Adding Conditional Control to Text-to-Image Diffusion Models

- Supplementary Material -

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala
Stanford University
{lvmin, anyirao, maneesh}@cs.stanford.edu

This document includes additional results, implementation parameters, and experimental details for the main paper.

Contents

1 Implementation Detail

1.1 Training Parameters

We have trained several ControlNet models with different image-based conditions that can control large pretrained diffusion models in different ways. An overview is listed in Table 1.

Canny Edge We use a Canny edge detector [2] (with random thresholds) to obtain 3M edge-image-caption pairs from the internet. (We use the utils of LAION [15], and image captions are obtained directly from internet websites, but the actual image sources are constructed by us and differ from LAION to reduce problems like copyright and duplication.) The model is trained using 600 GPU-hours with NVIDIA A100 80GB GPUs. The base model is Stable Diffusion V1.5 (SD V1.5). The batch size is 32. The learning rate is 1e-5. We do not use ema (SD's implementation of exponential moving average) weights [16].

Canny Edge (Scale Test) We use images with the highest resolutions of the above Canny edge dataset and sample several subsets with 1K, 10K, 50K, 500K samples. For example, the 200K subset is the images with the top 200K highest resolutions in the original dataset. We use the same experimental settings as used above to test the effect of dataset scale. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

Hough Line We use a learning-based deep Hough transform [5] to detect straight lines from Places2 [23], and then use BLIP [8] to generate captions. We obtain 600K edge-image-caption pairs. We use the above Canny model as a starting checkpoint and train the model with 150 GPU-hours with NVIDIA A100 80GB GPUs. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

HED Boundary We use HED boundary detection [22] to obtain 3M edge-image-caption pairs from internet (the same source of the Canny dataset). The model is trained with 300 GPU-hours with NVIDIA A100 80GB GPUs. The base model is Stable Diffusion V1.5. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

User Scribble We synthesize human scribbles from images using a combination of HED boundary detection [22] and a set of strong data augmentations (random thresholds, randomly masking out a random percentage of scribbles, random morphological transformations, and random non-maximum suppression). We obtain 500K scribble-image-caption pairs from internet. (Captions are obtained directly from internet websites.) The model is trained with 300 GPU-hours with NVIDIA A100 80GB GPUs. The base model is Stable Diffusion V1.5. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

Human Pose (Openpifpaf) We use learning-based pose estimation method [7] to "find" humans from internet using a simple rule: an image with human must have at least 30% of the key points of the whole body detected. We obtain 80K pose-image-caption pairs. (Captions are obtained directly from internet websites.) Note that we directly use visualized pose images with human skeletons as training condition. The model is trained using 400 GPU-hours on a single NVIDIA RTX 3090TI GPU. The base model is Stable Diffusion V2.1. The batch size is 18 (physical batch size is 3, with 6× gradient accumulation). The learning rate is 1e-5. We do not use ema weights.

Human Pose (Openpose) We use learning-based pose estimation method [3] to find humans from internet using the same rule in the above Openpifpaf setting. We obtain 200K pose-image-caption pairs. (Captions are obtained directly from internet websites.) Note that we directly use visualized pose images with human skeletons as training condition. The model is trained using 300 GPU-hours with NVIDIA A100 80GB GPUs. This model is trained with Stable Diffusion V1.5. Other settings are the same as the above Openpifpaf. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

Semantic Segmentation (COCO) The COCO-Stuff dataset [1] captioned by BLIP [8]. We obtain 164K segmentation-image-caption pairs. The model is trained with 400 GPU-hours on a single

Table 1: Training setting. And for all experiments, the ema weights are not used.

Conditions	Training samples	Training GPU Type and Hours	Base model
Canny Edge	3M Internet	∼600 A100	Stable Diffusion V1.5
Hough Line	600K Places2	∼150 A100	Resumed from the Canny model
HED Boundary	3M Internet	∼300 A100	Stable Diffusion V1.5
User Sketching	500K Internet	∼150 A100	Resumed from the Canny model
Human Pose (Openpifpaf)	200K Openpifpaf	∼400 3090TI	Stable Diffusion V2.1
Human Pose (Openpose)	200K Openpose	∼300 A100	Stable Diffusion V1.5
Semantic Mask (COCO)	164K COCO	∼400 3090TI	Stable Diffusion V1.5
Semantic Mask (ADE20K)	20K ADE20K	∼200 A100	Stable Diffusion V1.5
Depth	3M Internet	∼500 A100	Stable Diffusion V1.5
Normal Maps	25K DIODE	∼100 A100	Stable Diffusion V1.5
Cartoon Line Drawing	1M Internet	∼300 A100	Waifu Diffusion

NVIDIA RTX 3090TI GPU. The base model is Stable Diffusion V1.5. The batch size is 18 (physical batch size is 3, with 6× gradient accumulation). The learning rate is 1e-5. We do not use ema weights.

Semantic Segmentation (ADE20K) The ADE20K dataset [24] captioned by BLIP [8]. We obtain 20K segmentation-image-caption pairs. The model is trained with 200 GPU-hours on NVIDIA A100 80GB GPUs. The base model is Stable Diffusion V1.5. The batch size is 256 (4x gradient accumulation). The learning rate is 1e-5. We do not use ema weights.

Depth (large-scale) We use the Midas [12] and BLIP [8] to obtain 3M depth-image-caption pairs from internet. (Captions are obtained directly from internet websites.) The model is trained with 500 GPU-hours using NVIDIA A100 80GB GPUs. The base model is Stable Diffusion V1.5. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

Depth (small-scale) We rank the image resolutions (using the above method for Canny dataset) of the above depth dataset to sample a subset of 200K pairs. This set is used in experiments aimed at finding the minimal required dataset size to train the model. We use Stable Diffusion V1.5 and Stable Diffusion V2.1 to train two different models for this test.

Normal Maps We take RGB images and normal maps from the DIODE dataset [18] and generate captions for RGB images by BLIP [8] to obtain 25,452 normal-image-caption pairs. The model is trained using 100 GPU-hours on NVIDIA A100 80GB GPUs. The base model is Stable Diffusion V1.5. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

