DiffSketching: Sketch Control Image Synthesis with Diffusion Models Supplementary Material

Qiang Wang
wanqqiang@bupt.edu.cn
Di Kong
dikong@bupt.edu.cn
Fengyin Lin
fylin@bupt.edu.cn
Yonggang Qi[✉]
qiyg@bupt.edu.cn

Beijing University of Posts and Telecommunications, Beijing, China

1 Details on Methods

Fine-tuning procedure After pretraining diffusion models, we get the diffusion kernel $\hat{\varepsilon}_\theta$. We clone the latent noise x_T to $\hat{x}_T(\theta)$, and iterate through Eq. 3 to get x_0. The image is sampled through the reverse generation process. And guided by image identity loss $\mathcal{L}_i$ and sketch perceptual loss $\mathcal{L}_p$, the kernel $\hat{\varepsilon}_\theta$ is updated. We repeat this update process M times until the model converges, shown in Algorithm 1.

Algorithm 1 Fine-tuning DiffSketching model

```
input  : pretrained model $\hat{\varepsilon}_\theta$, timesteps $T$, fine-tuning iterations $M$
output: fine-tuned model $\hat{\varepsilon}_{\hat{\theta}}$

for $m = 1, 2, \cdots, M$ do
    Clone the latent $\hat{x}_T(\theta) \leftarrow x_T$.
    for $t = T, T-1, \cdots, 1$ do
        $x_{t-1} \leftarrow \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} x_t + \left( \sqrt{1-\alpha_{t-1}} - \sqrt{\frac{(1-\alpha_t)(\alpha_t - 1)}{\alpha_t}} \right) \hat{\varepsilon}_\theta$
        $\hat{\varepsilon}_{\hat{\theta}} \leftarrow \hat{\varepsilon}_\theta$
    end
    $\mathcal{L} \leftarrow \lambda \mathcal{L}_i + (1-\lambda) \mathcal{L}_p$
    Take a gradient step on $\nabla_{\hat{\theta}} \mathcal{L}$.
end
```

Classifier Guidance Prafulla *et al.* [2] proved that classifier guidance can not only conditionally control the synthesis, but also improve the quality and diversity of the generation.

According to the score-based conditioning trick proposed by Song *et al.* [11], we use the gradient $\nabla_{x_t} \log p_\theta(x_t, t)$ of the classifier to guide the reverse generation process $p_{\theta,\phi}(x_t, t \mid y)$ and can be substituted to a score function $\nabla_{x_t} \log p_\theta(x_t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta(x_t, t)$.

$$\begin{aligned}
\nabla_{x_t} \log p_{\theta,\phi}(x_t, t \mid y) &= \nabla_{x_t} \log \left(p_\theta(x_t, t)\, p_\phi(y \mid x_t, t)\right) \\
&= \nabla_{x_t} \log p_\theta(x_t, t) + \nabla_{x_t} \log p_\phi(y \mid x_t, t) \\
&= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta(x_t, t) + \nabla_{x_t} \log p_\phi(y \mid x_t, t)
\end{aligned}$$

We define a new noise prediction $\hat{\varepsilon}_\theta(x_t, t) = -\sqrt{1-\bar{\alpha}_t}\nabla_{x_t} \log p_{\theta,\phi}(x_t, t \mid y)$ and obtain the Eq. 4. The parameters of the classifier are fixed during the fine-tuning process.

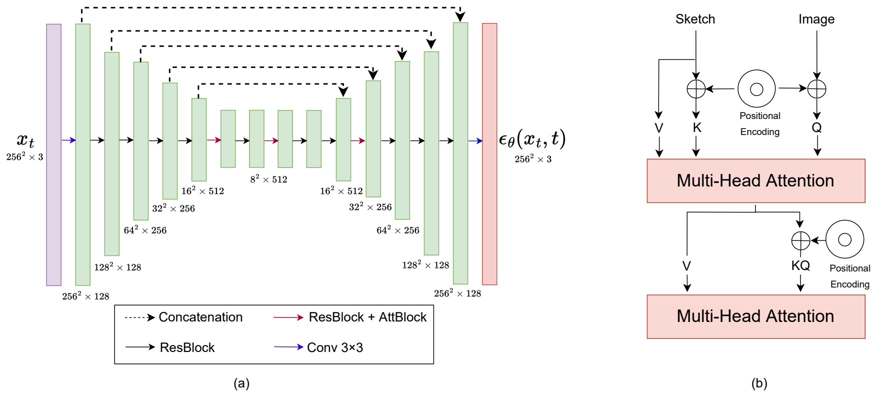

Figure 1: Network. (a) U-Net architecture with Wide Resnet blocks and single-head attention blocks of the Diffusion model that generates 256×256 images. (b) Multi-head attention structure integrates sketch and image information to achieve the purpose of image editing.

