


ControlNet++: Improving Conditional Controls with Efficient Consistency Feedback

Supplementary Material

Ming Li¹, Taojiannan Yang¹, Huafeng Kuang², Jie Wu²,
Zhaoning Wang¹, Xuefeng Xiao², and Chen Chen¹ 

¹ Center for Research in Computer Vision, University of Central Florida

² ByteDance

1 Overview of Supplementary

The supplementary material is organized into the following sections:

- Section 2: Implementation details for all experiments.
- Section 3: Proof for Eq.(7) in the main paper.
- Section 4: More experiments and analysis.
 - Section 4.1: Effectiveness of conditioning scale of existing methods.
 - Section 4.2: Human evaluation on controllability, text guidance and image quality.
- Section 5: Discussion of broader impact and limitation.
- Section 6: More visualization results.

2 Implementation Details

2.1 Dataset Details

Considering that the training data for ControlNet [8] has not been publicly released, we need to construct our training dataset. In this paper, we adhere to the dataset construction principles of ControlNet [8], which endeavor to select datasets with more accurate conditional conditions wherever possible. Specifically, for the segmentation condition, previous works have provided datasets with accurately labeled segmentation masks [2, 9, 10]. Therefore, we opt to train our model using these accurately labeled datasets following ControlNet [8]. For the Hed, LineArt edge tasks, it is challenging to find datasets with real and accurate annotations. As a result, following ControlNet [8], we train the model using the MultiGen20M dataset [6], which is annotated by models, to address this issue. Regarding the depth task, existing datasets include masks of certain pixels as having unknown depth values, making them incompatible with the current ControlNet pipeline. Therefore, we also adapt the MultiGen20M depth dataset, which is similar to the dataset constructed by ControlNet [8]. In terms of the canny edge task, no human labels are required in the process, so we also adapt the MultiGen20M dataset. We provide details of the datasets in Table 1.

Table 1: Dataset and evaluation details of different conditional controls. \uparrow denotes higher is better, while \downarrow means lower is better.

	Segmentation Mask	Canny Edge	Hed Edge	LineArt Edge	Depth Map
Dataset	ADE20K [9, 10], COCOStuff [2]	MultiGen20M [6]	MultiGen20M [6]	MultiGen20M [6]	MultiGen20M [6]
Training Samples	20,210 & 118,287	2,560,000	2,560,000	2,560,000	2,560,000
Evaluation Samples	2,000 & 5,000	5,000	5,000	5,000	5,000
Evaluation Metric	mIoU \uparrow	F1 Score \uparrow	SSIM \uparrow	SSIM \uparrow	RMSE \downarrow

Table 2: Details of the reward model, evaluation model, and training loss under different conditional controls. ControlNet* denotes we use the same model to extract conditions as ControlNet [8]

	Seg. Mask	Depth Edge	Canny Edge	Hed Edge	LineArt Edge
Reward Model (RM)	UperNet-R50	DPT-Hybrid	Kornia Canny	ControlNet*	ControlNet*
RM Performance	ADE20K(mIoU): 42.05	NYU(AbsRel): 8.69	-	-	-
Evaluation Model (EM)	Mask2Former	DPT-Large	Kornia Canny	ControlNet*	ControlNet*
EM Performance	ADE20K(mIoU): 56.01	NYU(AbsRel): 8.32	-	-	-
Consistency Loss	CrossEntropy Loss	MSE Loss	MSE Loss	MSE Loss	MSE Loss
Loss Weight λ	0.5	0.5	1.0	1.0	10

2.2 Reward Model and Evaluation Details

In general, we deliberately choose slightly weaker models as the reward model and opt for stronger models for evaluation. This practice not only ensures the fairness of the evaluation but also helps to determine whether performance improvements result from alignment with the reward model’s preferences or from a genuine enhancement in controllability. While such an approach is feasible for some tasks (Segmentation, Depth), it becomes challenging to implement for others (Hed, Canny, LineArt Edge) due to the difficulty in finding two distinct reward models. In such cases, we use the same model as both the reward model and the evaluation model. We utilize standard evaluation schemes from their respective research fields to evaluate the input conditions and extracted conditions from the generated images, as demonstrated in Section 4.1 of the main paper. We use the same Hed edge detection model and LineArt edge detection model as ControlNet [8]. We provide details of reward models and evaluation in Table 2.

