SCEdit: Efficient and Controllable Image Diffusion Generation via Skip Connection Editing

Supplementary Material

In the supplementary material, we provide more implementation details (Appendix A) including the dataset, architecture design, and hyperparameters used in training and inference. Then, we demonstrate the ablation experiments (Appendix B) with SC-Tuner and CSC-Tuner on different tasks. Furthermore, we showcase additional comparisons with existing methods and qualitative results (Appendix C).

A. Implementation details

A.1. Dataset description

In this work, we consider three datasets for our experiments: COCO Dataset [9], Customized Style Dataset [11], and LAION Dataset [16]. For the text-to-image generation setting, we utilize the well-known COCO2017 Captions, which consists of 118,287 training images and 591,753 captions for efficient fine-tuning, and Customized Style, which contains 30 training images of different styles for few-shot fine-tuning. We use LAION Dataset for the controllable image synthesis setting. The three datasets are illustrated in Tab. 1.

A.2. Hyperparameters

We provide an overview of the hyperparameters for all trained models, divided by the task in Tab. 2.

A.3. Architectures design

In the SCEdit framework, the central strategy involves editing the skip connections, which gives rise to two architectures: SC-Tuner for text-to-image generation and CSC-Tuner for controllable generation. These architectures are straightforward to implement and can be easily transferred to other similarly designed modules. In Alg. 1, we provide the forward function implementation of SCEdit written in PyTorch-like style.

A.4. Conditions for generation

We generally follow the implementations of condition extraction from ControlNet [22] and T2I-Adapter [13], with details as follows:

- **Canny Edge Map.** We employ canny edge detector [4], utilizing random thresholds during training and fixed thresholds with a low value of 100 and a high value of 200 during inference. The sample images are presented in Fig. 8a.
- **Depth Map.** We use MiDaS depth estimation [14] with default settings. The sample images are shown in Fig. 8b.
- **HED Boundary Map.** We use HED boundary detection [20] with default settings. The sample images are illustrated in Fig. 9a.
- **Semantic Segmentation Map.** We employ the UniFormer [8] semantic segmentation model, which was trained on the ADE20K [23] dataset. The sample images can be seen in Fig. 9b.
- **Pose Keypoint.** We employ OpenPose [5] as the human pose estimation model and visualize its prediction as conditions. The sample images are showcased in Fig. 9c.
- **Color Map.** We preserve the spatial hierarchical color

Table 1. The summary of the datasets for the experiments.

Dataset	**#Description**	**#Task**	**#Train**		**#Test**	
			image	prompt	image	prompt
Common Objects in Context (COCO)						
COCO2017 Captions [9]	common objects	text-to-image	118,287	591,753	5,000	25,014
Customized Style Dataset						
3D [11]	3D style	text-to-image (few-shot)	30	30	-	-
Anime [11]	ainme style	text-to-image (few-shot)	30	30	-	-
Flatillustration [11]	flatillustration style	text-to-image (few-shot)	30	30	-	-
Oilpainting [11]	oilpainting style	text-to-image (few-shot)	30	30	-	-
Sketch [11]	sketch style	text-to-image (few-shot)	30	30	-	-
Watercolor [11]	watercolor style	text-to-image (few-shot)	30	30	-	-
Large-scale Artificial Intelligence Open Network (LAION)						
LAION-ART [16]	filtered version	controllable generation	624,558	624,558	-	-

Table 2. The summary of the training and inference settings for the experiments.

Config	#Task		
	Text-to-image	Text-to-image (few-shot)	Controllable Generation
Dataset	COCO [9]	Customized Style [11]	LAION-ART (Filtered) [16]
Batch size	32	8	64
Optimizer	AdamW [10]	AdamW [10]	AdamW [10]
Weight decay	0.01	0.01	0.01
Learning rate	0.00005	0.00005	0.00005
Learning rate schedule	Constant	Constant	Constant
Training steps	100000	1500	100000
Data preprocess	Resize, CenterCrop	Resize, CenterCrop	Resize, CenterCrop
Resolution	512×512	512×512	512×512
Pre-trained	SD v1.5 [1]	SD v1.5 [1]	SD v2.1 [2]
Sampler	DDIM [18]	DDIM [18]	DDIM [18]
Sample steps	50	50	50
Guide scale	3.0	7.5	7.5
Device	A100×8	A100×1	A100×16
Training strategy	AMP / Float16	AMP / Float16	AMP / Float16
Library	SWIFT [12]	SWIFT [12]	SWIFT [12]