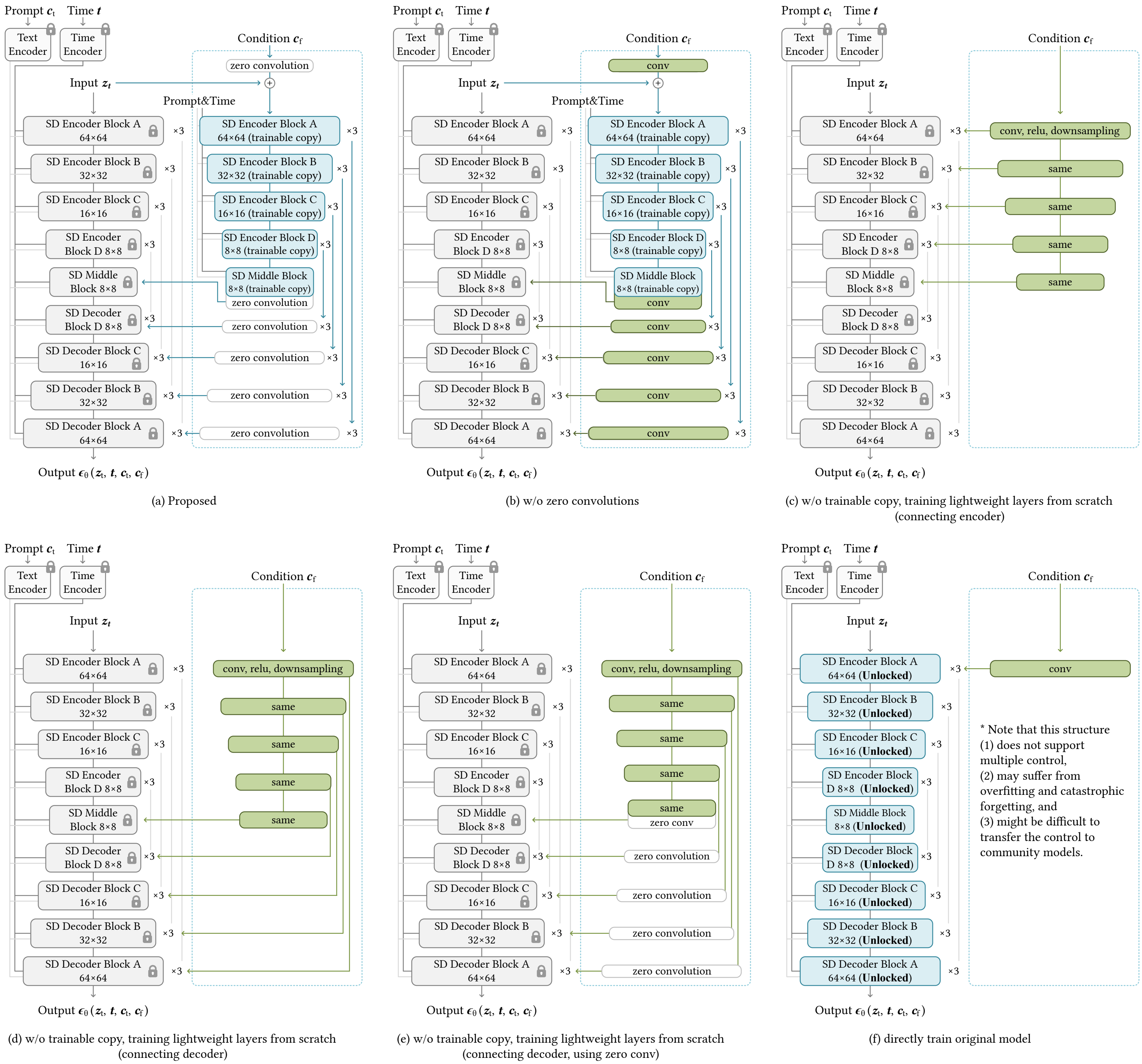
Normal Maps (extended) We use Midas [12] to compute depth map and then compute normal-from-distance to produce "coarse" normal maps. We use the above Normal model as a starting checkpoint and train the model with 200 GPU-hours using NVIDIA A100 80GB GPUs. The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

Cartoon Line Drawing We use a cartoon line drawing extracting method [21] to extract line drawings from cartoon illustration from internet. By sorting the cartoon images with popularity and captioning them by jointing Danbooru Tags, we obtain the top 1M lineart-cartoon-caption pairs. (Danbooru Tags are downloaded directly from internet.) The model is trained using 300 GPU-hours with NVIDIA A100 80GB GPUs. The base model is Waifu Diffusion [9] (a community-developed variation of stable diffusion). The batch size is 32. The learning rate is 1e-5. We do not use ema weights.

1.2 Inference Parameters

Unless otherwise clarified, we use 7.0 as a default cfg scale. We use DDIM as the sampler, and use 20 steps to sample each image.

We use the four prompt settings for experiments:


Prompt c_t
Time t
Text Encoder
Time Encoder
Condition c_f
zero convolution
Input z_t
Prompt&Time
SD Encoder Block A 64×64
SD Encoder Block A 64×64 (trainable copy)
SD Encoder Block B 32×32
SD Encoder Block B 32×32 (trainable copy)
SD Encoder Block C 16×16
SD Encoder Block C 16×16 (trainable copy)
SD Encoder Block D 8×8
SD Encoder Block D 8×8 (trainable copy)
SD Middle Block 8×8
SD Middle Block 8×8 (trainable copy)
SD Decoder Block D 8×8
SD Decoder Block C 16×16
SD Decoder Block B 32×32
SD Decoder Block A 64×64
×3
Output $\epsilon_\theta(z_t, t, c_t, c_f)$
conv
conv, relu, downsampling
same
zero conv
SD Encoder Block A 64×64 (Unlocked)
SD Encoder Block B 32×32 (Unlocked)
SD Encoder Block C 16×16 (Unlocked)
SD Encoder Block D 8×8 (Unlocked)
SD Middle Block 8×8 (Unlocked)
SD Decoder Block D 8×8 (Unlocked)
SD Decoder Block C 16×16 (Unlocked)
SD Decoder Block B 32×32 (Unlocked)
SD Decoder Block A 64×64 (Unlocked)

(a) Proposed

(b) w/o zero convolutions

(c) w/o trainable copy, training lightweight layers from scratch (connecting encoder)

(d) w/o trainable copy, training lightweight layers from scratch (connecting decoder)

(e) w/o trainable copy, training lightweight layers from scratch (connecting decoder, using zero conv)

* Note that this structure (1) does not support multiple control, (2) may suffer from overfitting and catastrophic forgetting, and (3) might be difficult to transfer the control to community models.

(f) directly train original model

Figure 1: Architectures in our ablative study. The (a), (b), and (c) are consistent to the ablative study in the main paper, while the (d), (e), and (f) are extended experiments.

