade-off involving an increase in the parameter number, as shown in Tab. S6 (d). All these results suggest that our single-depth search is already able to achieve near-optimal solutions despite its considerably smaller search space.

S7 Profiling of Stable Diffusion

Tab. S7 shows the proportions of the parameter number and latency of each depth level. It is important to note that half of the deeper depth levels occupy 73.4% of the parameters. This is because the layers at coarser levels tend to have more channels, which contributes to an increase in the number of parameters. This tendency enhances the effectiveness of the depth-skip compression method in reducing the number of parameters, as our approach begins by discarding the coarsest layers first.

Table S7: Parameter number and latency for each depth level to Stable Diffusion (version 1). Half of the deeper depth levels occupy 73.4% of the parameters.

Depth	Parameter	Sum	Latency	Sum
D01	0.69%	0.70%	5.09%	5.09%
D02	1.25%	1.95%	10.65%	15.74%
D03	1.37%	3.32%	10.77%	26.51%
D04	2.85%	6.17%	8.57%	35.08%
D05	4.39%	10.56%	13.03%	48.10%
D06	5.06%	15.62%	13.06%	61.16%
D07	10.98%	26.60%	5.06%	66.22%
D08	16.71%	43.31%	10.49%	76.70%
D09	17.47%	60.78%	9.22%	85.93%
D10	9.17%	69.95%	1.88%	87.80%
D11	9.37%	79.32%	2.76%	90.56%
D12	9.37%	88.68%	2.70%	93.26%
D13	11.32%	100.0%	6.74%	100.00%

S8 Challenge of Step Distillation in Image Restoration

Step distillation [9, 12, 15, 16, 19, 20] is an acceleration technique capable of more aggressively speeding up diffusion models compared to our method, potentially reducing the diffusion process to only one or two iterations. Thus, one may want to adopt step distillation to a downstream task such as image restoration. However, unlike our training-free acceleration method, step distillation techniques are not suited for image restoration tasks. In this section, we analyze the the reason behind such incompatibility of step distillation, contrasting it with the effectiveness of our time-step optimization.

Step distillation methods alter the objective of diffusion models from functioning as progressive denoisers to simply approximating the posterior mean. This transformation compels diffusion models to adhere to the deterministic process for sampling. However, this deterministic sampling in image restoration results in artifact-prone outcomes. Fig. S6 exemplifies this issue, where (a-c) and (e-g) illustrate restoration results across various models, such as StableSR [23], DiffBIR [11], and LDM [18], with 100 steps using the stochastic and deterministic processes of DDIM [21], respectively. While the stochastic sampling consistently yields higher-quality outputs, the deterministic process frequently results in artifacts, including unnatural textures and noisy high-frequency details. This issue is inevitably reproduced in step-distilled models, as shown in Fig. S6 (h), since they should use the deterministic sampling process. In contrast, our time-step optimization adopts a training-free acceleration approach, ensuring broad compatibility with downstream applications by preserving the original models and their sampling processes, as shown in Fig. S6 (d).

A main reason of the failure of the deterministic sampling may be attributed to the local optimum problem. Specifically, the output of diffusion models represents the gradient of the data distribution at a time step, denoted as $\nabla_x \log p_t(x)$ [22]. Stochastic sampling employs this gradient in an iterative process that subtracts the gradient while simultaneously introducing random noise. This process can

Fig. S6: Super-resolution outcomes using various configurations. Top and bottom row are outcomes using stochastic and deterministic sampling, respectively. (d) is a sampling result using our time-step optimization. (h) is a sampling result after applying a step distillation method [19].

Table S8: Latency and MACs [26] improvement after depth-skip. We measure the latencies and computational costs with 50 iterations for IP2P [2] and StableSR [23], and 20 iterations for ControlNet [24].

	InstructPix2Pix [2]		StableSR [23]		ControlNet [24]	
	Time(s)	MACs(T)	Time(s)	MACs(T)	Time(s)	MACs(T)
Original	6.312	50.81	2.807	18.75	1.572	18.21
D9	5.978	47.27	2.421	17.49	1.444	17.26
D8	5.685	41.75	2.204	15.57	1.372	15.79

be interpreted as gradient descent with simulated annealing, facilitating a more comprehensive exploration of the solution space. Meanwhile, the deterministic process removes the stochastic component, functioning akin to the pure gradient decent method. This modification suggests that without stochastic perturbations, there is an elevated risk becoming trapped in local optima with unnatural textures.

S9 Latency & Computational Cost Analysis

Tab. S8 reports the reduction of latencies and computational costs achieved by applying depth-skip pruning. The table shows that depth-skip pruning reduces the latency and computation of 5.3% and 7.0% for IP2P [2], 21.5% and 17.0% for StableSR [23], and 8.2% and 5.2% for ControlNet [24], respectively.

Table S9: Quantitative comparisons of time-step optimization using multistep DPM-Solver++ [14] schedular. The evaluation protocol is same as the one used in Tab. 3 in the main paper.