Algorithm 1 Implementation of SCEdit in PyTorch-like style.

```
# SC-Tuner
def forward(self, x, t=None, cond=dict()):
    ...
    # input_blocks
    hs = []
    for i, blk in enumerate(self.in_blks):
        h = blk(h, emb, context)
        hs.append(h)

    # middle_block
    h = self.mid_blk(h, emb, context)

    # output_blocks
    for i, blk in enumerate(self.out_blks):
        skip_h = self.tuners[i](hs.pop())
        h = torch.cat([h, skip_h], dim=1)
        h = blk(h, emb, context)
```

```
# Single CSC-Tuner
def forward(self, x, t=None, cond=dict()):
    ...
    # Dense Conv for conditions
    guid_hs = []
    guid_hint = self.in_hint_blks(hint, emb,
        context)
    for i, blk in enumerate(self.hint_blks):
        guid_hint = blk(guid_hint, emb, context)
        guid_hs.append(guid_hint)
    ...
    # output_blocks
    for i, blk in enumerate(self.out_blks):
        skip_h = self.tuners[i](hs.pop() + self.
            scale * guid_hs[::-1][i])
        h = torch.cat([h, skip_h], dim=1)
        h = blk(h, emb, context)
```

information through a process of 64× downsampling of the image, subsequently followed by an upsampling to its original dimensions. The sample images are demonstrated in Fig. 10a.

- **Inpainting.** We employ the mask generation strategy from LaMa [19] for conditional generation on the inpainting task. The sample images are demonstrated in Figs. 10b and 10c.

For all the aforementioned conditions, we utilize the same training dataset (LAION-ART [16]) and hyperparameters across the tasks. The exception is the pose-conditional task, for which we exclusively utilize a subset of images containing human poses, amounting to a total of 162,338 instances. Additionally, for the inpainting task, we follow the common approach of using both masks and cutouts as combined conditional inputs.

B. Ablation studies

B.1. SC-Tuner structure

We ablate our SC-Tuner using the default setting in Tab. 3. It is evident that our method allows for flexible design, including the intermediate dimensions of tuners, the number of utilized skip connection layers, and the selection of submodules.

In Tab. 3a, we retain the dimensions of the skip connection features as the default intermediate dimensions for the tuner. As the dimensions are reduced proportionally,

Table 3. **SC-Tuner ablation** experiments of efficient fine-tuning task on COCO2017. Default settings are marked in gray.

(a) Ablation on downscaling ratio of dimensions.

Ratio	FID	Params	Mem.
×1	13.82	19.68M	29.02G
×5	13.92	3.94M	28.29G
×10	13.99	1.98M	28.06G

(b) Ablation on skip connection (SC) layers.

SC Indexes	FID	Params	Mem.
{0,11}	14.45	3.48M	28.11G
{0,3,6,9,11}	13.96	7.79M	28.56G
{1,2, ..., 12}	13.82	19.68M	29.02G

(c) Ablation on tuner submodules.

Module	FID	Params	Mem.
Linear	13.82	19.68M	29.02G
Conv	13.88	22.13M	28.65G
ResPrefix [7]	14.38	21.64M	30.54G

Table 4. **CSC-Tuner ablation** experiments of controllable generation task on LAION dataset. Default settings are marked in gray.

(a) Ablation on convolution kernel size.

Kernel	FID	Params	Mem.
1	73.18	28.82M	34.78G
3	71.78	99.11M	35.28G

(b) Ablation on skip connection (SC) layers.

SC Indexes	FID	Params	Mem.
{0,3,4,6,7,9,11}	85.42	17.14M	34.48G
{1,2,3, ..., 12}	73.18	28.82M	34.78G

(c) Ablation on tuner submodules.