Reverse Diffusion Kernel The reverse diffusion kernel $\varepsilon_\theta(x_t, t)$ adopts U-net [9] architecture with single-head attention blocks and Wide Resnet blocks [14] shown in Fig. 1 (a). Skip connections connect the same spatial size layers and timestep t after the Transformer sinusoidal encoding [12] is embedded into each Wide Resnet block. In image editing task, we replace single-head attention blocks with multi-head attention blocks [7], as shown in Fig. 1 (b), to fuse sketch information to control synthesis.

2 Details on Experiment

Training details We trained Resnet50 [4] feature extractor on 40 epochs and achieved a classification accuracy of 99.99% on the Sketchy [10] dataset. We trained a classifier on ImageNet [1] for 1000 classes. Our model was fine-tuned for 80 epochs and consumed 10 hours on 2 NVIDIA T4 GPUs. We use Adam [6] optmizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to 0.0001. We use an EMA rate of 0.9999 and 16-bit precision, using loss scaling [8], for all experiments.

We train the same hyperparameters for our method. In particular, we set $\lambda = 0.4$ and batch size to 4 in all experiments. If the sketch input is less than 4, we reset the batch size to 1. In the inference stage, we set step interations T to 250. It takes 15 seconds to

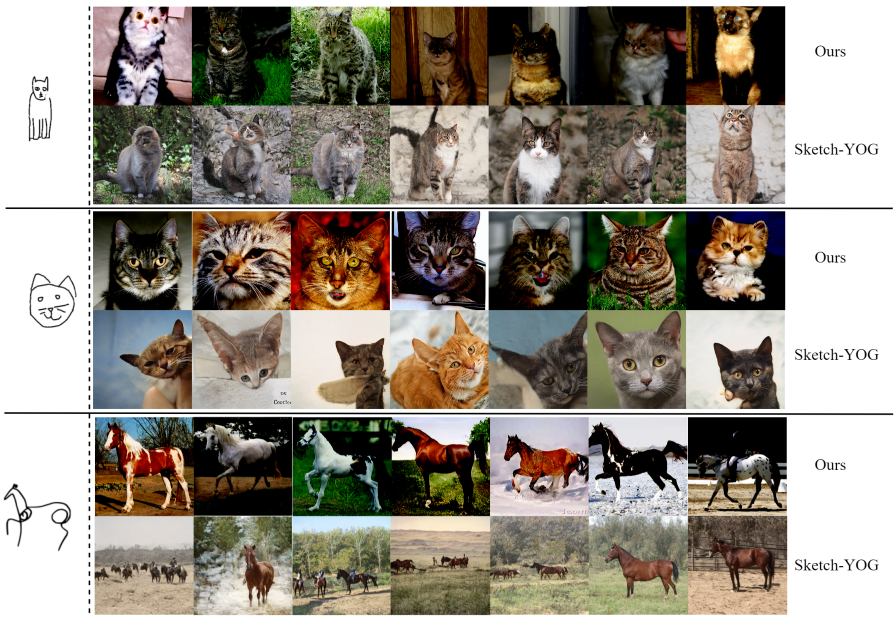

Figure 2: Comparison with Sketch-YOG. All the results of Sketch-YOG come from the published models. From top to bottom, we compare the three cases of "Standing cat", "Ganspace cat" and "Picasso horse" in turn.

get an image synthesis result. To further aid reproducibility, We release our source code https://github.com/XDUWQ/DiffSketching to reproduce our results.

