Appendix of "AutoDiffusion: Training-Free Optimization of Time Steps and Architectures for Automated Diffusion Model Acceleration"

Lijiang Li[1], Huixia Li[2], Xiawu Zheng[1,3,4,5], Jie Wu[2], Xuefeng Xiao[2],
Rui Wang[2], Min Zheng[2], Xin Pan[2], Fei Chao[1]*, Rongrong Ji[1,3,4,5],
[1]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Department of Artificial Intelligence, School of Informatics,
Xiamen University. [2]ByteDance Inc. [3]Peng Cheng Laboratory.
[4]Institute of Artificial Intelligence, Xiamen University. [5]Fujian Engineering
Research Center of Trusted Artificial Intelligence Analysis and Application, Xiamen University.
lilijiang@stu.xmu.edu.cn, zhengxw01@pcl.ac.cn, wujie10558@gmail.com, {feichao, rrji}@xmu.edu.cn
{lihuixia, xiaoxuefeng.ailab, ruiwang.rw, zhengmin.666, panxin.321}@bytedance.com

1. Pseudo-code of Evolutionary Search

The evolution algorithm utilized in our method is elaborated in Alg. 1. Given a trained diffusion model, we sample candidates from search space randomly to form an initial population. During each iteration, we calculate FID score for each candidate in the population. After that, the Top k candidates with the lowest FID score are selected as parents. We then apply cross and mutation to these parents to generate a new population for the next iteration. The aforementioned process is iteratively executed until the predetermined maximum number of iterations is attained.

Algorithm 1 Evolutionary search

Input: Pre-trained diffusion model D, Number of searched time steps K, population size P, max iteration $MaxIter$, mutation probability p, the number of candidate generated from cross N_c and mutation N_m.

Output: The best candidate $cand^*$

1: $\mathrm{P}_0 = InitializePopulation(P)$
2: Topk $= \emptyset$
3: **for** $i = 1 : MaxIter$ **do**
4:     Samples = $GenerationProcess(D, \mathrm{P}_{i-1})$
5:     $\mathrm{FID}_{i-1} = CalculateFID(\text{Samples})$
6:     Topk = $UpdataTopk(\text{Topk}, \mathrm{P}_{i-1}, \mathrm{FID}_{i-1})$
7:     $\mathrm{P}_{\text{cross}} = cross(\text{Topk}, N_c)$
8:     $\mathrm{P}_{\text{mutation}} = mutation(\text{Topk}, N_m, p)$
9:     $\mathrm{P}_i = \mathrm{P}_{\text{cross}} + \mathrm{P}_{\text{mutation}}$
10: **end for**
11: $cand^* = Top1(\text{Topk})$

2. Experiments details and more samples on Stable Diffusion

For the experiments on Stable Diffusion [4], we utilize the official code and the released "sd-v1-4.ckpt" checkpoint[1]. We employ the validation set of COCO 2014 dataset and 10k generated samples to obtain the FID score for Fig. 4 in the primary manuscript. And Tab. 1 displays the detailed FID score corresponding to Fig. 4 in the primary manuscript. Additional sampling results on Stable Diffusion using DPM-Solver [3] with and without our method are reported in Fig. 1 and Fig. 2.

3. Experiments details and more samples on ADM

We use the official code and released checkpoint[2] for the experiments with ADM-G and ADM [1] on ImageNet, LSUN cat, and LSUN bedroom. In these experiments, we utilize 50k generated images and pre-computed sample batches from the reference datasets available in the codebase [3] of ADM to calculate the FID score of Tabs 2 and 3 in our main manuscript. Additional sampling results on ImageNet 64×64 and LSUN cat are reported in Fig. 3 and Fig. 4, respectively.