Module	FID	Params	Mem.
Single Conv	73.18	28.82M	34.78G
Dual Conv	70.54	37.82M	35.31G

there is a corresponding decrease in the number of parameters. Despite this reduction, the decline in memory consumption is not substantial, and the FID [17] fails to show an improvement compared to the default setting. Similarly, in Tab. 3b, a performance degradation is observed when we reduce the number of skip connection layers by intervals. Our SC-Tuner is designed with the flexibility to interchange its internal components, allowing for the use of convolution networks or independent residual networks. As demonstrated in Tab. 3c, even the most elementary components, such as linear layers, can offer certain advantages while maintaining a comparable number of parameters.

B.2. CSC-Tuner structure

We conducted a series of ablation studies based on the modular design of the CSC-Tuner to evaluate the impact of each component on the overall performance.

From a quantitative perspective, in Tab. 4a, we can observe that larger convolution kernels of condition encoder, although increasing the number of parameters, also contribute to a certain reduction in the FID. In Tab. 4b, omitting some of the skip connections results in an increase in the FID. Subsequently, as shown in Tab. 4c, we ablate with altering the internal structure of the tuner by shifting from a single convolution layer to a dual convolution layer with dimension reduction, resulting in improved FID score.

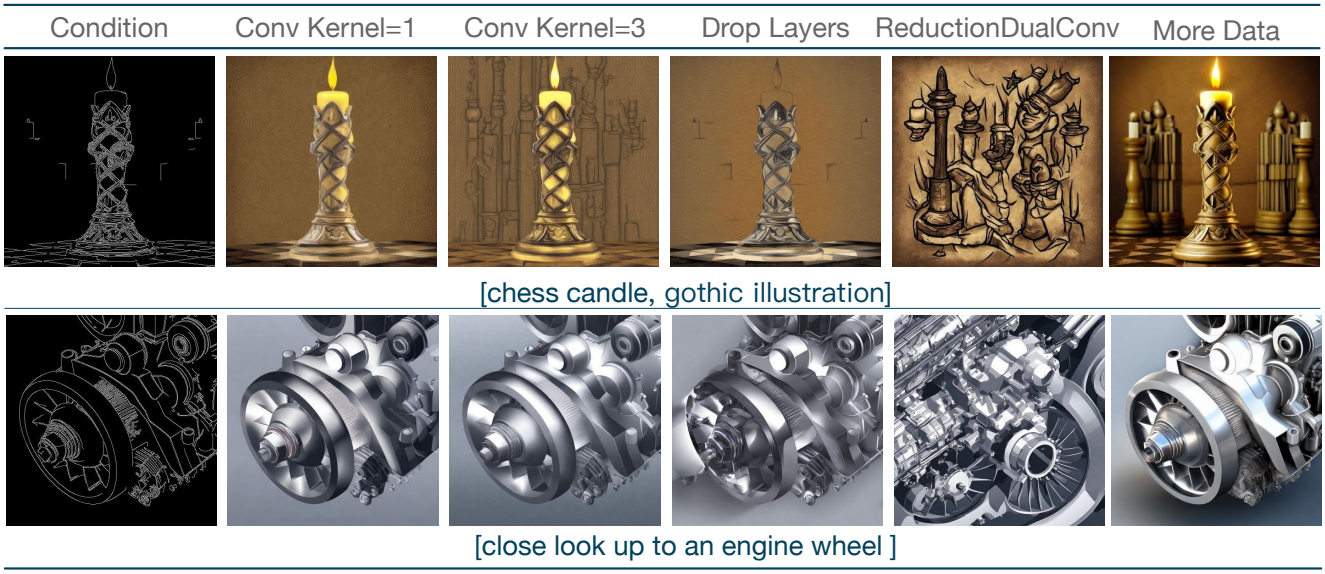


Figure 1. **Qualitative comparison** on various CSC-Tuner structure designs.

From a qualitative perspective, we compared the aforementioned experimental setups and also train on a larger dataset (24M) under the default setting. As evident from Fig. 1, the enlargement of the convolution kernel size expands the receptive field, achieving richer detail in the generated images. Training with more data also benefits from realistic effects. On the other hand, omitting some of the skip connections generally leads to a loss of image content. The dual convolution with dimension reduction exhibits poor control over conditions, underscoring the importance of the channel dimension in generation.

