Appendix for All are Worth Words: A ViT Backbone for Diffusion Models

Fan Bao[1], Shen Nie[2], Kaiwen Xue[2], Yue Cao[3], Chongxuan Li[2], Hang Su[1], Jun Zhu[1]
[1]Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center
[1]Tsinghua-Bosch Joint ML Center, THBI Lab,Tsinghua University, Beijing, 100084 China
[2]Gaoling School of Artificial Intelligence, Renmin University of China,
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, China
[3]Beijing Academy of Artificial Intelligence
bf19@mails.tsinghua.edu.cn; nieshen@ruc.edu.cn; {kevin.kaiwenxue, caoyue10}@gmail.com
chongxuanli@ruc.edu.cn; {suhangss, dcszj}@tsinghua.edu.cn

A. Experimental Setup

We list the experimental setup for U-ViT presented in the main paper in Table 1.

Dataset	CIFAR10	CelebA 64×64	ImageNet 64×64	ImageNet 256×256	ImageNet 512×512	MS-COCO
Latent space	×	×	×	✓	✓	✓
Latent shape	-	-	-	32×32×4	64×64×4	32×32×4
Image decoder	-	-	-	ft-EMA	ft-EMA	original
Batch size	128	128	1024	1024	1024	256
Training iterations	500K	500K	300K	500K	500K	1M
Warm-up steps	2.5K	5K	5K	5K	5K	5K
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Learning rate	2e-4	2e-4	3e-4	2e-4	2e-4	2e-4
Weight decay	0.03	0.03	0.03	0.03	0.03	0.03
Betas	(0.99, 0.999)	(0.99, 0.99)	(0.99, 0.99)	(0.99, 0.99)	(0.99, 0.99)	(0.9, 0.9)
Noise schedule	VP	VP	VP	SD	SD	SD
Sampler	EM	EM	DPM-Solver	DPM-Solver	DPM-Solver	DPM-Solver
Sampling steps	1K	1K	50	50	50	50
CFG	×	×	×	✓	✓	✓
p_{uncond}	-	-	-	0.1	0.1	0.1
Guidance strength	-	-	-	0.4	0.7	1
Convolution	✓	✓	✓	×	×	✓

Table 1. The experimental setup for U-ViT in the main paper. "ft-EMA" and "original" correspond to different weights of the image decoder provided in https://huggingface.co/stabilityai/sd-vae-ft-ema. "VP" represents the continuous-time variance preserving noise schedule [11]. "SD" represents the discrete-time noise schedule used in Stable Diffusion [9]. "EM" represents the Euler-Maruyama SDE sampler [11]. "DPM-Solver" represents the DPM-Solver ODE sampler [6]. "p_{uncond}" represents the unconditional training probability in classifier free guidance (CFG). "Convolution" represents whether to add a 3×3 convolutional block before output.

In our early experiments, we try learning rates between 1e-4 and 5e-4, and find that a learning rate of 2e-4 performs well for all datasets. On ImageNet 64×64, a learning rate of 3e-4 could further improve the performance. We try weight decay between 0.01 and 0.05, and find that a weight decay of 0.03 performs well for all datasets. We try the running coefficients β_1, β_2 of AdamW among $\{0.9, 0.99, 0.999\}$, and find that $(\beta_1, \beta_2) = (0.99, 0.99)$ performs well for all datasets. On CIFAR10, $\beta_2 = 0.999$ could further improve the performance. On MS-COCO, $(\beta_1, \beta_2) = (0.9, 0.9)$ could further improve the performance. We train with mixed precision for efficiency, and the training time and devices are listed in

Table 2. Besides, the training memory of U-ViT can be greatly reduced with the gradient checkpointing trick. For example, the memory for forward and backward on a single A100 can be reduced from 53GB to 10GB when training U-ViT-L/2 with a batch size of 128 on ImageNet 256×256.

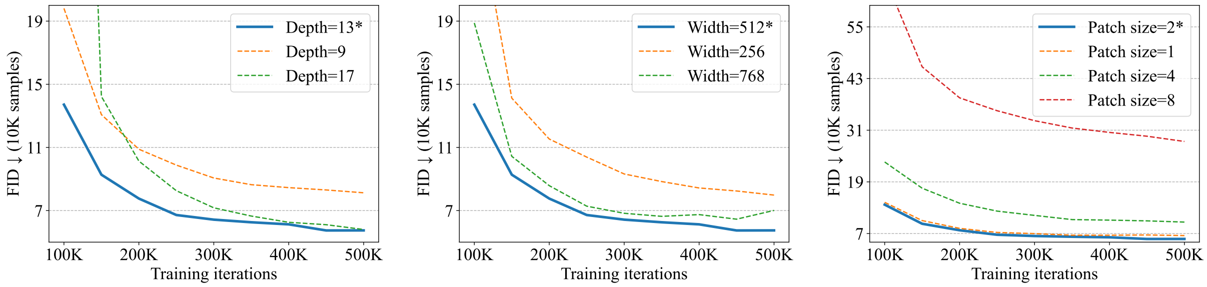
During inference, with 1 A100, generating 500 samples with DPM-Solver takes around 19 seconds, 34 seconds, 59 seconds, 89 seconds, with U-ViT-S, U-ViT-M, U-ViT-L, U-ViT-H respectively. The time would double if classifier-free guidance is used.