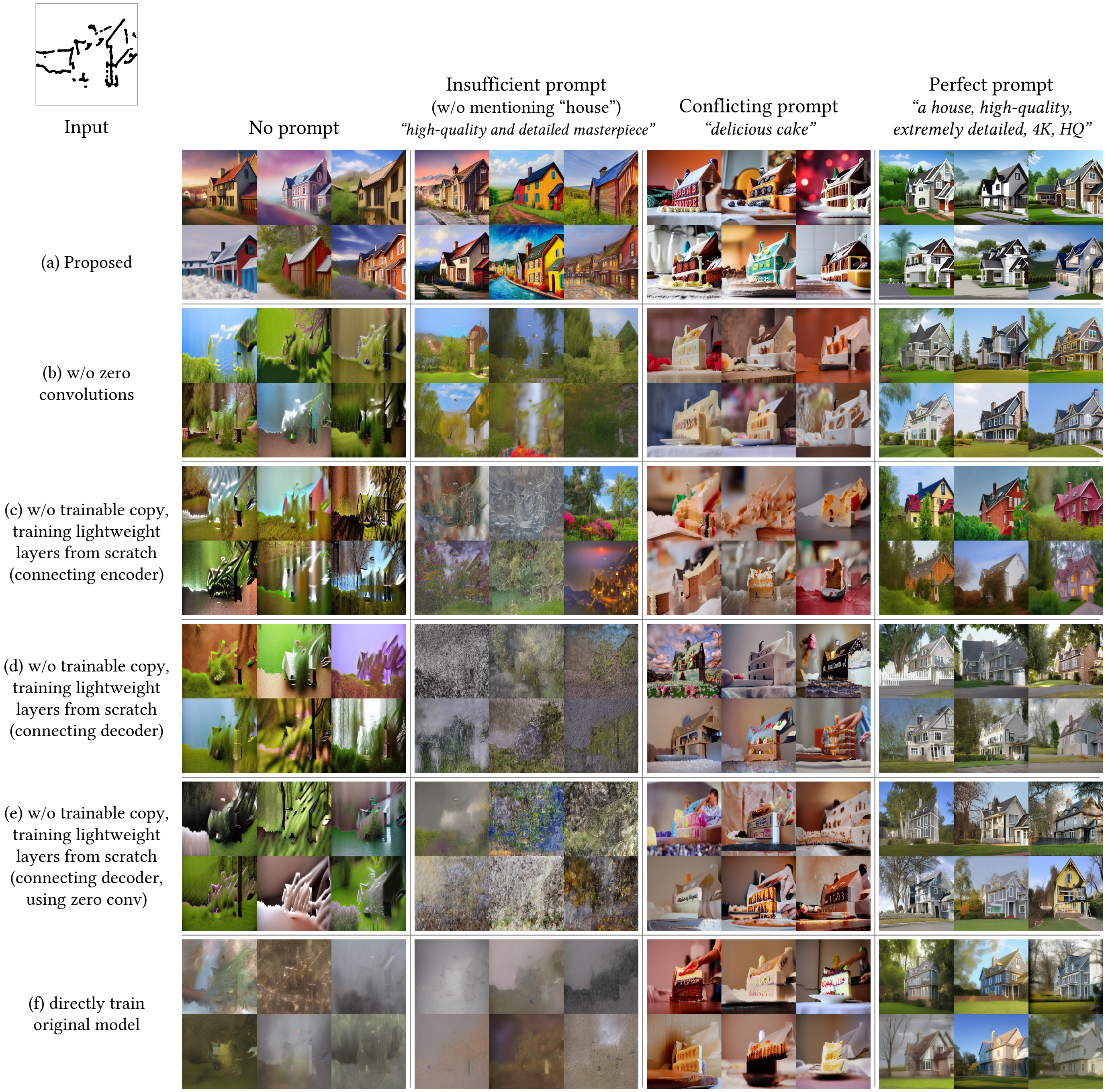

Input
No prompt
Insufficient prompt
(w/o mentioning "house")
"high-quality and detailed masterpiece"
Conflicting prompt
"delicious cake"
Perfect prompt
"a house, high-quality, extremely detailed, 4K, HQ"
(a) Proposed
(b) w/o zero convolutions
(c) w/o trainable copy, training lightweight layers from scratch (connecting encoder)
(d) w/o trainable copy, training lightweight layers from scratch (connecting decoder)
(e) w/o trainable copy, training lightweight layers from scratch (connecting decoder, using zero conv)
(f) directly train original model

Figure 2: Ablative study results. We compare different architectures with different prompt settings. The tested model is the scribble model and the input scribble is on the top-left.