C. Additional results

C.1. Fine-tuning on content images

Generation on customized style is one of the most common fine-tuning downstream tasks and is also widely used within the community. SCEdit also has great performance in fine-tuning with custom content images. In Fig. 2, we showcase the capability for detail generation on live subjects and objects using DreamBooth Datasets [15].

C.2. Generalization across different models

We also conduct experiments on various models in different tasks to demonstrate the generalization ability of our method across different models. The results of the text-to-image tuners based on SD2.1 [2] and SDXL [3] can be seen in Fig. 3, and the results of the conditional controllers based on SDXL are displayed in Fig. 4.

C.3. Performance with minimal parameters

We present the results of a text-to-image fine-tuning task with small parameters in Fig. 5. For the configuration with a parameter count of 19K, we only retain a single layer of SC-Tuner and find that the smaller parameter count requires careful training and extended training time.

C.4. Additional qualitative comparison

In Fig. 6, we present additional qualitative comparison for the controllable generation task, using canny edge maps, depth maps, and semantic segmentation maps as conditions, including comparisons with methods ControlNet [22], T2I-Adapter [13], ControlLoRA [6], and ControlNet-XS [21].

C.5. Additional qualitative results

In Fig. 7, we demonstrate the results of generating images by extracting different conditional information from the same image and using it as control conditions. In Fig. 8, Fig. 9, and Fig. 10, we present additional qualitative results for the controllable generation task, with conditions including canny edge map, depth map, hed boundary map, semantic segmentation map, pose keypoint, color map, outpainting, and inpainting.

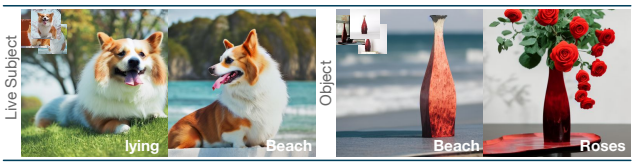


Figure 2. Fine-tuning on live subjects and objects using DreamBooth Datasets (5-6 images per class) with different contexts.

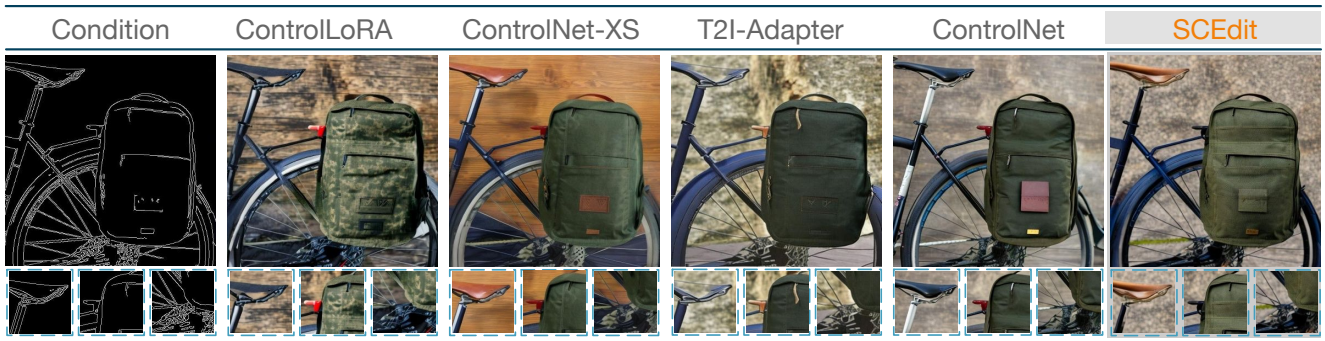
Figure 3. Generation results of various styles on SD2.1 and SDXL.

Figure 4. Generation results of various conditions on SDXL.

Figure 5. Generation results under different parameters in 3D style fine-tuning task with the same prompt.