Dataset	Model	Training devices	Training time	Training iterations
CIFAR10	U-ViT-S/2	4 GeForce RTX 2080 Ti	24 hours	500K
CelebA	U-ViT-S/4	4 GeForce RTX 2080 Ti	24 hours	500K
ImageNet 64×64	U-ViT-M/4	8 A100	59 hours	300K
ImageNet 64×64	U-ViT-L/4	8 A100	100 hours	300K
ImageNet 256×256	U-ViT-L/2	8 A100	100 hours	300K
ImageNet 256×256	U-ViT-H/2	8 A100	208 hours	500K
ImageNet 512×512	U-ViT-L/4	8 A100	166 hours	500K
ImageNet 512×512	U-ViT-H/4	8 A100	208 hours	500K
MS-COCO	U-ViT-S/2	4 A100	60 hours	1M
MS-COCO	U-ViT-S/2 (deep)	4 A100	74 hours	1M

Table 2. The training devices and time.

B. Effect of Depth, Width and Patch Size

In Figure 1, we present scaling properties of U-ViT by studying the effect of the depth (i.e., the number of layers), width (i.e., the hidden size D) and patch size on CIFAR10.



(a) Depth (#layers)

(b) Width (hidden size)

(c) Patch size

Figure 1. Effect of depth, width and patch size. The one marked with * corresponds to the setting of U-ViT-S/2.

C. Details of the U-Net Baseline on MS-COCO

We employ the U-Net with cross attention provided by LDM [9] for the baseline. The U-Net is performed on the 32×32 resolution latent representation, and down-samples it to 16×16, 8×8 and 4×4 resolution. The number of channels is 128 at 32×32 resolution, and 256 at other resolutions. Each resolution has 2 residual blocks. The U-Net performs self attention and cross attention at 16×16 and 8×8 resolution. Such a configuration leads to a total of 53M parameters, which is comparable to 45M of U-ViT-Small for a fair comparison. We use the AdamW optimizer with weight decay set to 0.01 and running coefficients β_1, β_2 set to (0.9, 0.999), which are the setting used across LDM [9]. We tune the learning rate of U-Net and find 2e-4 performs the best. The training iterations and the batch size of U-Net are the same to U-ViT for a fair comparison.

D. Results of Other Metrics and Configurations on ImageNet

We present results of FID [3], sFID [7], inception score (IS) [10], precision and recall [5] on ImageNet in Table 3. Our U-ViT is still comparable to state-of-the-art diffusion models based on U-Net on these metrics, and meanwhile has comparable if not smaller GFLOPs.

ImageNet 64×64	#Params	GFLOPs	FID↓	sFID↓	IS↑	Precision↑	Recall↑
ADM [2]	296M	110	2.07	4.29	-	0.74	0.63
U-ViT-M/4 (VP, trained 300K, w/ conv)	131M	35	5.85	4.09	33.71	0.69	0.61
U-ViT-L/4 (VP, trained 300K, w/ conv)	287M	77	4.26	3.77	40.66	0.71	0.62