*Corresponding author: fchao@xmu.edu.cn
[1]https://github.com/CompVis/stable-diffusion
[2]https://github.com/openai/guided-diffusion
[3]https://github.com/openai/guided-diffusion/tree/main/evaluations

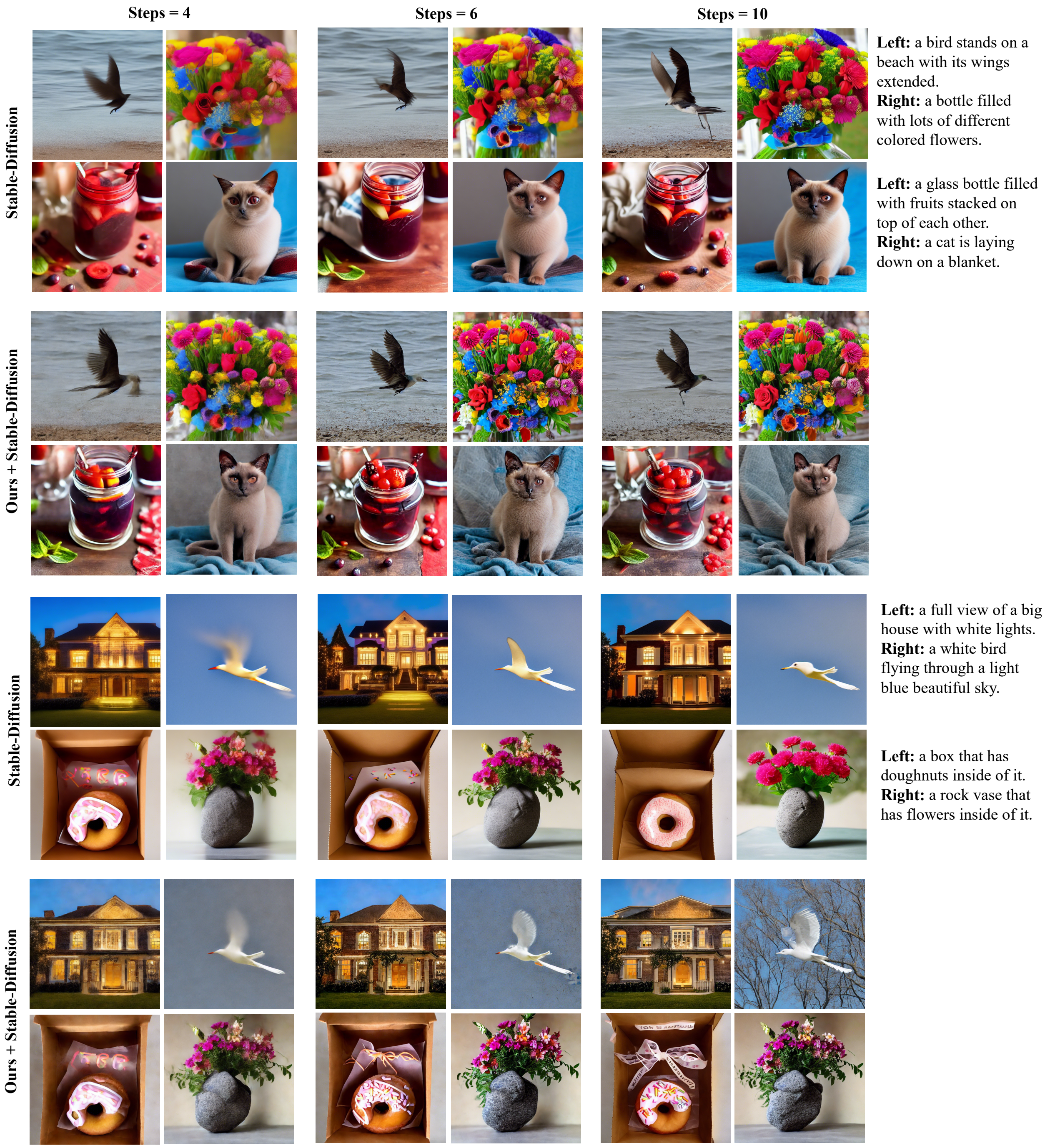

Figure 1. Samples obtained by Stable diffusion with and without our methods using the same random seed.