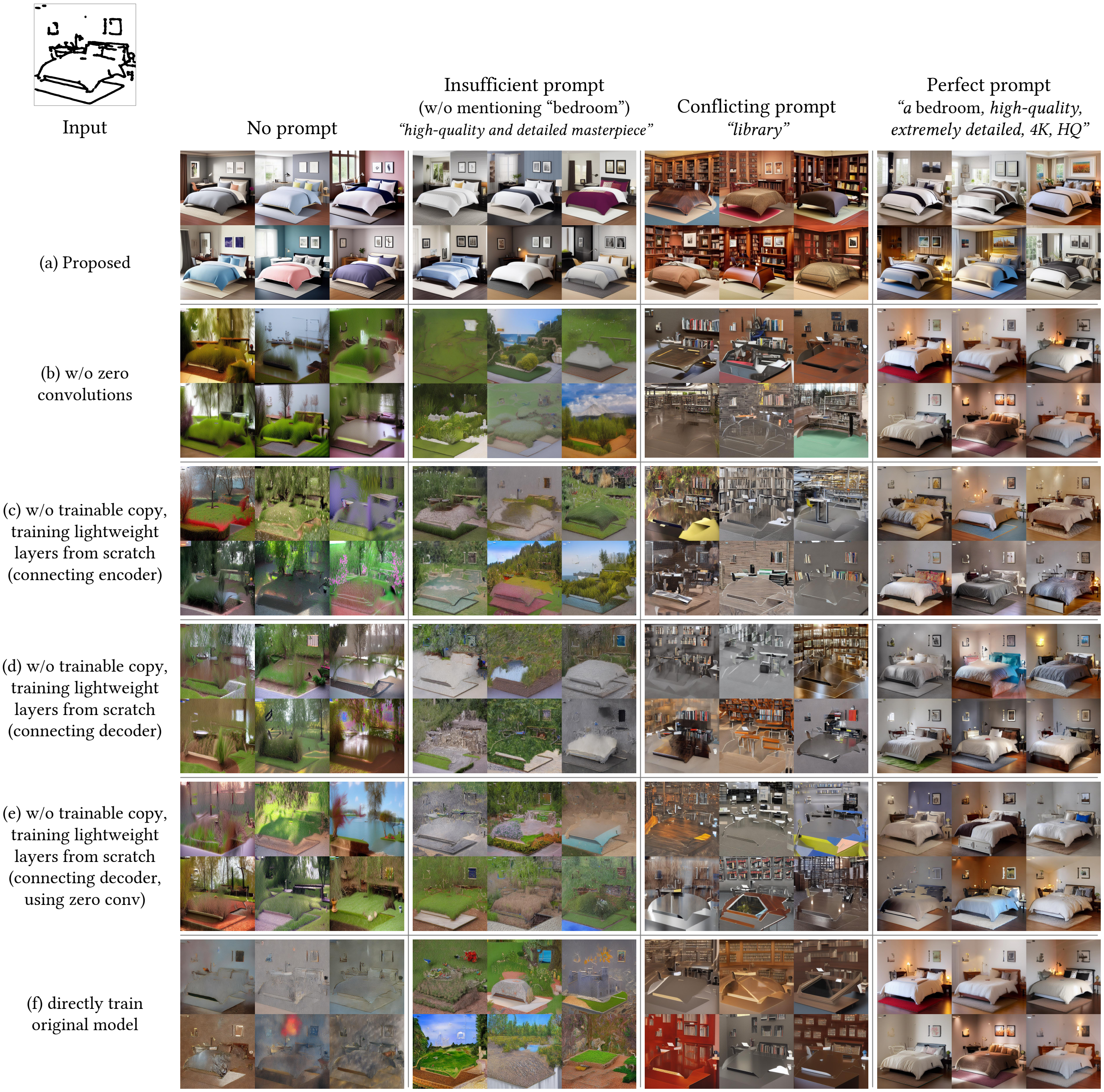

Input
No prompt
Insufficient prompt
(w/o mentioning "bedroom")
"high-quality and detailed masterpiece"
Conflicting prompt
"library"
Perfect prompt
"a bedroom, *high-quality,*
extremely detailed, 4K, HQ"
(a) Proposed
(b) w/o zero convolutions
(c) w/o trainable copy, training lightweight layers from scratch (connecting encoder)
(d) w/o trainable copy, training lightweight layers from scratch (connecting decoder)
(e) w/o trainable copy, training lightweight layers from scratch (connecting decoder, using zero conv)
(f) directly train original model

Figure 3: Ablative study results. We compare different architectures with different prompt settings. The tested model is the scribble model and the input scribble is on the top-left.

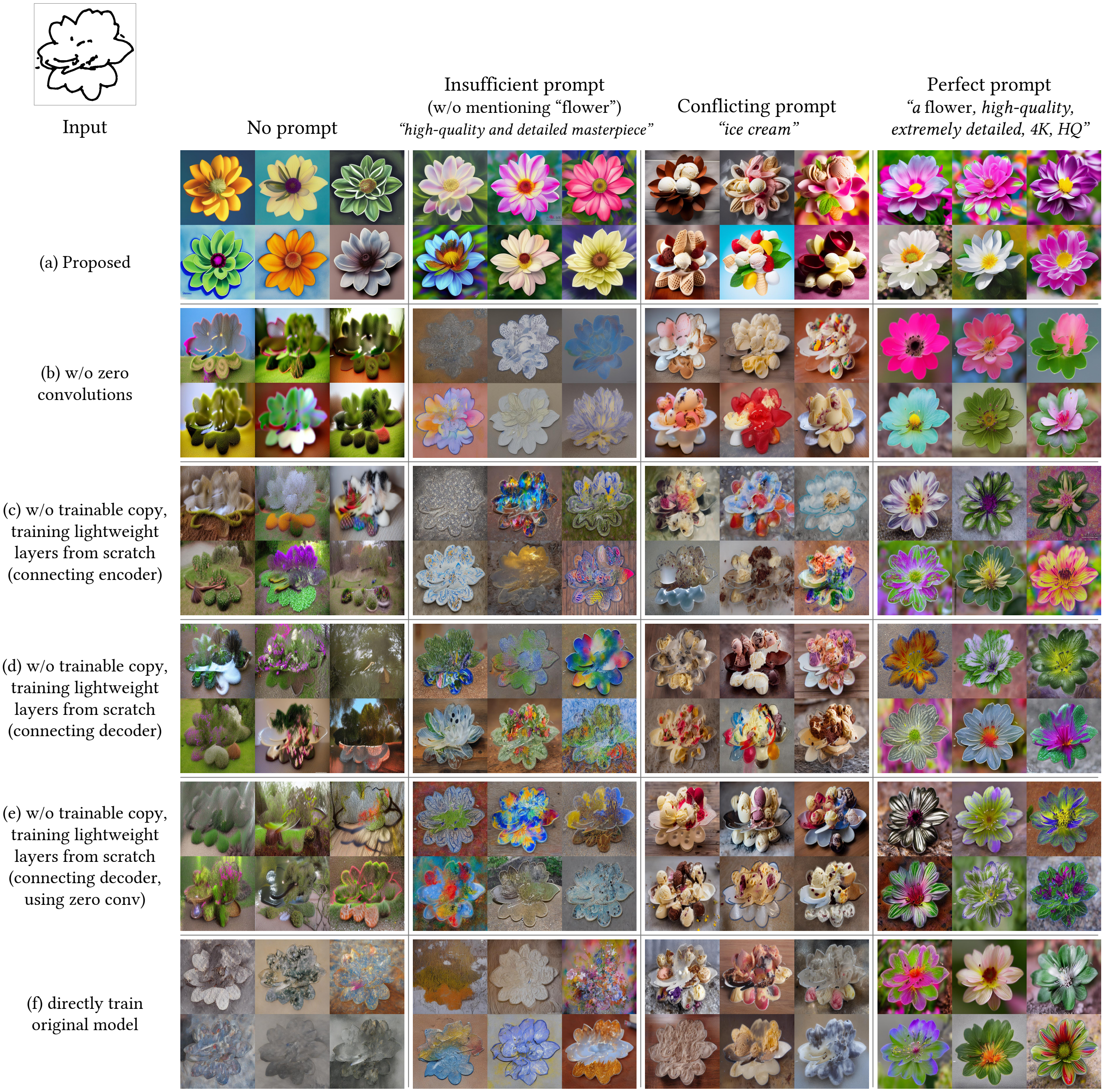
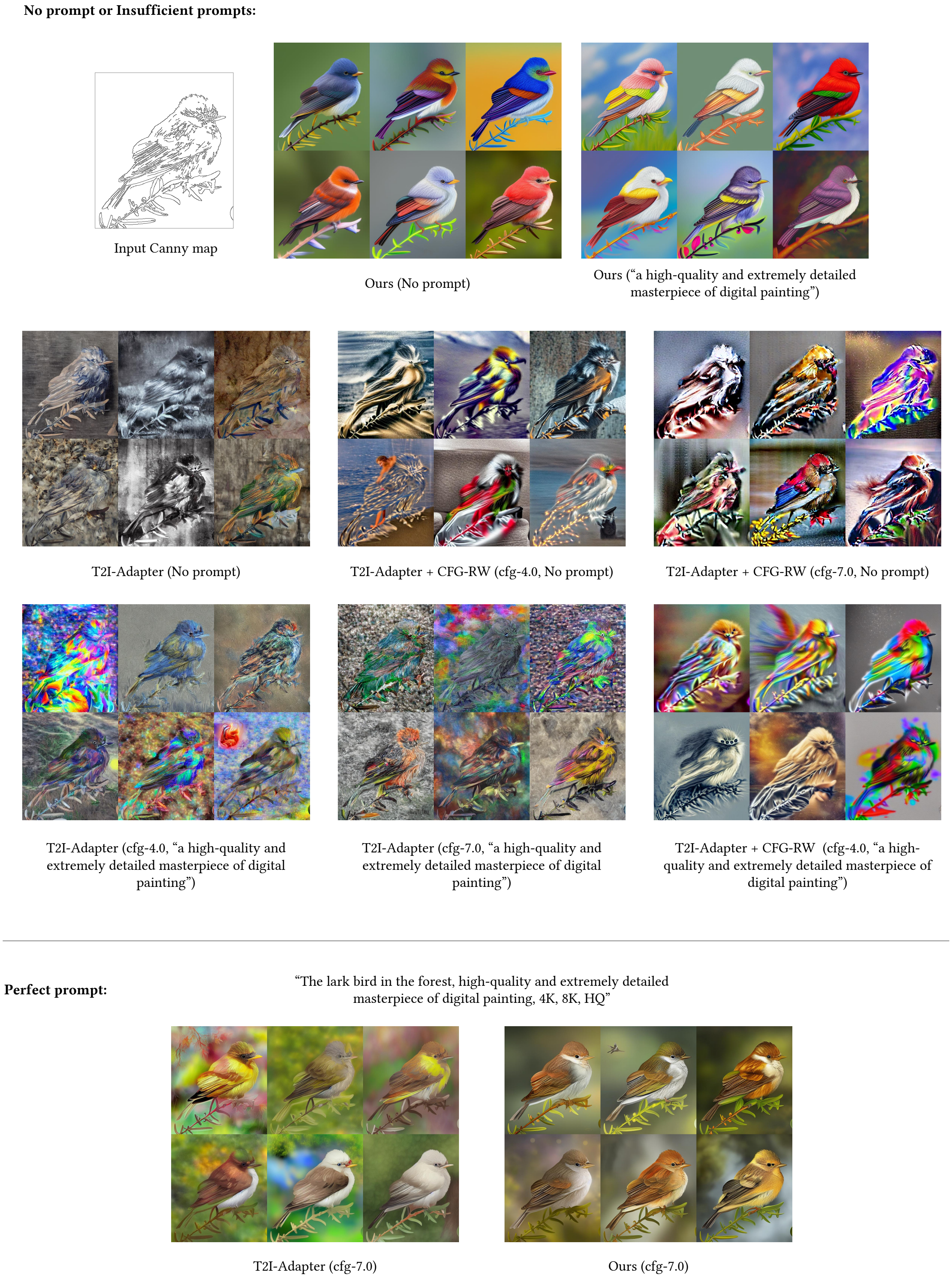