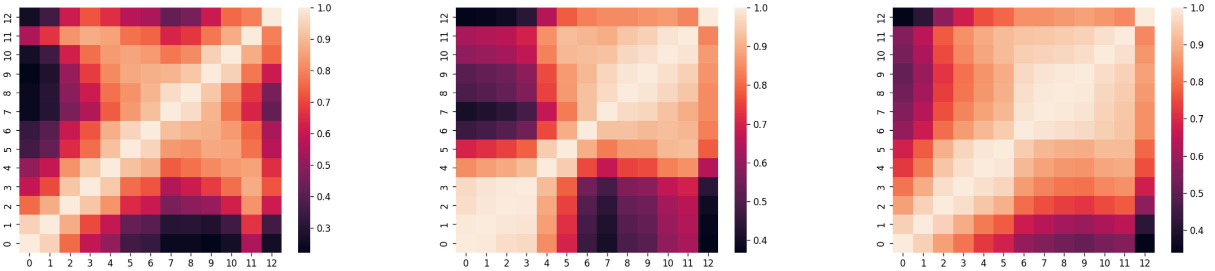
ImageNet 256×256	#Params	GFLOPs	FID↓	sFID↓	IS↑	Precision↑	Recall↑
ADM-G, ADM-U [2]	296M + 65M (Cls) + 312M (SR)	110 + 19 (Cls) + 632 (SR)	3.94	6.14	215.84	0.83	0.53
LDM [9]	400M + 55M (AE)	104 + 336 (AE)	3.60	-	247.67	0.87	0.48
U-ViT-L/2 (VP, trained 300K, w/ conv, original, p_{uncond}=0.15)	287M + 84M (AE)	77 + 312 (AE)	3.40	6.63	219.94	0.83	0.52
U-ViT-H/2 (VP, trained 300K, w/ conv, original, p_{uncond}=0.1)	501M + 84M (AE)	133 + 312 (AE)	3.10	6.70	250.82	0.84	0.53
U-ViT-H/2 (VP, trained 300K, w/o conv, original, p_{uncond}=0.1)	501M + 84M (AE)	133 + 312 (AE)	3.74	8.04	244.47	0.84	0.51
U-ViT-H/2 (SD, trained 300K, w/ conv, original, p_{uncond}=0.1)	501M + 84M (AE)	133 + 312 (AE)	3.14	7.81	229.03	0.82	0.55
U-ViT-H/2 (SD, trained 300K, w/o conv, original, p_{uncond}=0.15)	501M + 84M (AE)	133 + 312 (AE)	2.90	7.70	242.59	0.81	0.56
U-ViT-H/2 (SD, trained 300K, w/o conv, original, p_{uncond}=0.1)	501M + 84M (AE)	133 + 312 (AE)	2.78	7.55	251.83	0.82	0.56
U-ViT-H/2 (SD, trained 500K, w/o conv, original, p_{uncond}=0.1)	501M + 84M (AE)	133 + 312 (AE)	2.65	8.17	260.34	0.81	0.57
U-ViT-H/2 (SD, trained 500K, w/o conv, ft-EMA, p_{uncond}=0.1)	501M + 84M (AE)	133 + 312 (AE)	2.29	5.68	263.88	0.82	0.57

ImageNet 512×512	#Params	GFLOPs	FID↓	sFID↓	IS↑	Precision↑	Recall↑
ADM-G, ADM-U [2]	422M + 43M (Cls) + 309M (SR)	307 + 21 (Cls) + 2506 (SR)	3.85	5.86	221.72	0.84	0.53
U-ViT-L/4 (VP, trained 500K, w/ conv, original, p_{uncond}=0.15)	287M + 84M (AE)	77 + 1260 (AE)	4.67	5.87	213.28	0.87	0.45
U-ViT-H/4 (SD, trained 500K, w/o conv, original, p_{uncond}=0.1)	501M + 84M (AE)	133 + 1260 (AE)	4.34	8.44	261.13	0.84	0.48
U-ViT-H/4 (SD, trained 500K, w/o conv, ft-EMA, p_{uncond}=0.1)	501M + 84M (AE)	133 + 1260 (AE)	4.05	6.44	263.79	0.84	0.48

Table 3. Results of FID [3], sFID [7], inception score (IS) [10], precision and recall [5] on ImageNet. We also show the number of parameters as well as the GFLOPs.

E. CKA Analysis

Centered kernel alignment (CKA) is widely used to analyze similarity between hidden representations in deep neural networks [1, 4, 8]. In this section, we use the CKA method to analyze hidden representations of networks that employ three ways to combine long skip branches: (1) concatenation, i.e., $\text{Linear}(\text{Concat}(\boldsymbol{h}_m, \boldsymbol{h}_s))$; (2) addition, i.e., $\boldsymbol{h}_m + \boldsymbol{h}_s$; (3) no long skip connection. These three ways are elaborated in Section 3.1 in the main paper. We evaluate hidden representations after each transformer block and fix the input time as $t = 0.5$ on CIFAR10.



(a) Concatenation

(b) Addition

(c) No long skip connection

Figure 2. CKA analysis on hidden representations of networks that employ three ways to combine long skip branches. We analyze the similarity between hidden representations after each transformer block in the same network.

We find that the "addition" and "no long skip connection" settings share a similar phenomenon that neighboring blocks in the network have similar representations, e.g., blocks 0-3, 6-11 in Figure 2 (b), and blocks 0-5, 6-11 in Figure 2 (c). In contrast, the representations of neighboring blocks under the "concatenation" setting have low similarity, as shown in Figure 2 (a). Thus, the "concatenation" setting significantly changes the representations in the transformer, while the "addition" setting does not.

F. Compare with U-Net Under Similar Amount of Parameters and Computational Cost

On ImageNet 256×256, we also try replace our U-ViT with a U-Net with a similar amount of parameters and computational cost. The U-Net employs implementation from ADM [2]. We set the model channels as 320, the channel multiplier as (2, 2, 4), the number of residual blocks as 3, and employs attention at 2× and 4× down-sampling. This leads to a U-Net of 646M parameters and 135 GFLOPs, and our U-ViT has 501M parameters and 133 GFLOPs. We use the same optimizer configuration as ADM. As shown in Figure 3, our U-ViT consistently outperforms U-Net at different training iterations without classifier-free guidance. We also evaluate FID with 50K samples at 500K training iterations. With no classifier-free guidance, U-ViT obtains a FID of 6.58 and U-Net obtains a FID of 10.69. With a classifier-free guidance scale of 0.4, U-ViT obtains a FID of 2.29 and U-Net obtains a FID of 2.66. Under both settings, our U-ViT outperforms U-Net.