4. Ablation Study

4.1. Ablation on Performance Estimation

To assess the impact of performance estimation, we conduct experiments employing different evaluation metrics. Specifically, we replicate the experiment on ImageNet 64×64 with ADM-G using FID score, KID score, and KL-divergence as performance estimation. In these experiments, we only focus on time step optimization and use a complete noise prediction network. The results summarized in Tab. 2 indicate that there is little difference in the performance of FID score and KID score. This observation can

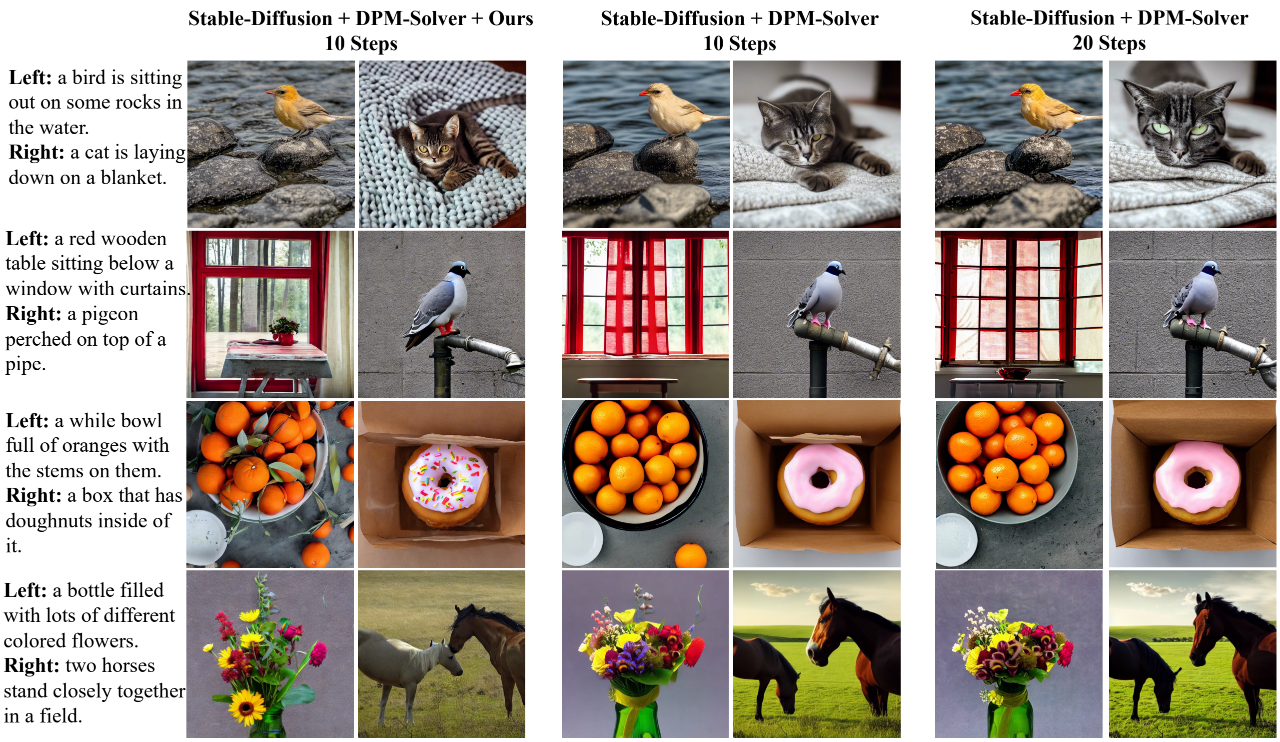

Figure 2. Samples generated by Stable Diffusion using DPM-Solver with our method at 10 steps are comparable to those generated only using DPM-Solver at 20 steps, and better than those generated only using DPM-Solver at 10 steps.