Comparison with Sketch-YOG [13] Sketch-YOG is the state-of-the-art GAN-based method in sketch-to-image synthesis task. Both our method and Sketch-YOG have overcome the limitations of the small type and number of sketch-image paired datasets, and have been successful in customizing synthesis models for users. More qualitative comparisons are shown in Fig. 2. (i) The advantage of our method over Sketch-YOG is that we can achieve superior image sample quality, as shown in Table 1. (ii) In the bottom two rows of Fig. 2, we compare the results synthesized from Picasso's sketch of horse. Because Picasso's painting style is very unique and artistic, Sketch-YOG fails for the abstract style. Our model measures the perceptual distance of sketch, so it can solve the impact of bad sketch on the synthesis results. (iii) Sketch-YOG can only be customized on a few separate categories such as cat, horse, church, etc. If users want to work on other categories, they can only pretrain StyleGAN2 [5] to achieve that, which will cost a lot of time and computing resources. Our method is guided by classifiers, so only one model is needed to cover up to 1000 categories.

However, the limitation of both methods is that the models cannot be customized in real time and they require at least 30K iterations to train. Because the sampling of GAN-based models require only one forward pass, while the diffusion model requires many steps to iterate, the reasoning inference is slower than Sketch-YOG.

Interpolation In addition to spherical linear interpolation, we show more interpolation results from two different initial x_T, as shown in Fig. 4 and Fig. 5. (i) Linear interpolation: $x_T^{(\alpha)} = \beta x_T^{(0)} + (1-\beta)x_T^{(1)}$, where $\beta \sim (0,1)$. (ii) Trigonometric interpolation: $x_T^{(\alpha)} = cos(\theta)x_T^{(0)} + sin(\theta)x_T^{(1)}$, where $\theta \sim (0,\pi/2)$.

Additional qualitative results As shown in Fig. 3, the first five rows of input are from the