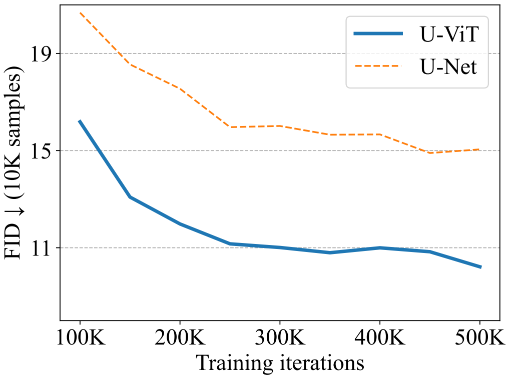


Figure 3. Compare with U-Net under similar amount of parameters and computational cost (w/o classifier-free guidance).

G. Additional Samples

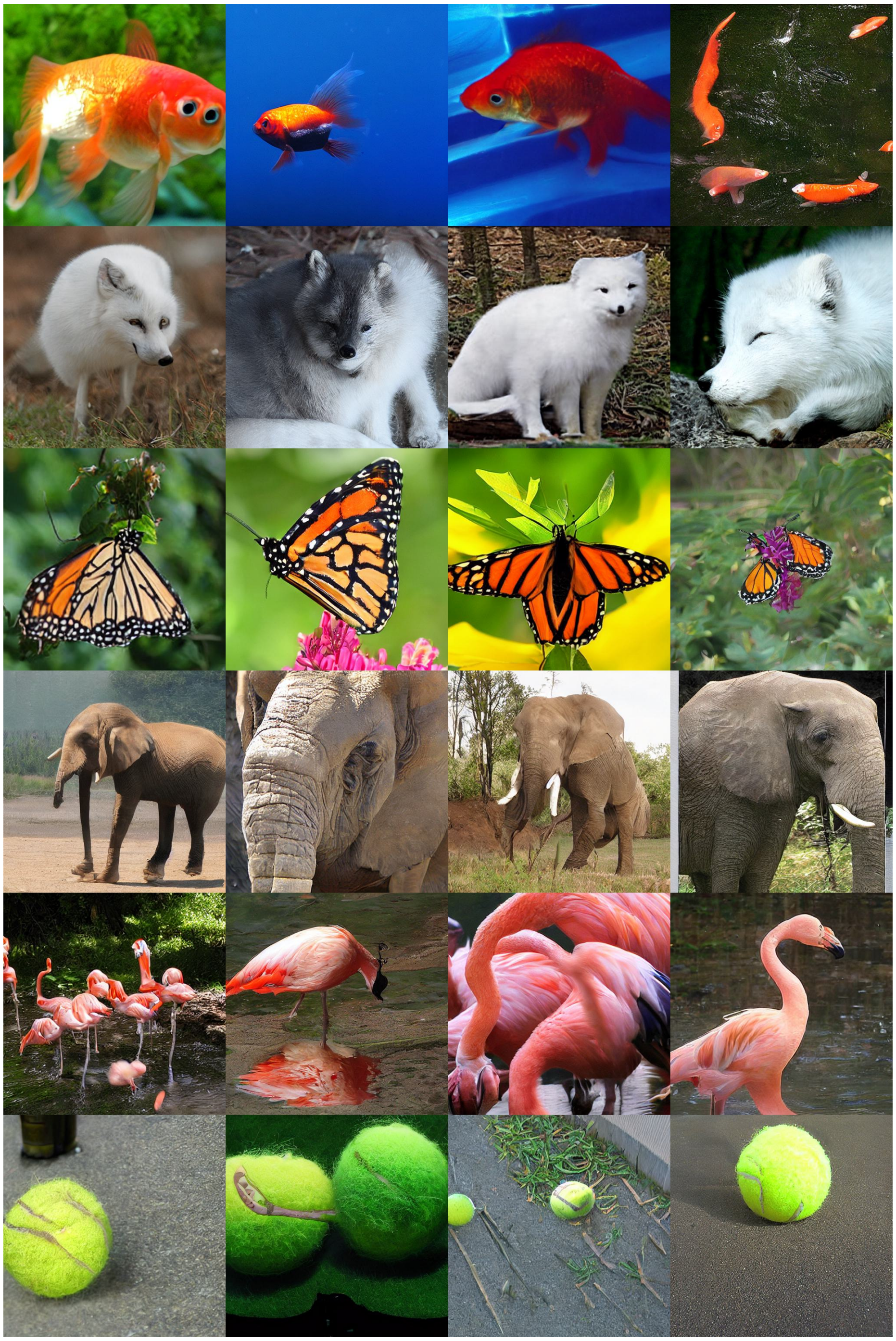


Figure 4. Generated samples on ImageNet 512×512, conditioned on goldfish (1), arctic fox (279), monarch butterfly (323), african elephant (386), flamingo (130), tennis ball (852).