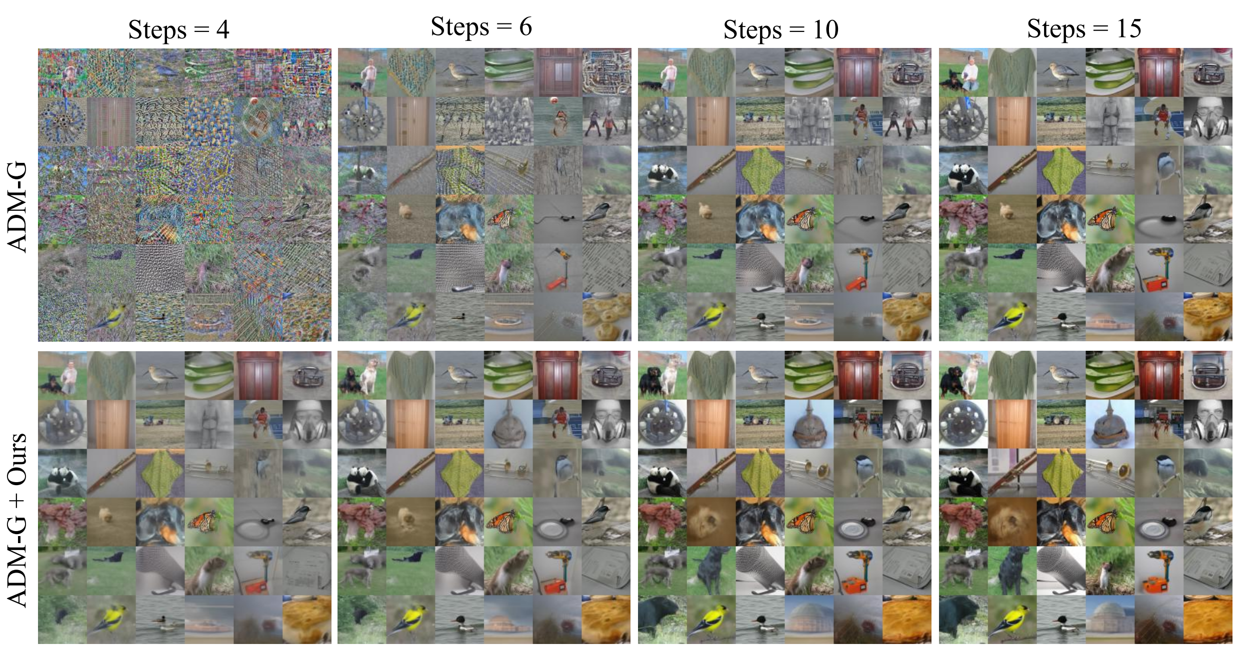

Figure 3. Samples generated by ADM pre-trained on ImageNet 64×64 cat with and without our method.

be attributed to the fact that both FID score and KID score gauge the distance between the statistical properties of the feature of generated samples and real samples. In contrast, the performance of KL-divergence is poor, which demon-

Ours	Steps	FID ↓ DPM-Solver	IS ↑ DPM-Solver	FID ↓ DDIM	IS ↑ DDIM	FID ↓ PLMS	IS ↑ PLMS
×	4	22.43	29.70	39.13	23.05	38.22	22.00
✓	4	18.22 (-4.21)	33.10 (+3.40)	26.72 (-12.41)	27.73 (+4.68)	20.94 (-17.28)	30.38 (+8.38)
×	6	17.36	34.03	18.87	31.63	32.40	24.41
✓	6	12.95 (-4.41)	34.26 (+0.23)	16.44 (-2.43)	33.70 (+2.07)	16.48 (-15.92)	33.58 (+9.17)
×	10	15.95	36.23	14.93	34.97	19.16	30.22
✓	10	12.67 (-3.28)	36.54 (+0.31)	14.06 (-0.87)	35.38 (+0.42)	13.57 (-5.59)	36.79 (+6.57)

Table 1. FID score and IS scores for Stable Diffusion using DPM-Solver [3], DDIM [5] and PLMS [2] with and without our method on COCO dataset, varying the number of time steps.

