Supplementary Material for Denoising Diffusion Step-aware Models

Shuai Yang [1,3] **Yukang Chen** [2] **Luozhou Wang** [1,3]
Shu Liu [4*] **Yingcong Chen** [1,5*]

[1]HKUST(GZ) [2]CUHK [3]HKUST(GZ) - SmartMore Joint Lab
[4]SmartMore [5]HKUST

Appendix

A Experiment on LSUN-bedroom and AFHQ

We've validated DDSM's effectiveness and scalability on two extra datasets, LSUN-bedroom (64x64) and AFHQ (64x64). LSUN-bedroom is entirely non-object-centric. While AFHQ is object-centric but from a domain different from CelebA. SaDiffusion achieves 59% and 71% acceleration on the respective datasets without compromising image quality, as detailed in Table A.

The search results in Figure A of LSUN-bedroom and AFHQ further validate our analysis of the dataset's attributes. The LSUN-bedroom results parallels those of CIFAR-10, due to their shared high variety, while AFHQ echoed the patterns seen in CelebA-HQ, as both centrally feature face contours.

Table A: Evaluation on LSUN-bedroom and AFHQ

	LSUN-bedroom		AFHQ	
Method	FID	GFLOPs	FID	GFLOPs
ADM	**5.027**	49.88	7.244	49.88
DDSM	5.289	**20.48**	**6.249**	**14.72**

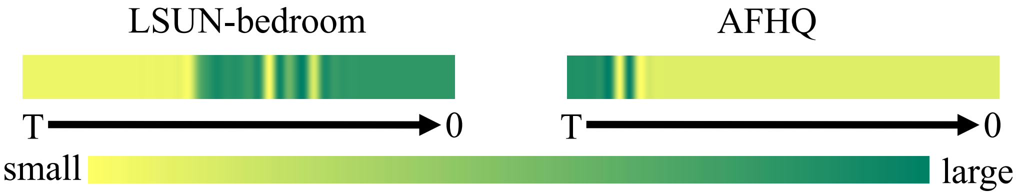


Figure A: search results on LSUN-bedroom and AFHQ

B Comparison with Concurrent Work Diff-Pruning

We elucidate the differences and provide an empirical comparison with the recent Diff-pruning Fang et al. (2023a;b). Diff-pruning employs a Taylor expansion over pruned timesteps to effectively reduce computational overhead. This approach primarily focuses on pruning the entire network. Our method, in contrast, adopts a strategy of adaptively pruning the network at various timesteps. This approach allows for a more nuanced and efficient solution. For a clearer understanding of these differences, we conducted an empirical comparison with Diff-pruning. Table B showcases our experiment on CelebA 64x64 using 100 steps. Our DDSM not only achieves a FID comparable to Diff-pruning but also demonstrates superior efficiency.

[*] Corresponding Author

Table B: Experimental comparison with Diff-Pruning, on CelebA 64x64, 100 steps.

Metric	Baseline	Diff-pruning	DDSM
FID (Lower is better)	6.48	6.24	**6.04**
GFLOPS (Lower is better)	49.9	26.6	**19.6**

C Combining with the EDM sampler

Our DDSM is compatible with recent sampling schedulers, such as DPM Lu et al. (2022a), DPM++ Lu et al. (2022b), uniPC Zhao et al. (2023), and EDM Karras et al. (2022). These samplers are primarily focused on devising novel noise schedules to enhance the performance of diffusion models. Our method, which adaptively prunes at different steps, is theoretically compatible with these approaches. Among these samplers, we have chosen EDM for our experiments, as we believe this selection could demonstrate our method's suitability for similar methods. In the experiment, we kept the pretrained weights of DDSM. We then replaced the original DDPM scheduler with EDM's Heun Discrete 2nd order method and initiated a new search process. The results, as illustrated in Table C, show that DDSM effectively integrates with EDM on the unconditional CIFAR-10 dataset, achieving 45% and 56% total TFLOPs saving for 50 steps and 100 steps. Note that, the quality can be further improved when retraining the network with EDM's setting.

Table C: Experiment result of combining DDSM with EDM, on CIFAR-10.

Metric	EDM50	EDM50+DDSM	EDM100	EDM100+DDSM
FID	3.65	3.61	3.41	3.49
Total TFLOPs	0.61	0.34	1.22	0.54

D Experiment on CelebA-HQ 128x128

We conduct experiments on CelebA-HQ-128x128 to prove that our DDSM is applicable with higher resolution data. In this experiment, we train a new slimmable supernet from scratch, and directly employ the search result on CelebA-HQ-64x64 to super network. Table D shows the result.