Input
No prompt
Insufficient prompt
(w/o mentioning "flower")
"high-quality and detailed masterpiece"
Conflicting prompt
"ice cream"
Perfect prompt
"a flower, high-quality, extremely detailed, 4K, HQ"
(a) Proposed
(b) w/o zero convolutions
(c) w/o trainable copy, training lightweight layers from scratch (connecting encoder)
(d) w/o trainable copy, training lightweight layers from scratch (connecting decoder)
(e) w/o trainable copy, training lightweight layers from scratch (connecting decoder, using zero conv)
(f) directly train original model

Figure 4: Ablative study results. We compare different architectures with different prompt settings. The tested model is the scribble model and the input scribble is on the top-left.



No prompt or Insufficient prompts:

Input Canny map

Ours (No prompt)

Ours ("a high-quality and extremely detailed masterpiece of digital painting")

T2I-Adapter (No prompt)

T2I-Adapter + CFG-RW (cfg-4.0, No prompt)

T2I-Adapter + CFG-RW (cfg-7.0, No prompt)

T2I-Adapter (cfg-4.0, "a high-quality and extremely detailed masterpiece of digital painting")

T2I-Adapter (cfg-7.0, "a high-quality and extremely detailed masterpiece of digital painting")

T2I-Adapter + CFG-RW (cfg-4.0, "a high-quality and extremely detailed masterpiece of digital painting")

Perfect prompt:

"The lark bird in the forest, high-quality and extremely detailed masterpiece of digital painting, 4K, 8K, HQ"

T2I-Adapter (cfg-7.0)

Ours (cfg-7.0)

Figure 5: Non-cherry-picked Comparison to T2I-Adapter [10]. We compare to T2I-Adapter with different settings.