2.3 Training Details

The loss weight λ for reward consistency loss is different for each condition. Specifically, λ is 0.5, 0.5, 1.0, 1.0, and 10 for segmentation mask, depth, hed edge, canny edge, and LineArt edge condition, respectively. For all experiments, we first fine-tune the pre-trained ControlNet until convergence using a batch size of 256 and a learning rate of $1e-5$. We then employ the same batch size and learning rate for 10k iterations reward fine-tuning. To this end, the valid training samples for reward fine-tuning is $256 \times 10,000 = 2,560,000$. We set threshold $t_{\text{thre}} = 200$ of Eq.8 in the main paper for all experiments. Diverging from existing methods that use OpenCV’s [1] implementation of the canny algorithm, we have adopted Kornia’s [7] implementation to make it differentiable. Our codebase is based on the implementation in HuggingFace’s Diffusers [5], and we do not use classifier-free guidance during the reward fine-tuning process following diffusers.

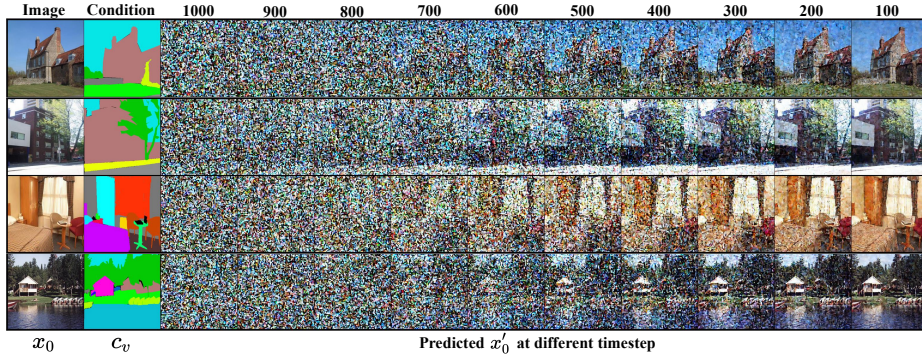


Fig. 1: Illustration of predicted image x'_0 at different timesteps t . A small timestep t (i.e., small noise ϵ_t) leads to more precise estimation $x'_0 \approx x_0$.

3 Proof of Equation 7 in the Main Paper

The diffusion models define a Markovian chain of diffusion forward process $q(x_t|x_0)$ by gradually adding noise to input data x_0 :

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, I), \quad (1)$$

at any timestep t we can use the predicted $\epsilon(x'_t, c_v, c_t, t - 1)$ to estimate the real noise ϵ in Eq. 1, and the above equation can be transformed through straightforward algebraic manipulation to the following form:

$$\begin{aligned} x_t &\approx \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(x'_t, c_v, c_t, t - 1), \\ x_0 &\approx x'_0 = \frac{x'_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(x'_t, c_v, c_t, t - 1)}{\sqrt{\bar{\alpha}_t}}. \end{aligned} \quad (2)$$

To this end, we can obtain the predicted original image x'_0 at any denoising timestep t and use it as the input for reward consistency loss. However, previous work demonstrates that this approximation only yields a smaller error when the time step t is relatively small [3]. Here we find similar results as shown in Figure 1, which illustrates the predicted x'_0 is significantly different at different timesteps. We kindly encourage readers to refer to Section 4.3 and Figure 5 in the DDPM [3] paper for more experimental results.

4 More Experiments

In this section, we provide additional supplements to the experiments discussed in the main paper, including human evaluation on generated data samples on the Segmentation Mask condition in Sec. 4.2, analysis on conditioning scale of existing methods such as ControlNet [8] and T2I-Adapter [4] in Sec. 4.1.