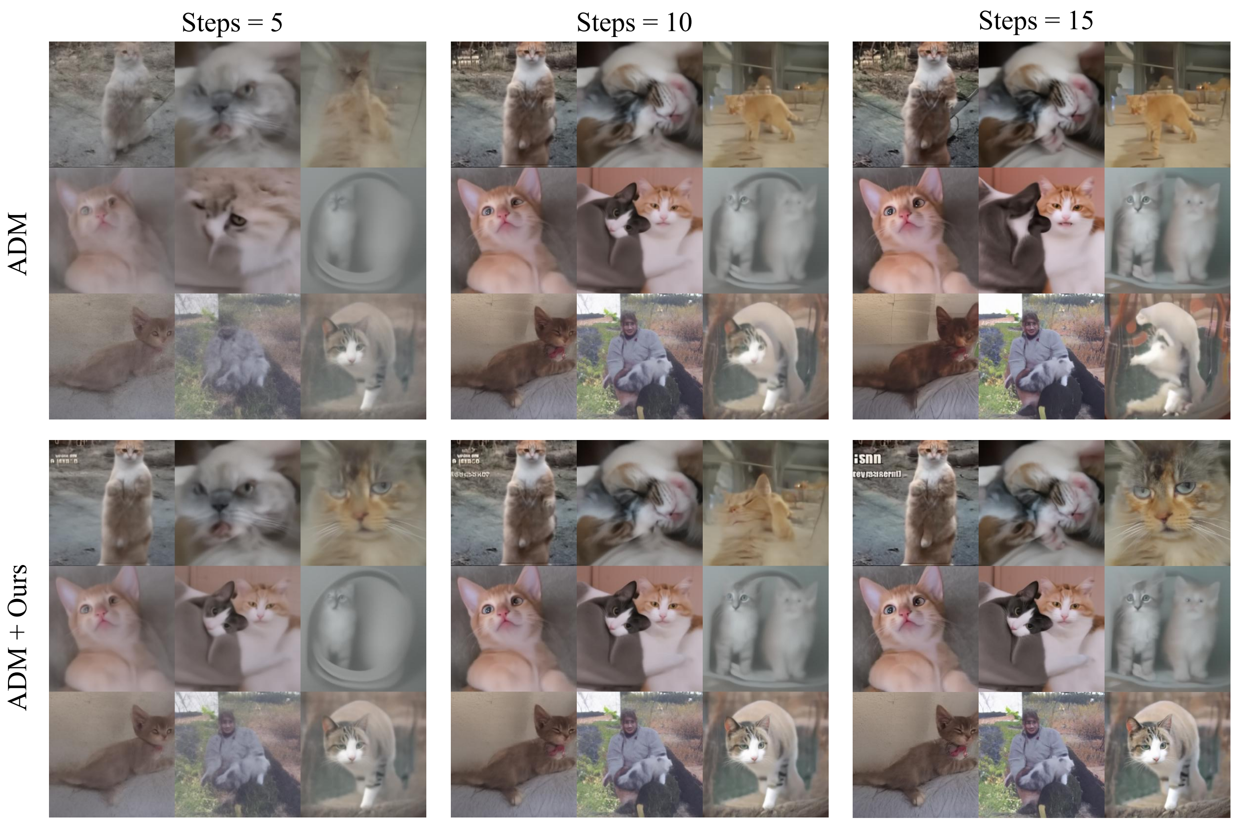

Figure 4. Samples generated by ADM pre-trained on LSUN cat with and without our method.

strates that KL-divergence is inadequate in estimating the performance of the time steps sequence properly.