Table D: Experiment result on CelebA-HQ 128x128.

Metric	ADM	DDSM
FID (Lower is better)	7.53	7.71
GFLOPS (Lower is better)	194.00	76.18

E Experiment on Conditional CIFAR-10

We also conduct experiments on conditional CIFAR-10 to prove that our DDSM is applicable with guided diffusion. In this experiment, we train a guided diffusion slimmable supernet from scratch, and directly employ the search result on unconditional CIFAR-10 to super network. Table E shows the result.

F Implementation Details

Architecture Our implementation of slimmable networks draws inspiration from the US-Net. These networks are characterized by their ability to operate at various widths, providing a flexible and universal solution for network scalability. During the training of slimmable networks, we focus on optimizing the smallest, largest, and a selection of randomly sampled middle-sized subnetworks. This approach implicitly enhances the performance of all potential networks within the supernet.

In practical terms, our slimmable UNet adheres to the structure of ADM, but with a significant modification: all convolution layers are replaced by slimmable convolutions. These specialized

Table E: Experiment result on conditional CIFAR-10.

Metric	ADM	DDSM
FID (Lower is better)	2.48	2.52
GFLOPS (Lower is better)	12.14	6.20

convolutions are capable of adaptively processing tensors with a varying number of input channels. To accommodate a broader range of sub-networks, we adjusted the group number in the group normalization layer from 32 to 16. Additionally, we chose to omit the Batch Normalization (BN) calibration stage, as proposed in [11], since our diffusion UNet exclusively utilizes GroupNorm.

Regarding the sizes of the sub-networks, we offer seven different options, corresponding to $\frac{2}{8}$, $\frac{3}{8}$, $\frac{4}{8}$, $\frac{5}{8}$, $\frac{6}{8}$, $\frac{7}{8}$, and $\frac{8}{8}$ of the original ADM's width. To find more efficient strategies, we manually exclude some large width options. For example, in CIFAR-10, we exclude the $\frac{7}{8}$ and $\frac{8}{8}$ options. In most cases, the total strategy space is of $7^{\text{num\_timesteps}}$.

Search and Evaluation In the searching phase, the FID was computed by comparing the entire training dataset with the generated images. Empirical findings indicate that the FID score offers more consistent performance measurement compared to the loss; therefore, we aimed to search for a strategy yielding a lower FID. For the search parameters, the process encompasses a total of 10 iterations, with each iteration involving a population of 50, and maintaining a mutation rate of 0.001. The initial generation crucially includes a mix of uniform non-step-aware strategies and some random strategies. This specific approach to initialization and mutation has been empirically found to facilitate easier convergence of the search algorithm. Furthermore, a weight parameter is incorporated, which multiplies the GFLOPs to strike a balance between image quality and computational efficiency. For CIFAR-10, we set the weight parameter to 0.1 to favor higher image quality, while for CelebA, the FLOPs weight parameter is adjusted to 0.25. These parameters were manually selected to ensure that there are no compromises in generation quality.

Compatibility Experiments For the DDIM experiment, we adopted the weights from the 1000-step ADM model and applied the DDIM sampling schedule. As indicated in Table 4, our method significantly boosts DDIM's speed by 48%, 45%, 53%, and 62% for 10, 50, 100, and 1000 steps, respectively. This suggests that while additional steps enhance generation quality, they also introduce excess computational load. Our approach proves increasingly beneficial as the number of steps increases. In our latent diffusion experiment, we employed the AutoEncoderKL of SD1.4 to transform images into latent vectors, upon which our DDSM was trained. To adapt to the reduced spatial size, we modified the U-Net downsampling from 4 to 2. Our DDSM achieved a 60% acceleration in latent diffusion methods.

G Discussion on the extra cost

In this section, we discuss the extra cost introduced by our DDSM in the training and searching stage. Although we introduce extra training and search cost, DDSM still shows its advantages in inference. In deep learning algorithms, a common manner is we train a model once and repeatedly use this model to infer results. Nowadays, well-known Diffusion models all conform to this manner, like StableDiffusion and Imagen. They require a large amount of time to train. But once they are trained, they could repeatedly generate billions of images. Therefore, in terms of efficiency, we usually care more about the inference speed due to its repetitiveness.

Our method aims to reduce the inference cost of diffusion models. We prune the diffusion inference process with its step-aware strategy, accomplishing at most 76% acceleration. These cost savings could be counted repeatedly. The more samples we produce, the more benefits our framework demonstrates. In DDSM, we indeed introduce extra training and search cost, but these costs only take once. Concretely, the training cost of the slimmable network is about 2 times to 3 times of the normal diffusion model's training. The search cost depends on the search hyper-parameters. In our setting, it is approximately the same time as training a normal diffusion model. In Table F, we present the time cost of our DDSM on NVIDIA RTX 3090's GPU hours.