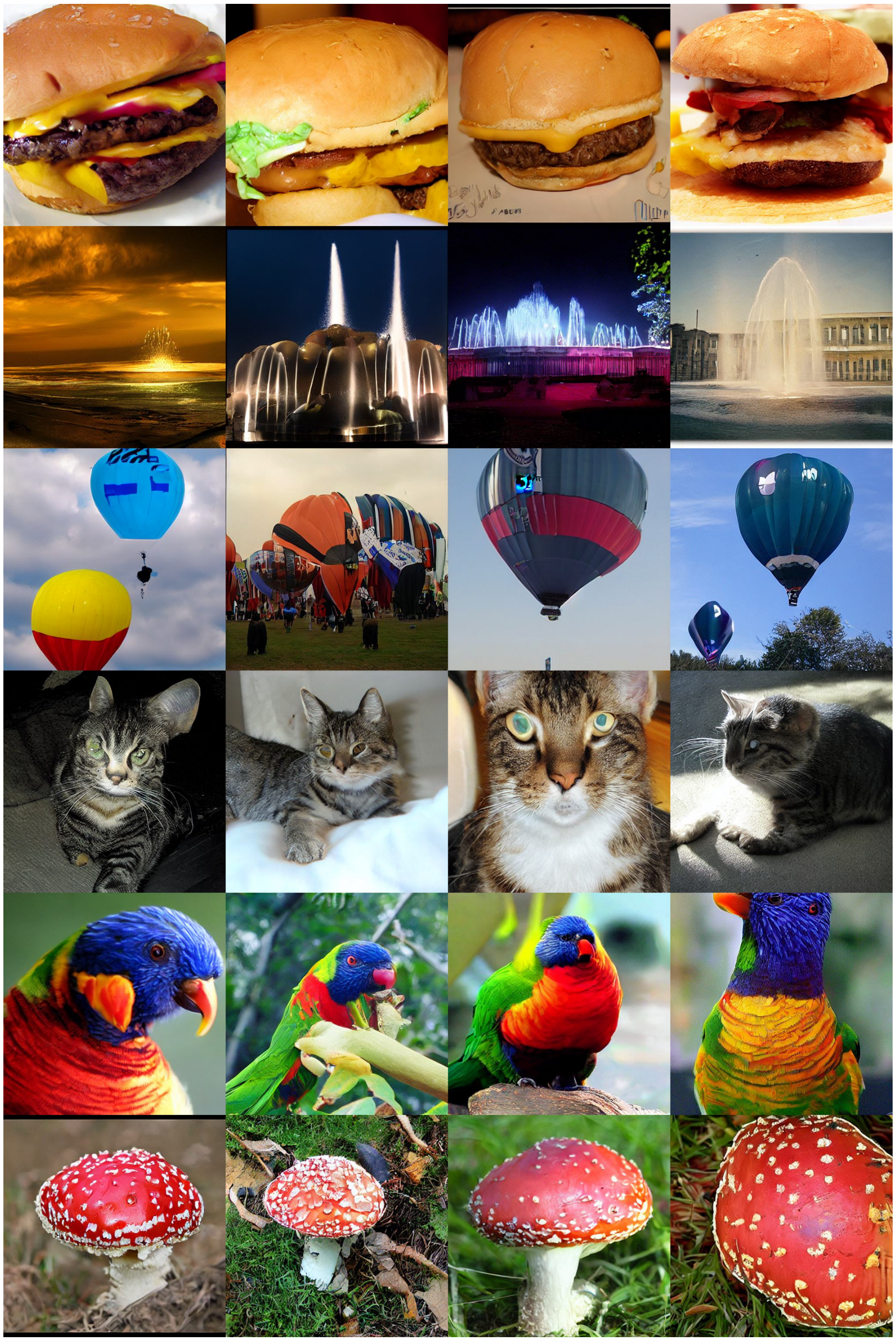


Figure 5. Generated samples on ImageNet 512×512, conditioned on cheeseburger (933), fountain (562), balloon (417), tabby cat (281), lorikeet (90), agaric (992).