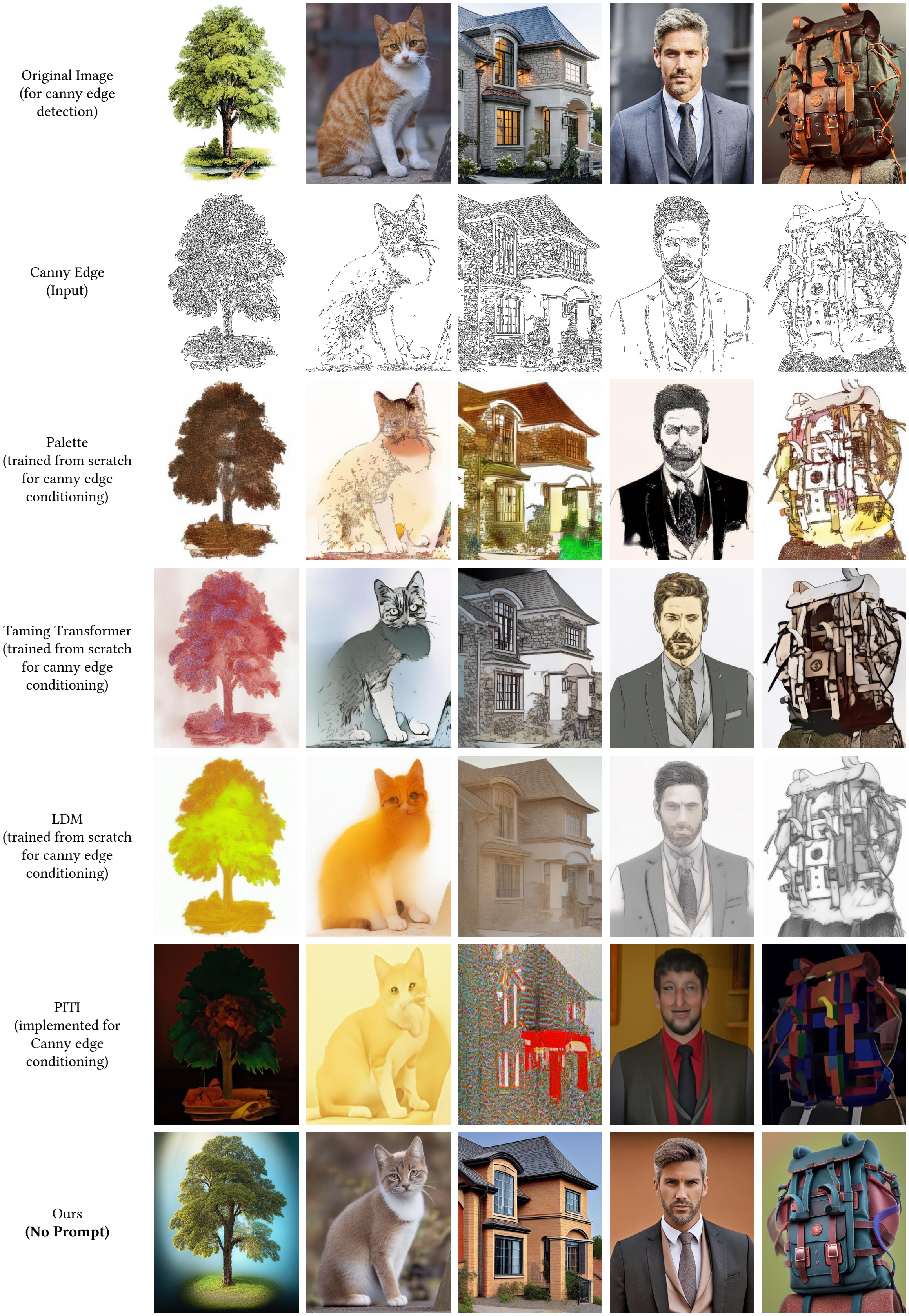
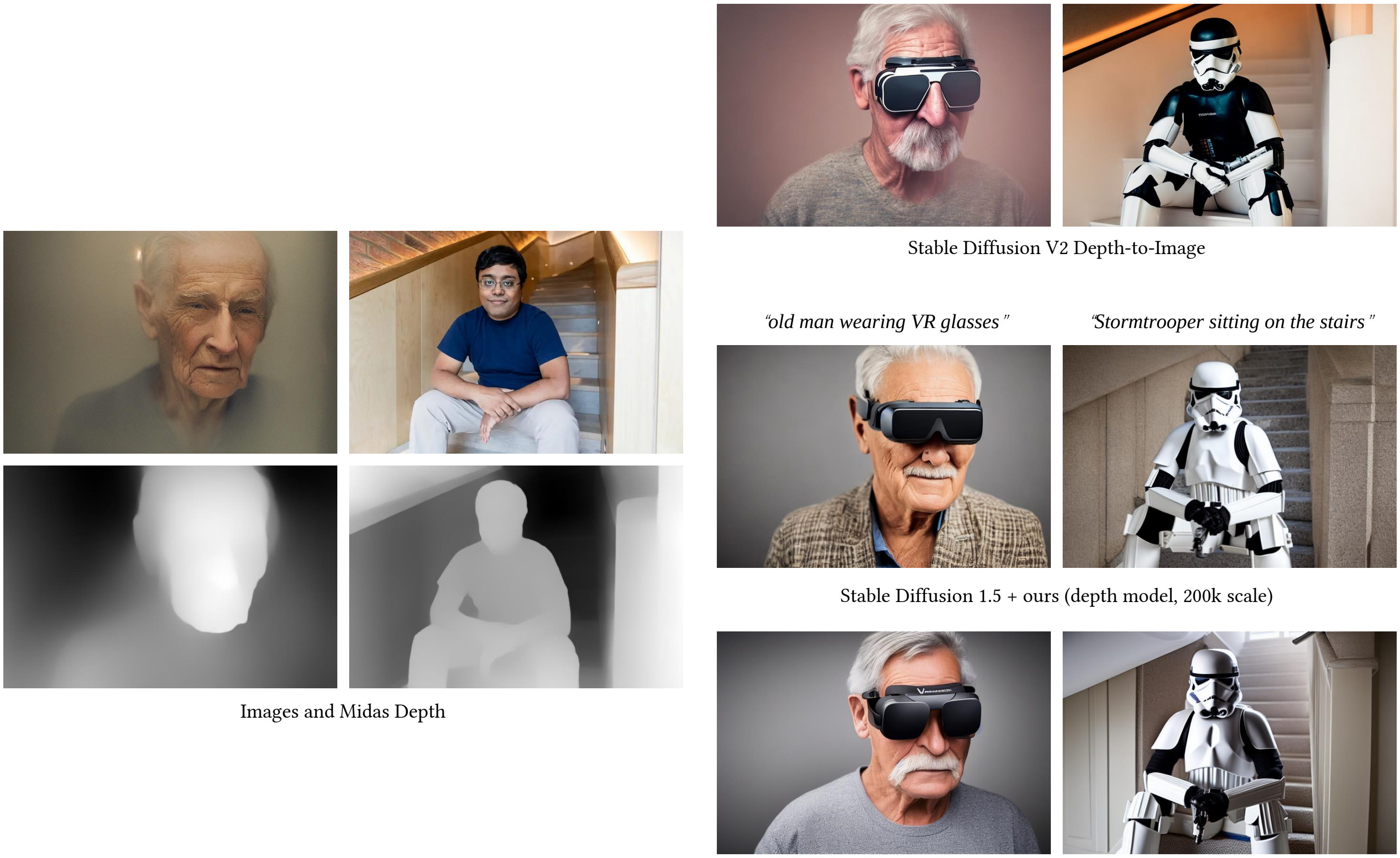

Original Image (for canny edge detection)
Canny Edge (Input)
Palette (trained from scratch for canny edge conditioning)
Taming Transformer (trained from scratch for canny edge conditioning)
LDM (trained from scratch for canny edge conditioning)
PITI (implemented for Canny edge conditioning)
Ours **(No Prompt)**

Figure 6: Canny-edge-to-image in general domain without using prompts. We present visual comparison of different methods.



Stable Diffusion V2 Depth-to-Image

"old man wearing VR glasses"

"Stormtrooper sitting on the stairs"

Images and Midas Depth

Stable Diffusion 1.5 + ours (depth model, 200k scale)

Stable Diffusion 2.1 + ours (depth model, 200k scale)

Figure 7: Comparison of Depth-based ControlNet and Stable Diffusion V2 Depth-to-Image. Note that in this experiment, the Depth-based ControlNet is trained at a relatively small scale to test minimal required computation resources. We also provide relatively stronger models that are trained at relatively large scales.

No Prompt Use empty strings as input prompt, *e.g.*, "".

Insufficient Prompt prompts that do not fully cover objects in conditioning images, *e.g.*, "a high-quality image" that does not mention the actual image contents.

Conflicting Prompt prompts that change the semantics of conditioning images, *e.g.*, "a dog" for a cat image.

Perfect Prompt prompts that describe all necessary content semantics, *e.g.*, "a masterpiece digital painting of a house".

Based on these four basic prompt types, we provide the experiments with prompt derivatives:

Default Prompt We use "a professional, detailed, high-quality image" as a default prompt to generate qualitative results. Note that the default prompt is an **insufficient prompt**, and it can be used for any image.