4.2. Ablation on Search Algorithm

We conduct experiments to examine the impact of various search algorithms on experimental results. Specifically, we utilize evolutionary search and random search to search the optimal time steps sequence for ADM-G on ImageNet 64×64. The results presented in Tab. 3 illustrate that the selection of the search algorithm does not significantly influence the experimental results. Notably, We observe that even the time steps sequence searched by the simplistic random search algorithm produces better sample quality than the uniform time steps sequence.

5. Search Results

5.1. Time Steps Sequence

The optimal time step sequences in the evolutionary search for different diffusion models are presented in Tab. 4 and Tab. 5. Besides, Fig. 5 illustrates the occurrence number of time steps of the top-15 candidates in the evolutionary

Performance Estimation Strategy \Steps	4	6	10
FID score	17.86 / 34.88	11.17 / 43.47	6.24 / 57.85
KID score	21.06 / 30.78	12.68 / 39.42	9.72 / 42.60
KL-divergence	414.9 / 1.125	414.3 / 1.13	414.8 / 1.14

Table 2. FID score / IS score for the performance estimation ablation on ImageNet 64×64.

Method \Steps	4	6	10
Evolutionary Search	17.86 / 34.88	11.17 / 43.47	6.24 / 57.85
Random Search	18.84 / 34.17	11.17 / 43.02	7.05 / 51.43
Uniform Time steps	138.66 / 7.06	23.71 / 31.53	8.86 / 46.50

Table 3. FID score / IS score for the search algorithm ablation on ImageNet 64×64.

Diffusion Models	**Dataset**	**Optimal Time Steps**
ADM-G + DDIM	ImageNet 64×64	[926, 690, 424, 153]
Stable Diffusion + PLMS	COCO	[848, 598, 251, 21]
Stable Diffusion + DPM-Solver	COCO	[0.9261, 0.7183, 0.5005, 0.2857, 0.0150]

Table 4. Optimal time steps sequence with length 4 for different diffusion models.

Diffusion Models	**Dataset**	**Optimal Time Steps**
ADM-G + DDIM	ImageNet 64×64	[123, 207, 390, 622, 830, 948]
Stable Diffusion + PLMS	COCO	[19, 130, 335, 519, 695, 931]
Stable Diffusion + DPM-Solver	COCO	[0.9261, 0.6670, 0.5005, 0.3340, 0.1548, 0.0150, 0.0120]

Table 5. Optimal time steps sequence with length 6 for different diffusion models

Index of removed model layers	Steps	N_{max}
{[], [], [], [55]}	4	232
{[], [], [], [], [], [52]}	6	350
{[], [], [], [], [], [], [30, 10, 39, 4, 15, 46, 49, 54, 8], [], [], []}	10	580

Table 6. Index of removed model layers in the optimal architecture searched for ADM-G on ImageNet 64×64. "[]" means no layer is removed at corresponding time step.

search. In these experiments, the max time step of Stable Diffusion with DPM-Solver is 1, while the other diffusion models are 1000. When searching the optimal time steps for Stable Diffusion with DPM-Solver, we follow the strategy of DPM-Solver that uses the time steps sequence with a length of (Steps + 1)[4]. We observe that the optimal time steps tend to cluster within a specific interval. In addition, the distribution of these optimal time steps markedly differs between ADM-G and Stable Diffusion due to their distinct guidance scales.

5.2. Model Architectures

The optimal architecture layers in the evolutionary search for ADM-G on ImageNet 64×64 are shown in Tab. 6. In these experiments, we number each layer of the complete noise prediction network ascending from the input layer to the output layer. As described in the main manuscript, we constrain the sum of model layers at each time step to be less than N_{max}. And the complete noise prediction network comprises 58 model layers. We observe that the number of removed model layers is higher when $N_{\text{max}} = 580$ compared to $N_{\text{max}} = 232$ and $N_{\text{max}} = 350$. This observation highlights an increase in model redundancy with an increase in the number of time steps.