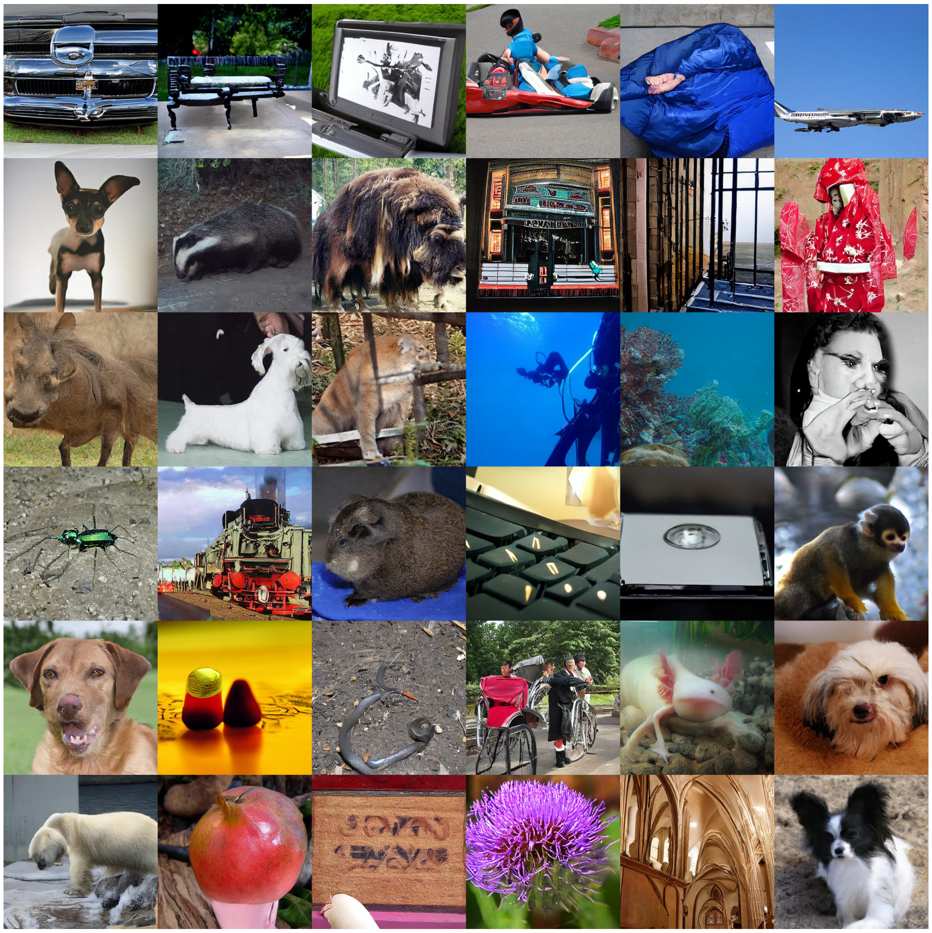


Figure 6. Random samples on ImageNet 512×512.