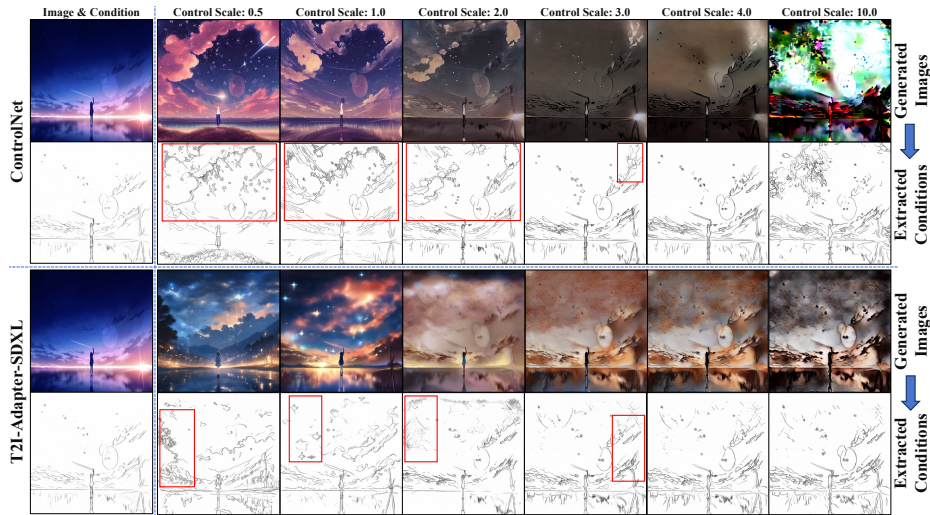


Fig. 2: Naively increasing the weight of image condition embedding compared to text condition embedding in existing methods (i.e., ControlNet and T2I-Adapter) **cannot** improve controllability while ensuring image quality. The red boxes in the figures highlight areas where the generated image is inconsistent with the input conditions. Please note that we employ the same line detection model to extract conditions from images.

4.1 Effectiveness of Conditioning Scale

To simultaneously achieve control based on text prompts and image conditions, existing controllable generation methods perform an addition operation between the image condition features and the text embedding features. The strength of different conditions can be adjusted through a weight value. Hence, an obvious question arises: can better controllability be achieved by increasing the weight of the image condition features? To answer this question, we conduct experiments under different control scales (The weight of image condition feature) in Figure 2. It demonstrates that naively increasing the control ratio of image conditions does not enhance controllability and may lead to severe image distortion.

4.2 Human Evaluation

Following ControlNet, we use a single condition for human evaluation. We ask 20 users (12 in ControlNet paper) to select the best image based on three distinct criteria as shown in Table 3. Our ControlNet++ offers better controllability without sacrificing image quality or text guidance.

Table 3: Win rate on ADE20K validation dataset (Segmentation).

<i>20 annotators in total</i>	Ours	ControlNet	T2I-Adapter	UniControl
Image-Mask Alignment	76.8%	16.7%	2.0%	4.5%
Image Quality	26.1%	25.8 %	23.6%	24.5 %
Image-Text Alignment	25.3%	25.1%	24.9%	24.7%

5 Broader Impact and Limitation

In this paper, we use visual discriminative models to evaluate and improve the controllability of text-to-image models. However, we also realize that this work is still insufficient and discuss the following issues:

Conditions Expansion: While we have achieved notable improvements under six control conditions, our future work aims to broaden the scope by incorporating additional control conditions such as Human Pose and Scribbles. Ultimately, our objective is to control everything.

Beyond Controllability: While our current focus lies predominantly on controllability, we acknowledge the significance of quality and aesthetic appeal in the generated outputs. To address this, we plan to leverage human feedback to annotate controllability images. Subsequently, we will optimize the controllability model to simultaneously enhance both controllability and aesthetics.

Joint Optimization: To further enhance the overall performance, we intend to employ a larger set of controllable images for joint optimization of the control network and reward model. This holistic approach would facilitate their co-evolution, leading to further improvements in the final generated outputs. Through our research, we aspire to provide insightful contributions to controllability in text-to-image diffusion models. We hope that our work inspires and encourages more researchers to delve into this fascinating area.

Joint Optimization: To further enhance the overall performance, we intend to employ a larger set of controllable images for joint optimization of the control network and reward model. This holistic approach would facilitate their co-evolution, leading to further improvements in the final generated outputs. Through our research, we aspire to provide insightful contributions to controllability in text-to-image diffusion models. We hope that our work inspires and encourages more researchers to delve into this fascinating area.

Discussion on the necessity of controllability: Controllability is important since it allows users to modify image conditions to achieve more flexible and accurate generation. Take LineArt Edge as an example: (1) Freely generating in foreground may change the appearance (*e.g.*, adding a beard for women) that we usually do not expect. (2) Freely generating in background will damage some applications (*e.g.*, blur background). (3) Global free generating may destroy the overall artistic effect of the input image, such as lighting, composition, contrast, etc. Furthermore, we show in Fig.5 of the main paper that more controllable diffusion can in return improve the performance of discriminative models. Beyond image generation, the controllable conditional generation can also be combined with ID preserving methods to perform controllable image editing.