Table F: Total GPU hours of our DDSM.

	CIFAR-10	CelebA
DDPM-train	278	435
DDSM-train	502	1036
DDSM-search	320	524

H Separated Line Graphs of Search Results

We plot the search result separately for better visualization, as shown in Figure B

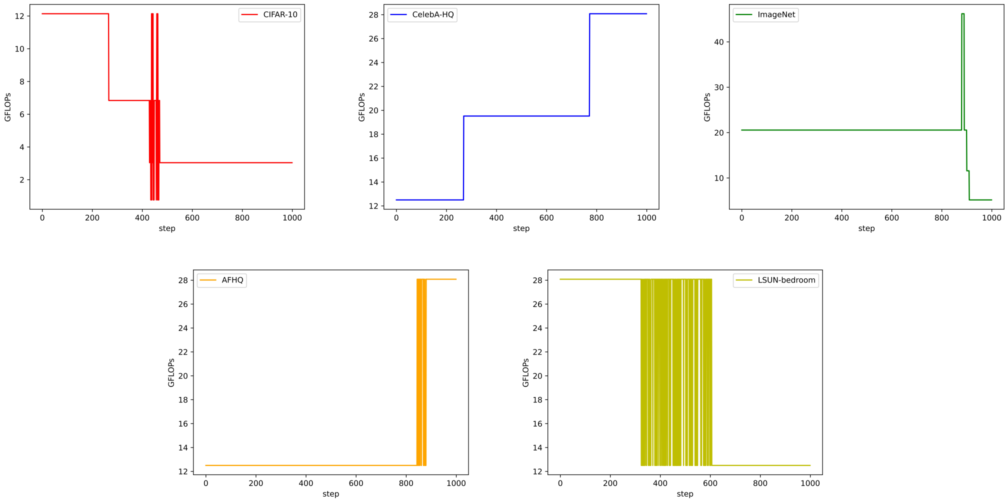
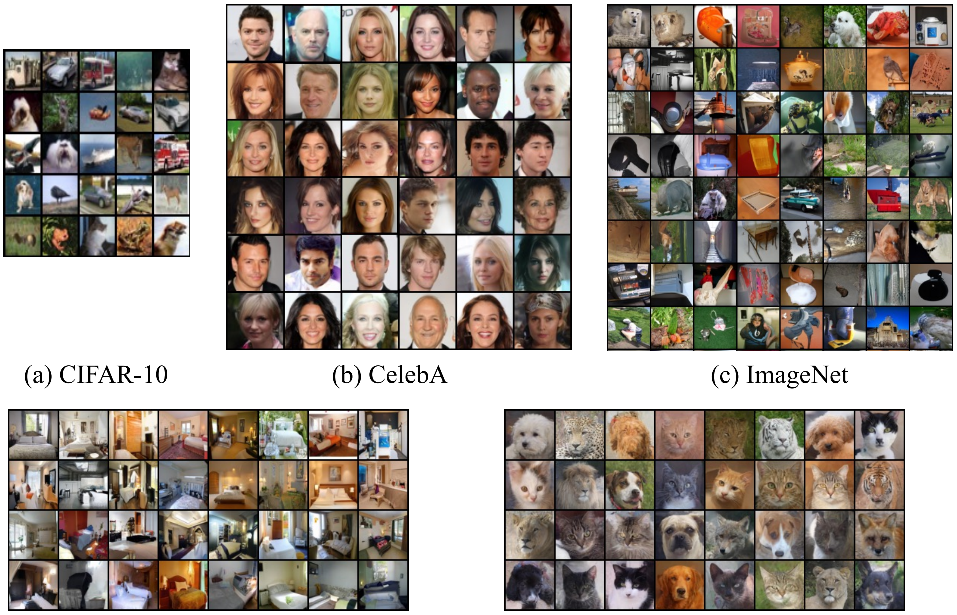


Figure B: Line graphs of the search results across all 5 datasets.

I Qualitative Result

We present the generation result of DDSM in Figure C. It shows that the efficient DDSM could produce high-quality images.



(a) CIFAR-10

(b) CelebA

(c) ImageNet

(d) LSUN-bedroom

(e) AFHQ

Figure C: Generated images of DDSM on CIFAR-10, CelebA, ImageNet, LSUN-bedroom, and AFHQ.

REFERENCES

Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16091–16101, 2023a.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Structural pruning for diffusion models. In *Advances in Neural Information Processing Systems*, 2023b.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *NeurIPS*, 2023.