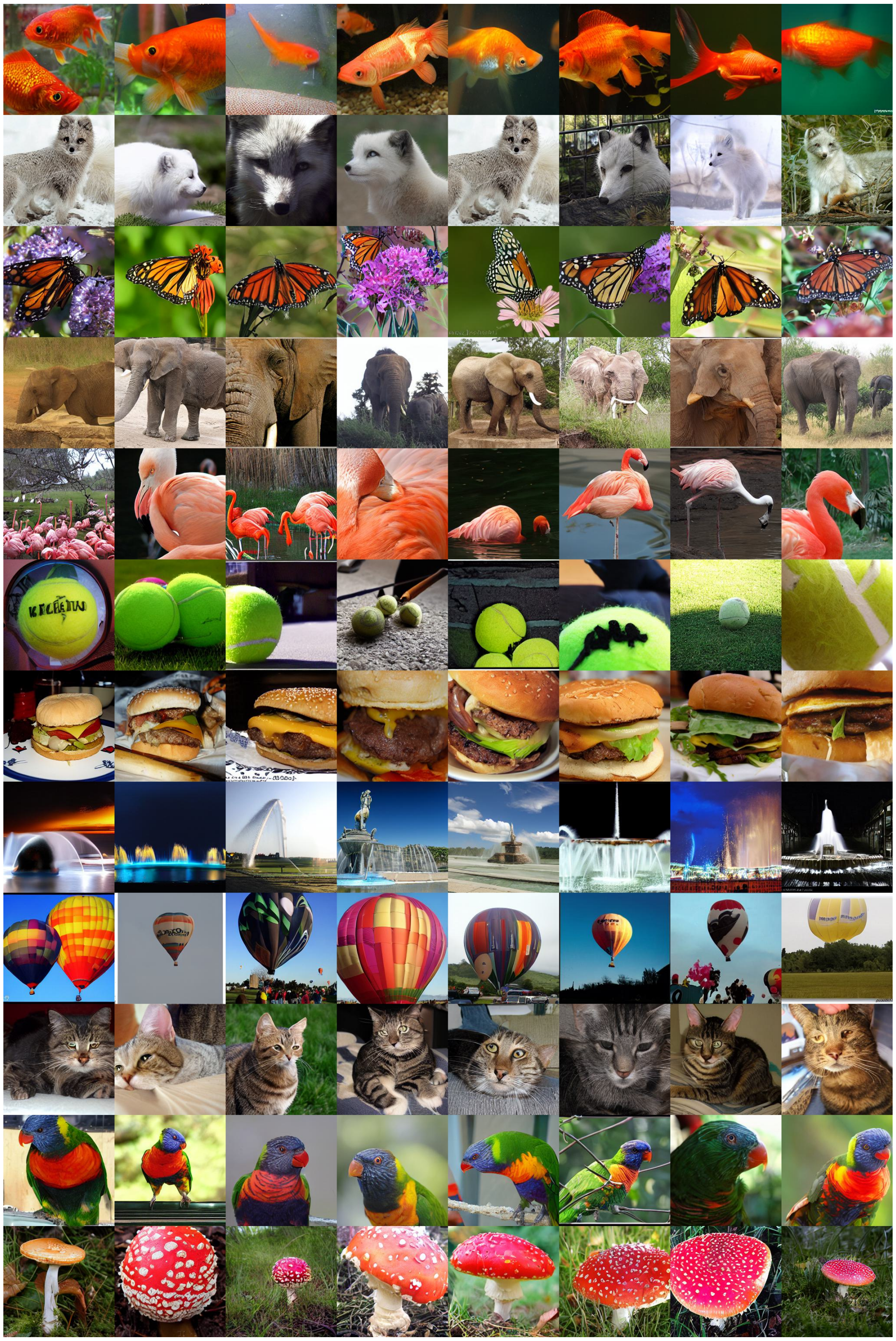


Figure 7. Generated samples on ImageNet 256×256, conditioned on goldfish (1), arctic fox (279), monarch butterfly (323), african elephant (386), flamingo (130), tennis ball (852), cheeseburger (933), fountain (562), balloon (417), tabby cat (281), lorikeet (90), agaric (992).