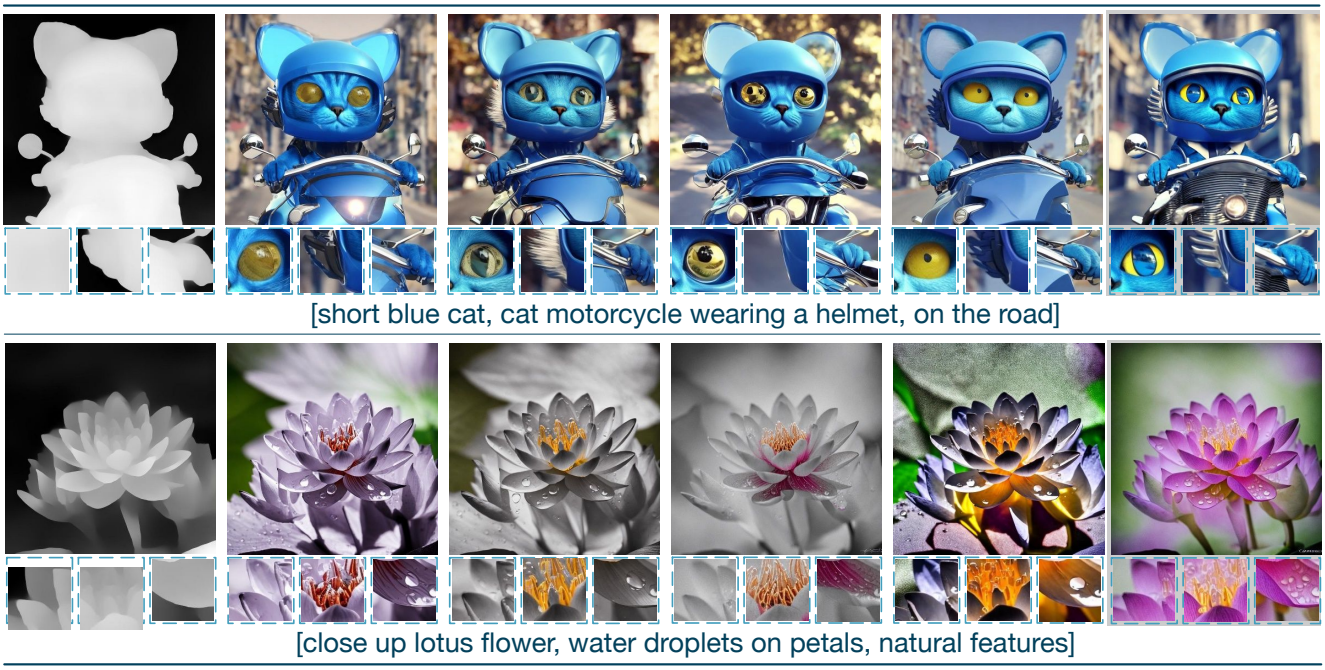
D. Limitations and societal impacts

This work aims to provide users with a method for efficient fine-tuning and controlled synthesis under diverse conditions. The tuning stage based on the pre-trained models while freezing the backbone network, so its transfer ability depends to a large extent on the performance of the upstream model. In addition, it generates results that meet expectations based on the training data and the specified conditional inputs supplied by the users. Conversely, the malicious utilization of high-risk data could potentially lead to the generation of misleading outcomes. This underscores the importance of ethical considerations in the deployment of generative models to prevent the propagation of harmfully biased or false information.



(a) Comparative results of generation conditioned on canny edge map.

(b) Comparative results of generation conditioned on semantic segmentation map.



(c) Comparative results of generation conditioned on depth map.

Figure 6. **Additional qualitative comparison** on the controllable generation of our approach with other strategies conditioned on canny edge maps, semantic segmentation maps, and depth maps. The areas in the boxes are enlarged for detailed comparisons.