6 More Visualization

More visualization results across different conditional controls for our image generation are shown in Figures 3,4,5,6,7.

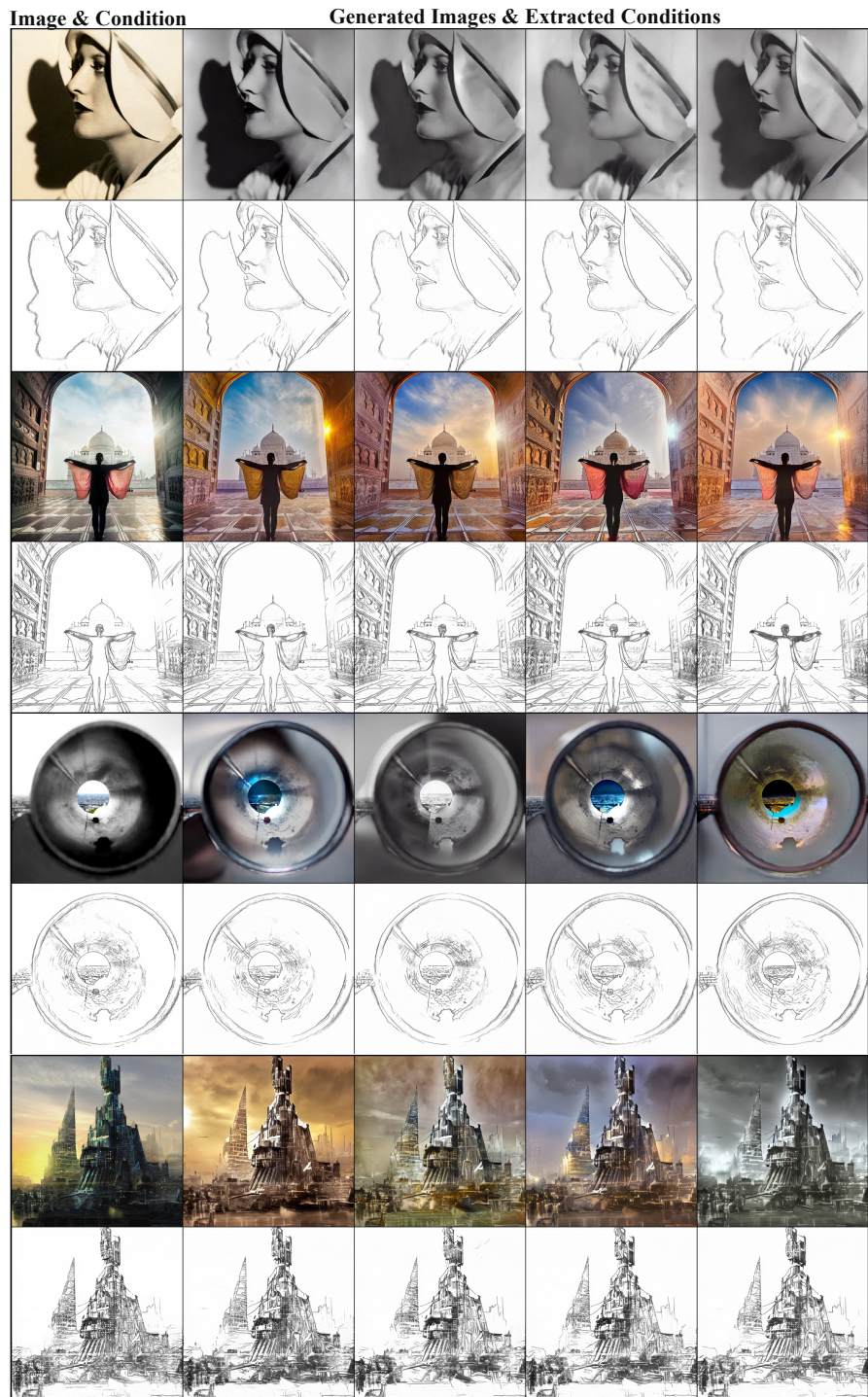


Fig. 3: More visualization results of our ControlNet++ (LineArt Edge)



Fig. 4: More visualization results of our ControlNet++ (Depth Map)