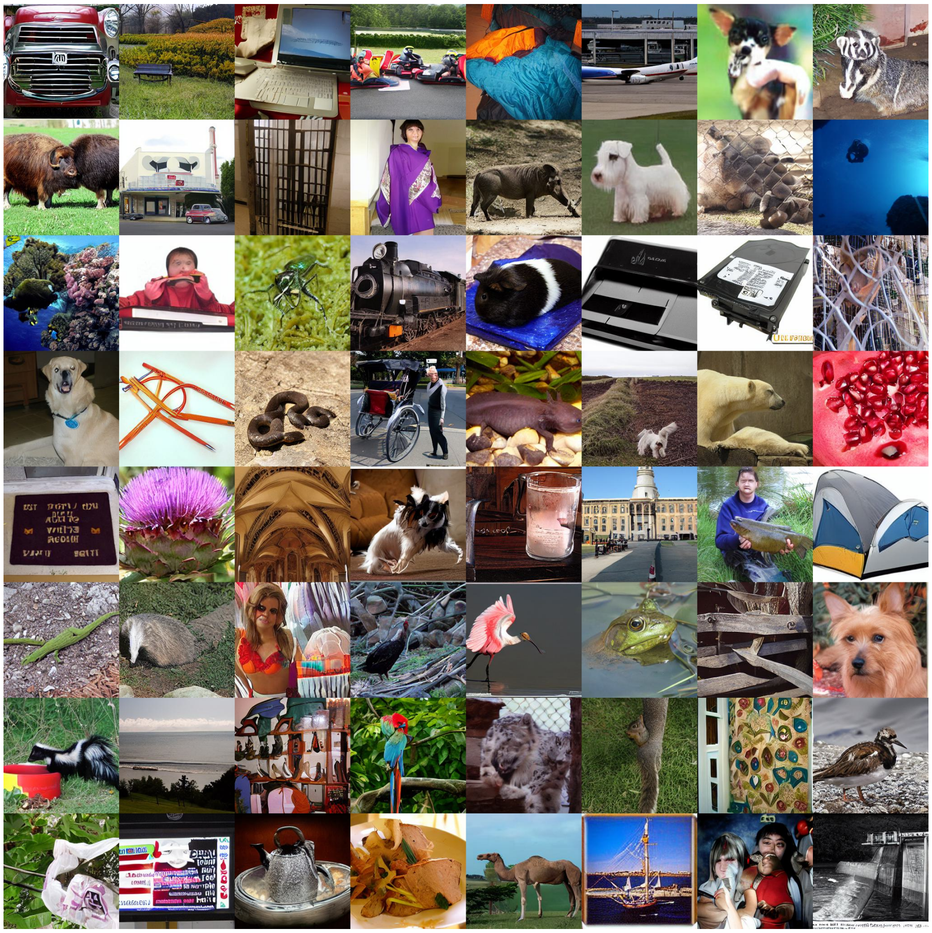
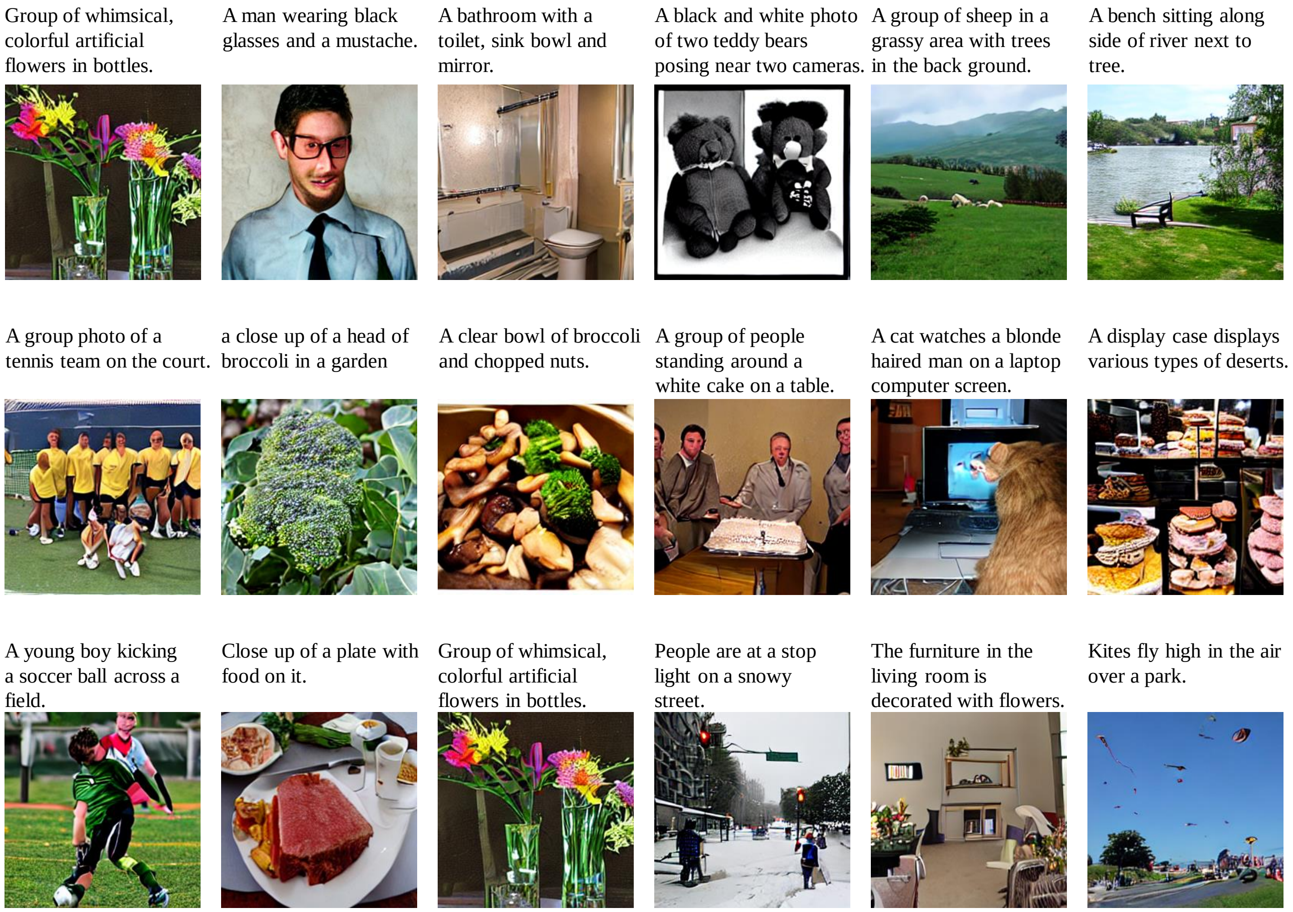


Figure 8. Random samples on ImageNet 256×256.



Group of whimsical, colorful artificial flowers in bottles.

A man wearing black glasses and a mustache.

A bathroom with a toilet, sink bowl and mirror.

A black and white photo of two teddy bears posing near two cameras.

A group of sheep in a grassy area with trees in the back ground.

A bench sitting along side of river next to tree.

A group photo of a tennis team on the court.

a close up of a head of broccoli in a garden

A clear bowl of broccoli and chopped nuts.

A group of people standing around a white cake on a table.

A cat watches a blonde haired man on a laptop computer screen.

A display case displays various types of deserts.

A young boy kicking a soccer ball across a field.

Close up of a plate with food on it.

Group of whimsical, colorful artificial flowers in bottles.

People are at a stop light on a snowy street.

The furniture in the living room is decorated with flowers.

Kites fly high in the air over a park.

Figure 9. Random samples on MS-COCO. Prompts are randomly drawn from the validation set.

References

[1] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012. 3

[2] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv preprint*, 2021. 3, 4

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 2, 3

[4] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 3

[5] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 3

[6] 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*, 2022. 1

[7] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 2, 3

[8] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. 3

[9] 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*, 2022. 1, 2, 3

[10] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 2, 3

[11] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations*, 2021. 1