		InstructPix2Pix		StableSR	
	# Steps	Ours	AutoDiffusion	Ours	AutoDiffusion
PSNR	5	**20.54**	18.73	**27.41**	26.79
(dB)	10	**26.82**	24.33	**32.09**	30.58

S10 Time-step optimization using different scheduler

Although we evaluated our time-step optimization using DDIM sampler in the main paper, our method can be applied to other schedulers as well, since the optimization process imposes no constraints, such as differentiability. Also, it is still effective in other schedulers, such as multistep DPM-Solver++ [14], as demonstrated in S9.

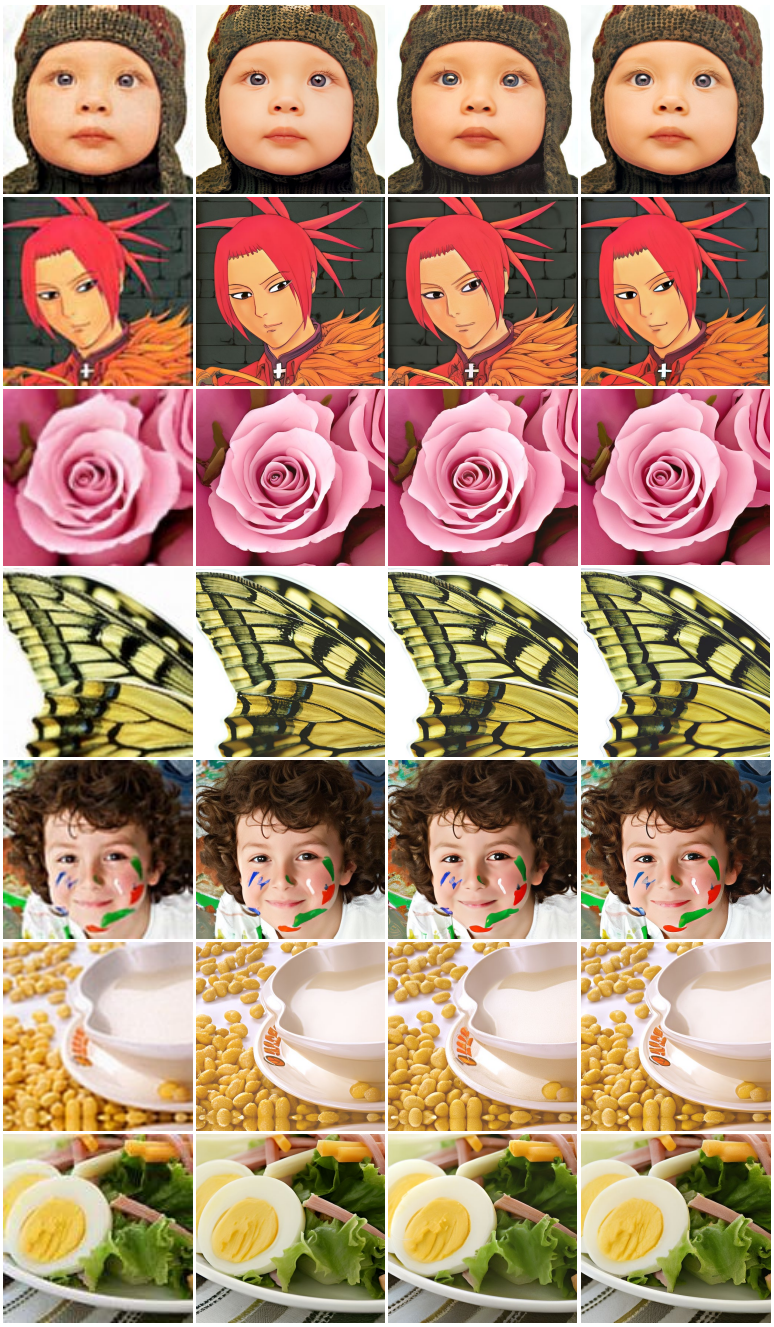

Fig. S7: Additional qualitative results from StableSR [23]. (c) shows the outputs after applying depth-skip compression at $D8$ with 50 steps. (d) displays the outcomes that combine depth-skip and time-step optimization.

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. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR. pp. 18392–18402 (2023)
3. 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)
4. 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)
5. 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)
6. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
7. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. IJCV (2020)
8. 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)
9. 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)
10. 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)
11. 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)
12. 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)
13. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)
14. 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)
15. 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)
16. 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)
17. 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)
18. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
19. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: ICLR (2022)
20. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. arXiv preprint arXiv:2311.17042 (2023)
21. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models (2021)
22. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. ICLR (2021)
23. 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)
24. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: CVPR. pp. 3836–3847 (2023)
25. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
26. Zhu, L.: Thop. `https://github.com/Lyken17/pytorc-OpCounter` (2019)