Automatic Prompt In order to test the state-of-the-art maximized quality of a fully automatic pipeline, we also try using automatic image captioning methods (*e.g.*, BLIP [8]) to generate prompts using the results obtained by “default prompt” mode. In this mode, we generate images with default prompts, then detect the captions automatically, then generate images again with the detected prompts.

User Prompt Users give the prompts.

Since Stable Diffusion relies on CFG-Scale to generate high-quality images, and CFG-Scale uses a “negative prompt” to guide the denoising during inference, we use the below setting:

Negative Prompt For all comparisons to other methods and in the “No Prompt” setting, we use an empty string as the negative prompt. For other qualitative results, we use “ugly, low-quality” as the negative prompt.

CFG Scale We use 7.0 as a default setting of the CFG scale. The CFG-RW (mentioned in the main paper) is applied to “No Prompt” or “Insufficient Prompt” tests.

2 Ablation Study

2.1 Ablative Architectures

We study the following ablative architectures as shown in Figure 1:

Proposed The proposed architecture in the main paper.

Without Zero Convolution Replacing the zero convolutions with standard convolution layers initialized with Gaussian weights.

Lightweight Layers connected to encoder This architecture does not use a trainable copy, and directly initializes single convolution layers for each U-Net level. The outputs are added to the encoder of the original diffusion model.

Lightweight Layers connected to decoder This architecture does not use a trainable copy, and directly initializes single convolution layers for each U-Net level. The outputs are added to the decoder of the original diffusion model.

Lightweight Layers connected to Decoder with zero convolutions Same as “lightweight Layers connected to decoder” but uses zero convolutions to connect to the original diffusion model.

Directly finetune original weights Only adds one layer to the first layer of Stble Diffusion and trains the original Stable Diffusion weights. This architecture is mathematically the same as Stable Diffusion V2’s depth-to-image finetuning method. Note that this architecture has some limitations, like not supporting multiple controls, suffering from overfitting or forgetting when the finetuning dataset is relatively small, and it is relatively difficult to transfer this form of control to other community models.

2.2 Results

We present the results of this ablative study in Figure 2, Figure 3, and Figure 4. We can see that the proposed structure is relatively robust for diverse prompt settings.

3 Comparison

We present a comparison of canny-edge-to-image without using any class guidance or prompts in Figure 6. The compared methods are Palette [14], Taming Transformer [4], LDM [13], PITI [20], and our Canny model. These methods are implemented on the same dataset using the same amount of GPU hours.

Table 2: Average User Ranking (AUR) of result quality and condition fidelity. We report the user preference ranking (1 to 5 indicates worst to best) of different methods.

Method	Result Quality ↑	Condition Fidelity ↑
PIPT [20](sketch)	1.10 ± 0.05	1.02 ± 0.01
Sketch-Guided [19] ($\beta = 1.6$)	3.21 ± 0.62	2.31 ± 0.57
Sketch-Guided [19] ($\beta = 3.2$)	2.52 ± 0.44	3.28 ± 0.72
ControlNet-lite	3.93 ± 0.59	4.09 ± 0.46
ControlNet	$\mathbf{4.22 \pm 0.43}$	$\mathbf{4.28 \pm 0.45}$

Table 3: Evaluation of semantic segmentation label reconstruction (ADE20K) with Intersection over Union (IoU ↑).

ADE20K (GT)	VQGAN [4]	LDM [13]	PIPT [20]	ControlNet-lite	ControlNet
0.58 ± 0.10	0.21 ± 0.15	0.31 ± 0.09	0.26 ± 0.16	0.32 ± 0.12	$\mathbf{0.35 \pm 0.14}$

We present a comparison to the concurrent work T2I-Adapter [10] in Figure 5 using different prompt settings.

We present a comparison to Stable Diffusion V2's depth-to-image in Figure 7 using two different models trained to control Stable Diffusion V1.5 and Stable Diffusion V2.1.

4 Quantitative Evaluation

4.1 User Study

Participant The user study involves 12 people: 10 non-artist amateurs and 2 professionals with artistic or design knowledge.

Setup We sample 20 unseen hand-drawn sketches, and then assign each sketch to 5 methods: PIPT [20]'s sketch model, Sketch-Guided Diffusion (SGD) [19] with default edge-guidance scale ($\beta = 1.6$), SGD [19] with a relatively high edge-guidance scale ($\beta = 3.2$), the aforementioned ControlNet-lite (option (c) in our ablative study), and our proposed method. Note that the Sketch-Guided Diffusion is a re-implementation based on their paper.

User guideline We invited all 12 participants to rank these 20 groups of 5 results individually in terms of *"the quality of displayed images"* and *"the fidelity to the sketch"*. In this way, we obtained 100 rankings for result quality and 100 for condition fidelity.

Evaluation metric We use the Average Human Ranking (AHR) as a preference metric where users rank each result on a scale of 1 to 5 (lower is worse). The average rankings are shown in Table 2.

4.2 Comparison to Industrial Models

Stable Diffusion V2 Depth-to-Image (SDv2-D2I) [17] is trained with a large-scale NVIDIA A100 cluster, thousands of GPU hours, and more than 12M training images.

We train two ControlNets for both SD V1.5 and V2.1 with the same depth conditioning but only using 200K training samples, and a single NVIDIA RTX 3090Ti GPU, with 5 days of training. A visual comparison is presented in Figure 7.

We compare our models for Stable Diffusion V2.1 and the industrial model SDv2-D2I with a user study. We use 100 images generated by each SDv2-D2I and ControlNet and show them to our 12 users in a labeled form so that they can learn to distinguish the two methods. Afterwards, we generate 200 images and ask the users to tell which model generated each image. The average precision of the users is 0.52 ± 0.17, indicating that the two methods yield almost indistinguishable images.

Table 4: Evaluation for image generation conditioned by semantic segmentation. We report FID, CLIP text-image score, and CLIP aesthetic scores for our method and other baselines. We also report the performance of Stable Diffusion without segmentation conditions. Methods marked with "*" are trained from scratch.

Method	FID ↓	CLIP-score ↑	CLIP-aes. ↑
Stable Diffusion	6.09	0.26	6.32
VQGAN [4](seg.)*	26.28	0.17	5.14
LDM [13](seg.)*	25.35	0.18	5.15
PIPT [20](seg.)	19.74	0.20	5.77
ControlNet-lite	17.92	0.26	6.30
ControlNet	15.27	0.26	6.31

4.3 Quantitative Metrics

We quantitatively evaluate the segmentation conditioning fidelity by using semantic segmentation models to segment the generated images again and then compute metrics of the segmentation reconstruction. The state-of-the-art segmentation method OneFormer [6] achieves an Intersection-over-Union (IoU) of 0.58 on the ground-truth set, which means that the error between ground-truth segmentation and OneFormer's segmentation is IoU 0.58. We use different methods to generate images with ADE20K segmentation maps and then apply OneFormer to detect the segmentation maps again to compute the reconstructed IoUs (Table 3), noting that the recomputed average IoU will be generally worse than the 0.58 for the ground truth. This IoU reflects to the extent to which the segmentation conditioning is consistent to the inputs.