[4]https://github.com/CompVis/stable-diffusion/blob/main/ldm/models/diffusion/dpm_solver/dpm_solver.py

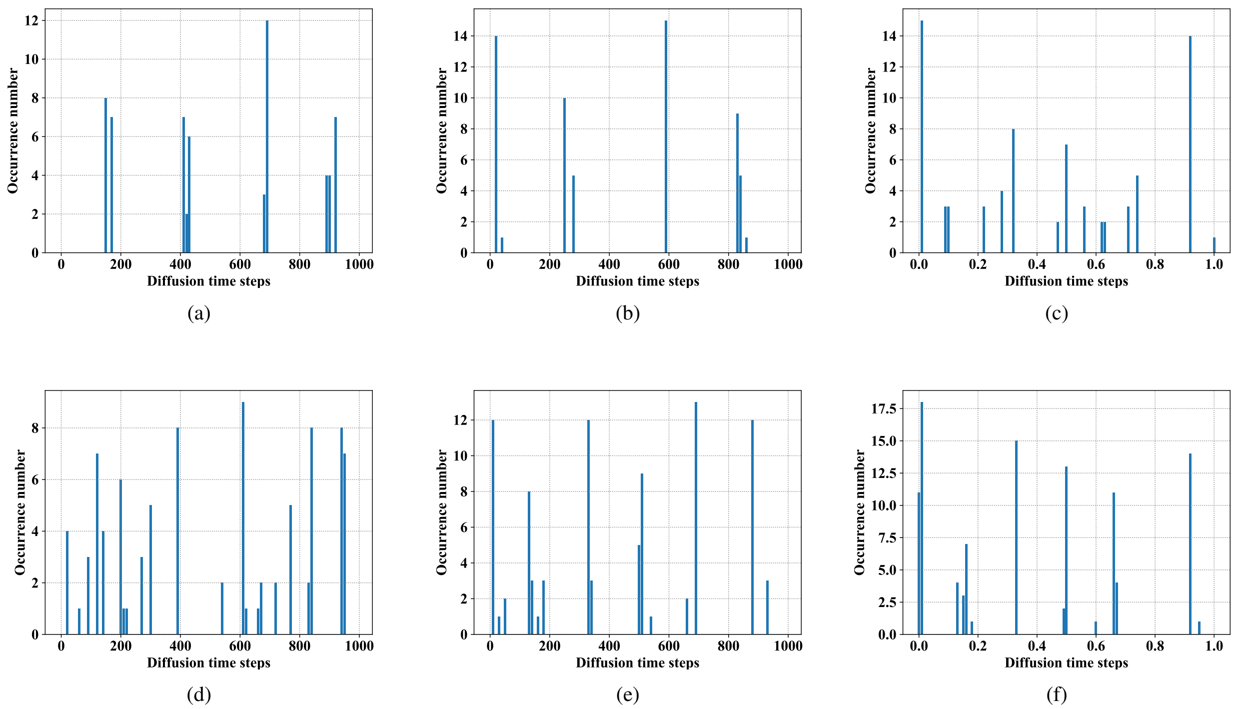

Figure 5. The occurrence number of time steps of top-15 candidates in Evolutionary search. (a). Occurrence number of time steps in the top-15 sequence with length 4 for ADM-G using DDIM on ImageNet64×64. (b). Occurrence number of time steps in the top-15 sequence with length 4 for Stable Diffusion using PLMS on COCO dataset. (c). Occurrence number of time steps in the top-15 sequence with length 4 for Stable Diffusion using DPM-Solver on COCO dataset. (d). Occurrence number of time steps in the top-15 sequence with length 6 for ADM-G using DDIM on ImageNet64×64. (e). Occurrence number of time steps in the top-15 sequence with length 6 for Stable Diffusion using PLMS on COCO dataset. (f). Occurrence number of time steps in the top-15 sequence with length 6 for Stable Diffusion using DPM-Solver on COCO dataset.

References

[1] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[2] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

[3] 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. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[5] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.