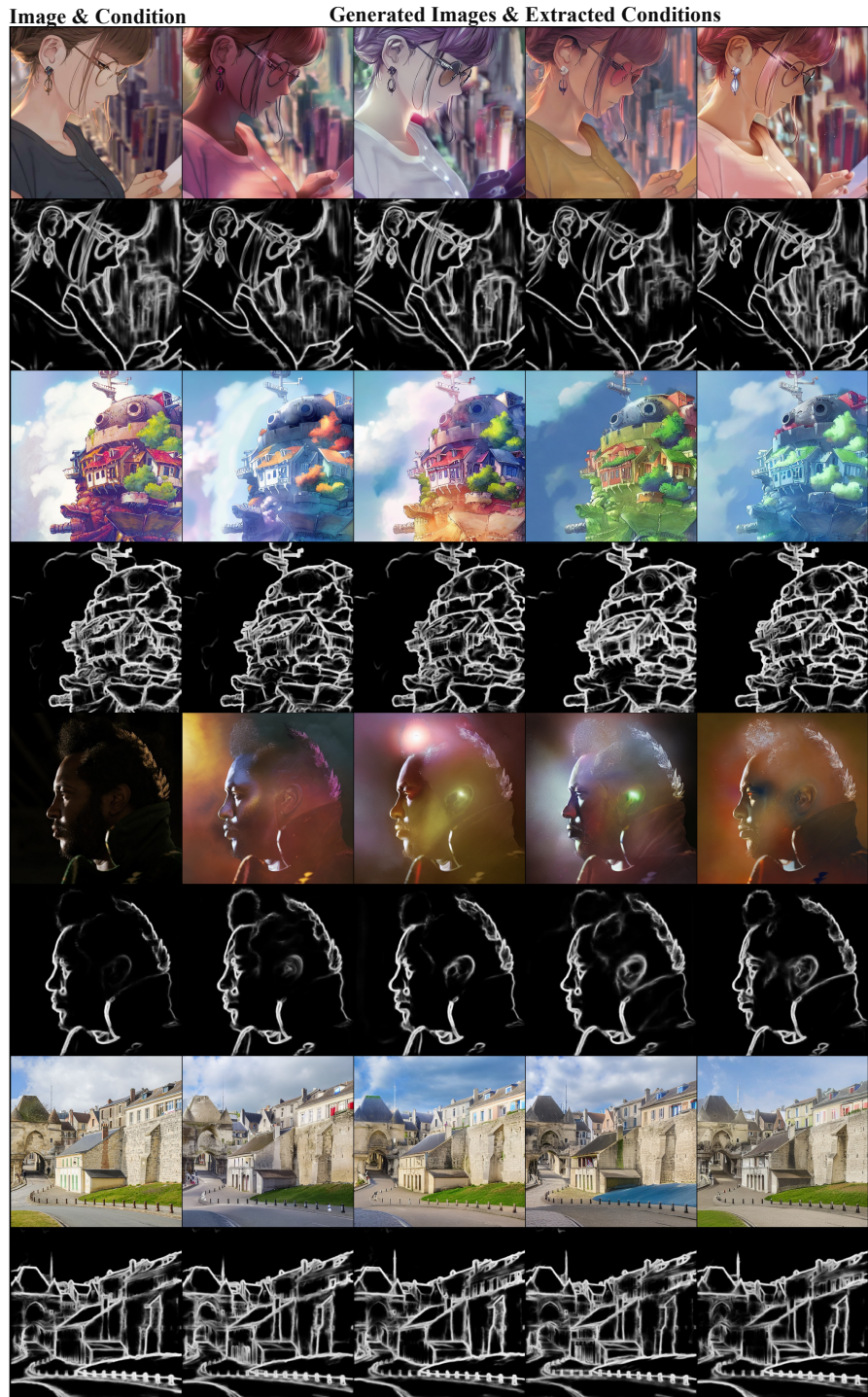


Fig. 5: More visualization results of our ControlNet++ (Hed Edge)