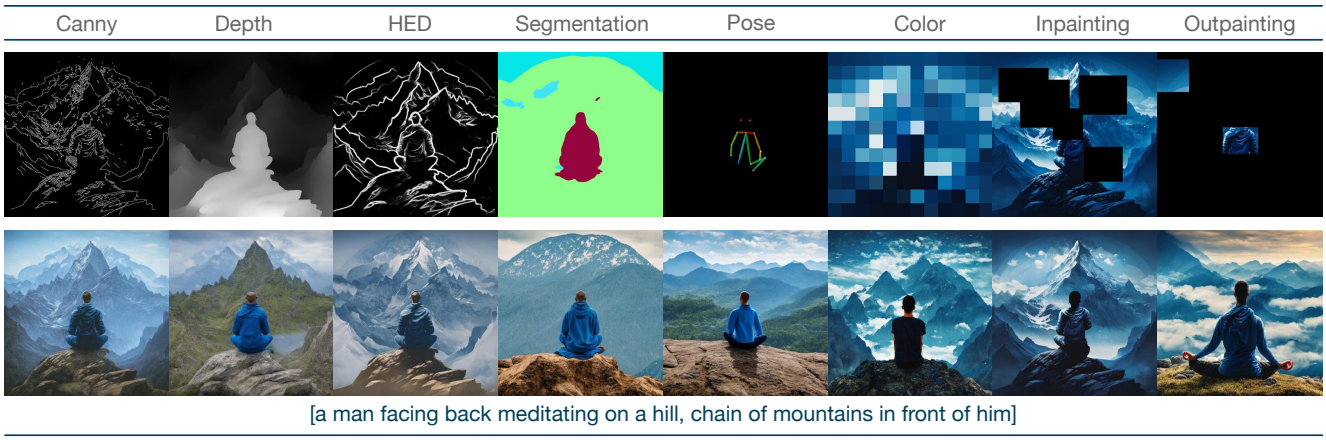


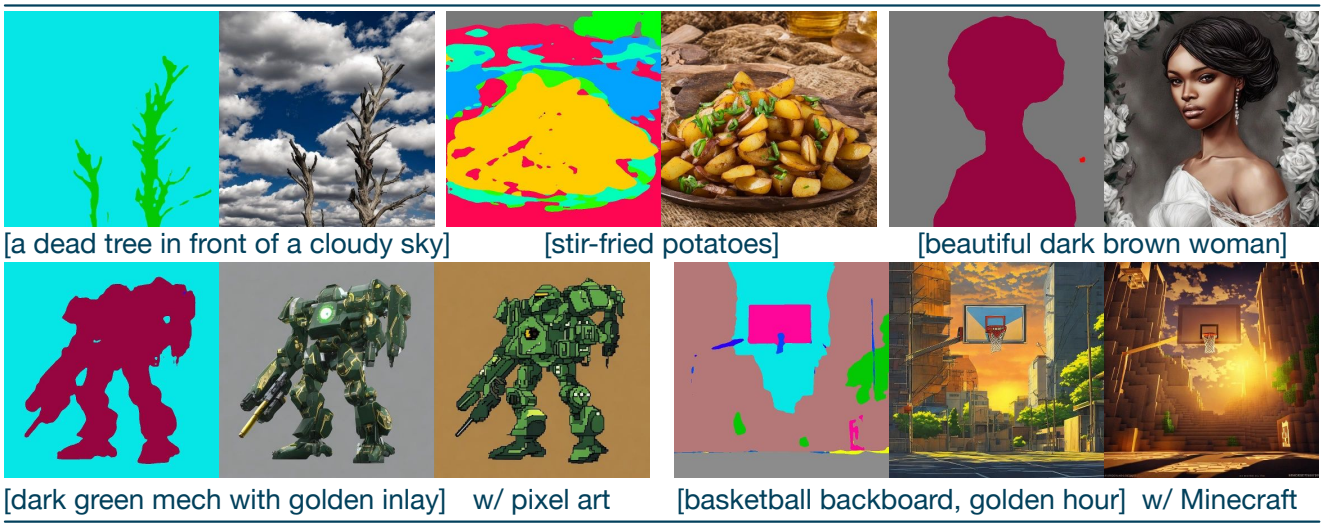
Figure 7. **Additional qualitative results** on controllable generation using the same original image for different conditions.

(a) Generative results conditioned on canny edge map.

(b) Generative results conditioned on depth map.