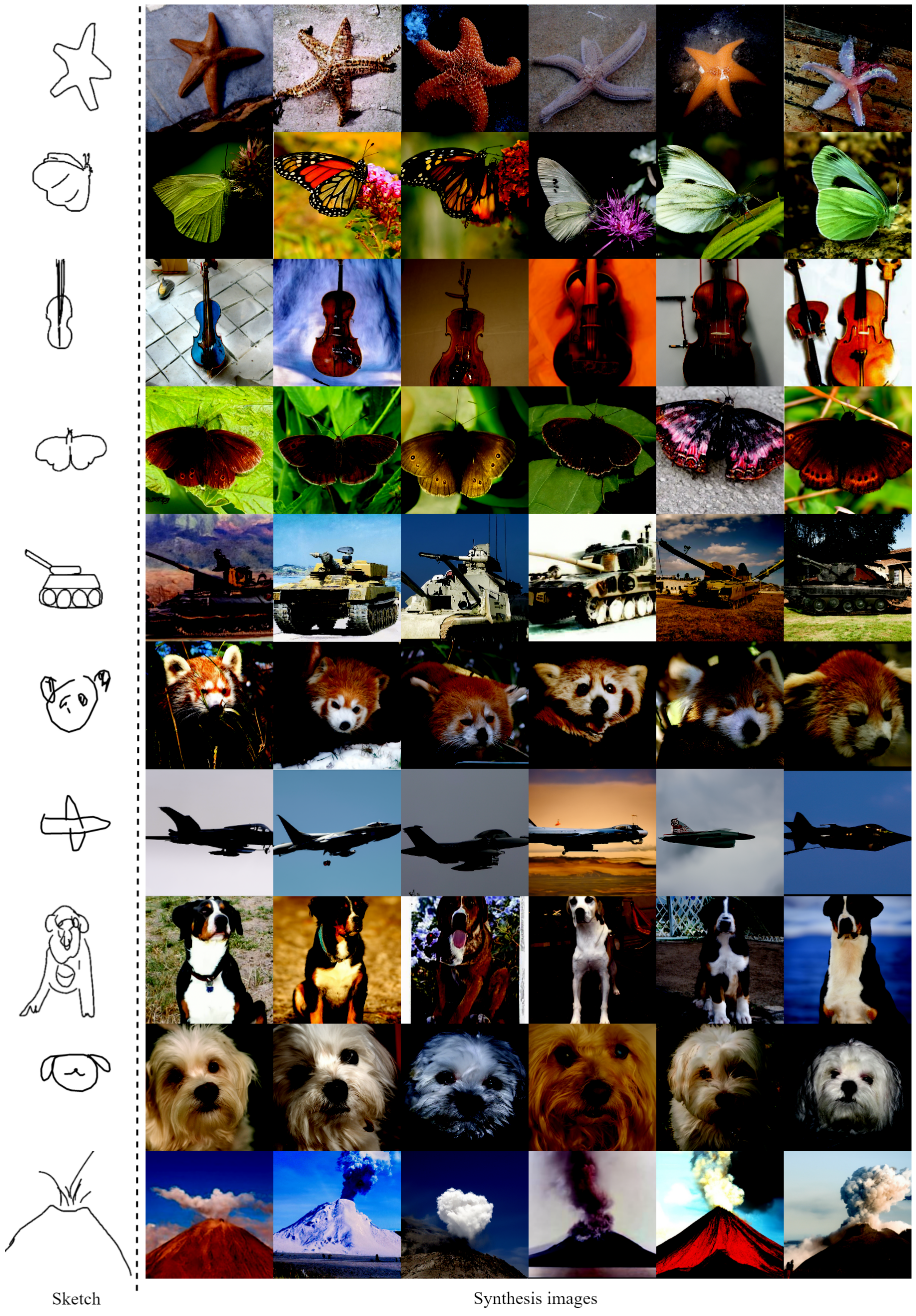

Figure 3: Additional qualitative results.

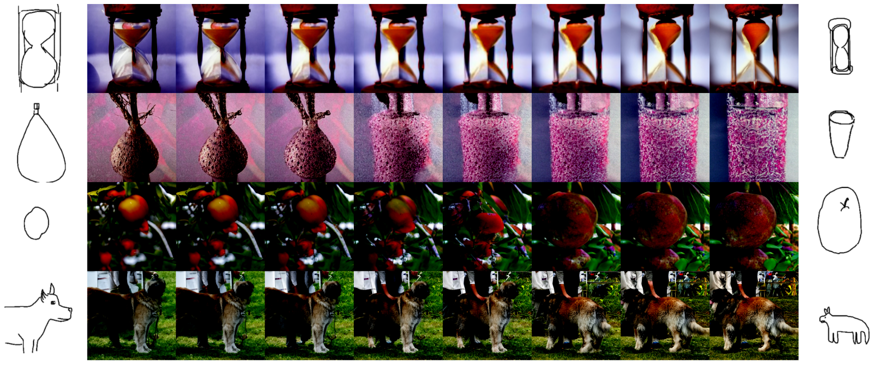

Figure 4: Linear interpolation.

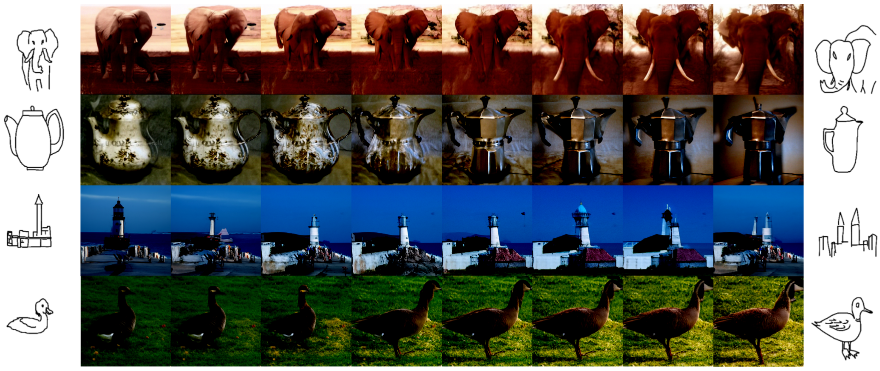

Figure 5: Trigonometric interpolation.

Sketchy [10] dataset and the last five rows are from the Quickdraw [3] dataset.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[3] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Troy Luhman and Eric Luhman. Diffusion models for handwriting generation. *arXiv preprint arXiv:2011.06704*, 2020.

[8] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[10] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35 (4):1–12, 2016.

[11] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14050–14060, 2021.

[14] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.