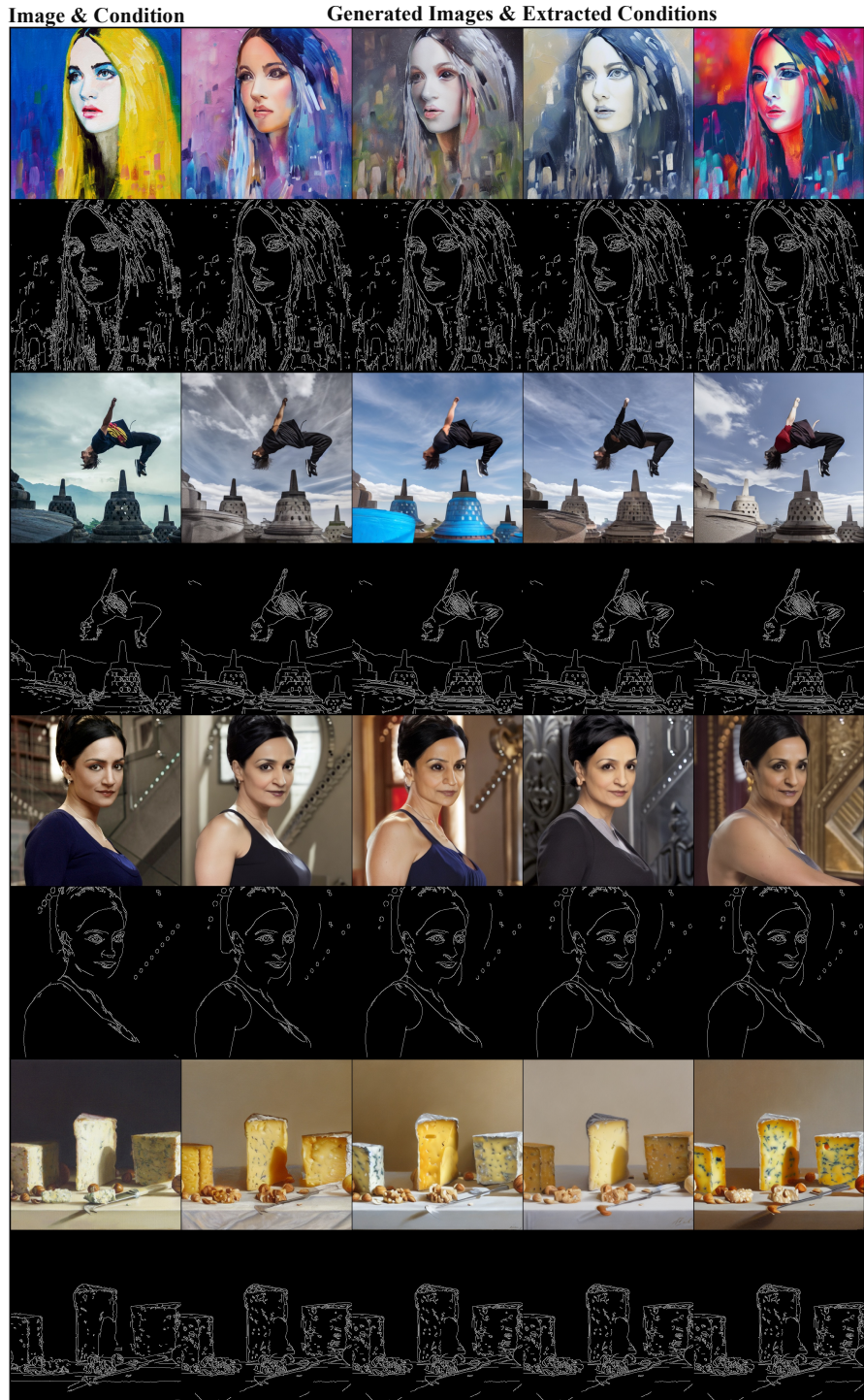


Fig. 6: More visualization results of our ControlNet++ (Canny Edge)



Fig. 7: More visualization results of our ControlNet++ (Segmentation Mask)

References

1. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
2. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018)
3. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
4. Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., Qie, X.: T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. arXiv preprint arXiv:2302.08453 (2023)
5. von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Wolf, T.: Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers> (2022)
6. Qin, C., Zhang, S., Yu, N., Feng, Y., Yang, X., Zhou, Y., Wang, H., Niebles, J.C., Xiong, C., Savarese, S., et al.: Unicontrol: A unified diffusion model for controllable visual generation in the wild. NeurIPS (2023)
7. Riba, E., Mishkin, D., Ponsa, D., Rublee, E., Bradski, G.: Kornia: an open source differentiable computer vision library for pytorch. In: CVPR (2020)
8. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)
9. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralla, A.: Scene parsing through ade20k dataset. In: CVPR (2017)
10. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralla, A.: Semantic understanding of scenes through the ade20k dataset. IJCV (2019)