Figure 8. Additional qualitative results on controllable generation using canny edge map and depth map conditions.

(a) Generative results conditioned on hed boundary map.

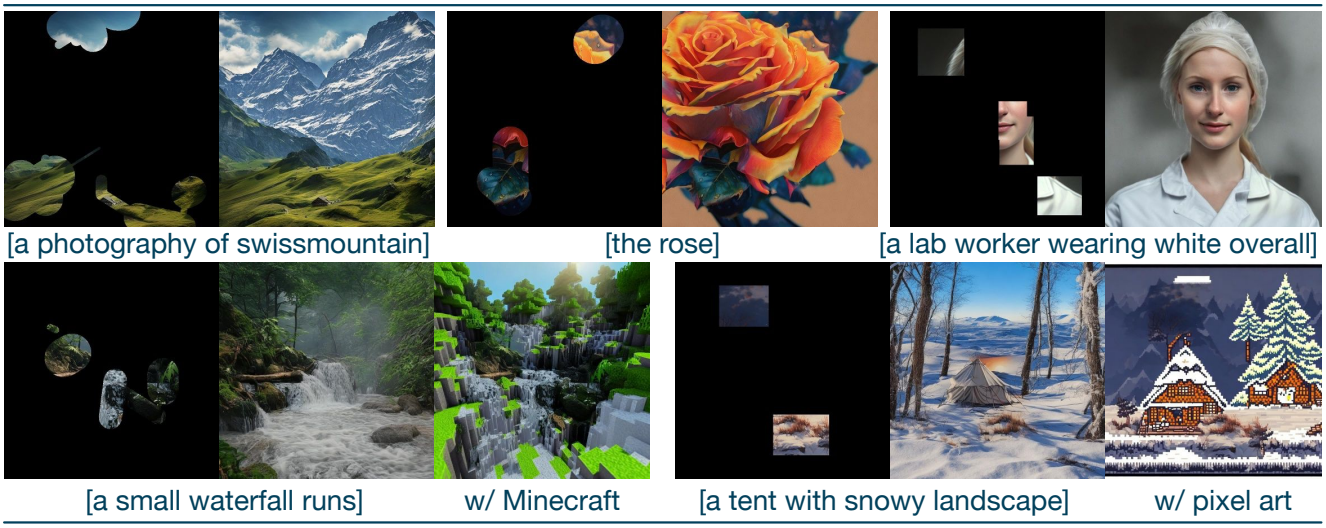


(b) Generative results conditioned on semantic segmentation map.

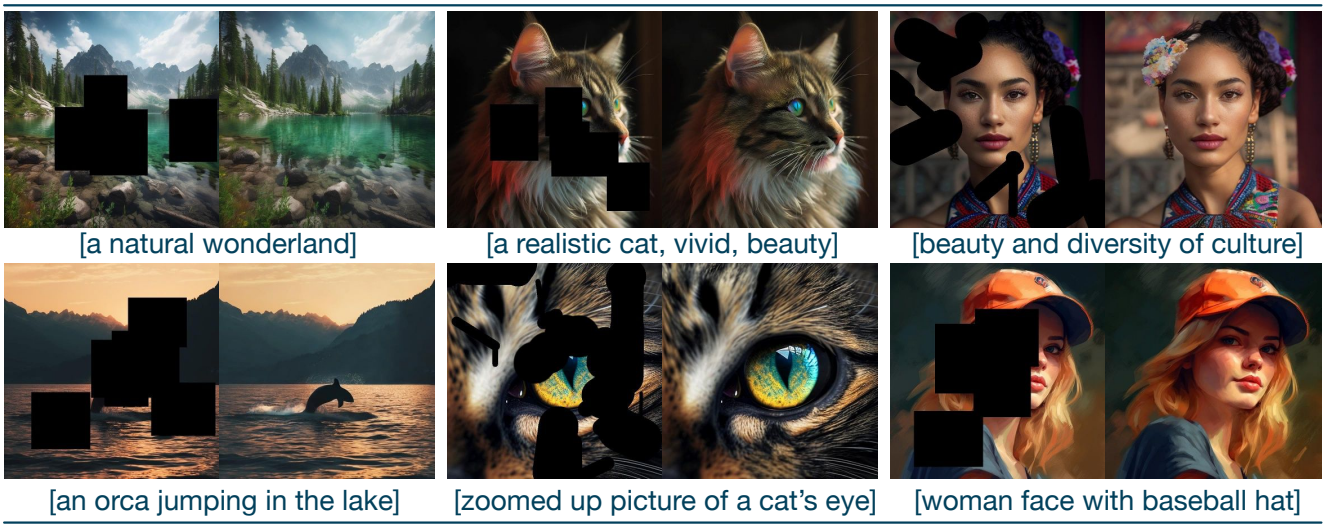
(c) Generative results conditioned on pose keypoint.

Figure 9. **Additional qualitative results** on controllable generation using hed boundary map, semantic segmentation map, and pose keypoint conditions.

(a) Generative results conditioned on color map.



(b) Generative results conditioned on outpainting.



(c) Generative results conditioned on inpainting.

Figure 10. **Additional qualitative results** on controllable generation using color maps, outpainting, and inpainting conditions.

References

[1] Runway AI. Stable Diffusion v1.5 Model Card, https://huggingface.co/runwayml/stable-diffusion-v1-5, 2022. 2

[2] Stability AI. Stable Diffusion v2-1 Model Card, https://huggingface.co/stabilityai/stable-diffusion-2-1, 2022. 2, 4

[3] Stability AI. Stable Diffusion XL Model Card, https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0, 2022. 4

[4] John Canny. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 679–698, 1986. 1

[5] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1):172–186, 2021. 1

[6] Wu Hecong. ControlLoRA: A Lightweight Neural Network To Control Stable Diffusion Spatial Information, https://github.com/HighCWu/ControlLoRA, 2023. 4

[7] Zeyinzi Jiang, Chaojie Mao, Ziyuan Huang, Ao Ma, Yiliang Lv, Yujun Shen, Deli Zhao, and Jingren Zhou. Res-Tuning: A Flexible and Efficient Tuning Paradigm via Unbinding Tuner from Backbone. In *Adv. Neural Inform. Process. Syst.*, 2023. 3

[8] Kunchang Li, Yali Wang, Gao Peng, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. UniFormer: Unified Transformer for Efficient Spatial-Temporal Representation Learning. In *Int. Conf. Learn. Represent.*, 2021. 1

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Eur. Conf. Comput. Vis.*, pages 740–755, 2014. 1, 2

[10] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Int. Conf. Learn. Represent.*, 2018. 2

[11] ModelScope. Customized Style Dataset Card, https://modelscope.cn/datasets/damo/style_custom_dataset/summary, 2023. 1, 2

[12] ModelScope. SWIFT(Scalable lightWeight Infrastructure for Fine-Tuning), https://github.com/modelscope/swift, 2023. 2

[13] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models. *arXiv preprint arXiv:2302.08453*, 2023. 1, 4

[14] Rene Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1623–1637, 2022. 1

[15] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 22500–22510, 2023. 4

[16] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W. Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R. Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Adv. Neural Inform. Process. Syst.*, 2022. 1, 2

[17] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch, https://github.com/mseitzer/pytorch-fid, 2020. 3

[18] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *Int. Conf. Learn. Represent.*, 2021. 2

[19] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-Robust Large Mask Inpainting With Fourier Convolutions. In *IEEE Winter Conf. Appl. Comput. Vis.*, pages 2149–2159, 2022. 2

[20] Saining Xie and Zhuowen Tu. Holistically-Nested Edge Detection. In *Int. Conf. Comput. Vis.*, pages 1395–1403, 2015. 1

[21] Denis Zavadski, Johann-Friedrich Feiden, and Carsten Rother. ControlNet-XS: Designing an Efficient and Effective Architecture for Controlling Text-to-Image Diffusion Models. *arXiv preprint arXiv:2312.06573*, 2023. 4

[22] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *Int. Conf. Comput. Vis.*, pages 3836–3847, 2023. 1, 4

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene Parsing through ADE20K Dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5122–5130, 2017. 1