The Frechet Inception Distance (FID) is a frequently used metric to evaluate diffusion models. Stable Diffusion's official method uses "50 PLMS steps and 10000 random prompts from the COCO2017 validation set, evaluated at 512x512 resolution" [16]. With this approach, Stable Diffusion V1.5 achieves a FID score of about 6.1.

When Stable Diffusion is controlled by additional conditions with ControlNet, we observe that the FID score is usually worse than that of standard Stable Diffusion V1.5. This may be because (1) Stable Diffusion has gone through a very competitive process to lead the benchmark and any external factors may slightly degrade metrics performance and/or (2) generating highly-controlled images with ControlNet is essentially more difficult than generating random images from prompts and/or (3) more parameter tuning (like cfg-scale) is needed to improve performance of ControlNet on this metric.

These results suggest that the FID score should not be used as a standalone metric to evaluate how well a model controls Stable Diffusion because the worst control (a model that does not influence SD at all) is equivalent to standalone Stabel Diffusion and will report a better FID score. Because of this, using multiple metrics, like the aforementioned "conditioning fidelity" metric is a must for a comprehensive evaluation.

To be specific, we use the Stable Diffusion's evaluation set of 10000 samples to evaluate different models' performance using the COCO segmentation conditioning. We use OneFormer to generate paired semantic segmentation maps of these samples and use random crops of 512×512 images as FID's target set. The FID's source set are 512×512 images generated by different methods using the paired semantic segmentation maps. The results are presented in Table 4.

We also present text-image CLIP scores [11] and CLIP aesthetic score [15] in Table 4. We observe that additional conditioning with ControlNet seems to have minimal influence on CLIP scores and aesthetic scores. The performance of methods that are not based on Stable Diffusion is generally worse than methods that use Stable Diffusion.

5 Gradient Calculation of Zero Convolution Layers

We briefly describe the gradient calculation of a zero convolution layer. Consider a 1×1 convolution layer with weight $\boldsymbol{W}$ and bias $\boldsymbol{B}$, at spatial position p and channel-wise index i. Given an input map

$\boldsymbol{I} \in \mathbb{R}^{h\times w\times c}$, the forward pass can be written as

$$\mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W},\boldsymbol{B}\})_{p,i} = \boldsymbol{B}_i + \sum_j^c \boldsymbol{I}_{p,j}\boldsymbol{W}_{i,j}. \tag{1}$$

Since a zero convolution layer in initialized with $\boldsymbol{W} = \boldsymbol{0}$ and $\boldsymbol{B} = \boldsymbol{0}$ (*i.e.*, before any optimization steps), anywhere that $\boldsymbol{I}_{p,i} \neq \boldsymbol{0}$ the gradients become

$$\begin{cases} \dfrac{\partial \mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W},\boldsymbol{B}\})_{p,i}}{\partial \boldsymbol{B}_i} = 1, \\ \dfrac{\partial \mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W},\boldsymbol{B}\})_{p,i}}{\partial \boldsymbol{I}_{p,i}} = \sum_j^c \boldsymbol{W}_{i,j} = 0, \\ \dfrac{\partial \mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W},\boldsymbol{B}\})_{p,i}}{\partial \boldsymbol{W}_{i,j}} = \boldsymbol{I}_{p,j} \neq \boldsymbol{0}. \end{cases} \tag{2}$$

We see that although a zero convolution can cause the gradient on the feature term $\boldsymbol{I}$ to become zero, the gradients for the weight and bias are not influenced. As long as the feature $\boldsymbol{I}$ is non-zero, the weight $\boldsymbol{W}$ will be optimized into a non-zero matrix in the first gradient descent iteration. Notably, in our case, the feature term is input data or condition vectors sampled from datasets, which naturally ensures non-zero $\boldsymbol{I}$.

For example, consider classic gradient descent with an overall loss function $\mathcal{L}$ and a learning rate $\beta_{\text{lr}} \neq 0$, if the "outside" gradient $\partial\mathcal{L}/\partial\mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W},\boldsymbol{B}\})$ is not zero, we have

$$\boldsymbol{W}^* = \boldsymbol{W} - \beta_{\text{lr}} \cdot \frac{\partial \mathcal{L}}{\partial \mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W},\boldsymbol{B}\})} \odot \frac{\partial \mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W},\boldsymbol{B}\})}{\partial \boldsymbol{W}} \neq \boldsymbol{0}, \tag{3}$$

where $\boldsymbol{W}^*$ is the weight after one gradient descent step and $\odot$ is Hadamard product. After this step, we have

$$\frac{\partial \mathcal{Z}(\boldsymbol{I};\{\boldsymbol{W}^*,\boldsymbol{B}\})_{p,i}}{\partial \boldsymbol{I}_{p,j}} = \sum_j^c \boldsymbol{W}^*_{i,j} \neq \boldsymbol{0}, \tag{4}$$

where non-zero gradients are obtained and the neural network begins to learn. In this way, the zero convolutions become a unique type of connection layer that progressively grows parameters from zero to optimized values in a learned way.

6 Additional Results

We present the results of the "Canny Edge" model in Figure 8.

We present the results of the "Hough Line" model in Figure 9.

We present the results of the "HED Boundary" model in Figure 11.

We present the results of the "User Sketching" model in Figure 10.

We present the results of the "Human Pose (Openpifpaf)" model in Figure 12.

We present the results of the "Semantic Segmentation (COCO)" model in Figure 16.

We present the results of the "Semantic Segmentation (ADE20K)" model in Figure 15.

We present the results of the "Normal Map" model in Figure 17.

We present the results of the "Cartoon Line Drawing" model in Figure 18.

We present the results of six control types based on the same source image, including Canny Edge, HED, M-LSD Line, Depth, Normal, and Scribbles, in Figure 23, Figure 24, Figure 25.

7 Discussion

Figure 1 compares a model trained without using ControlNet. That model is trained with exactly the same method as Stability's Depth-to-Image model (Adding a channel to the SD and continuing the training).

Figure 21 shows the training process. We see a “sudden convergence phenomenon” where the model suddenly is able to follow the input conditions. We have seen this happen during the training process typically somewhere between 5000 and 10000 steps when using 1e-5 as the learning rate.

Figure 22 shows Canny-edge-based ControlNets trained with different dataset scales.

Figure 19 shows that if the diffusion process is masked, the models can be used in pen-based image editing.

Figure 26 shows that when object is relatively simple, the model can achieve relatively accurate control of the details.

Figure 27 shows that when ControlNet is only applied to 50% diffusion iterations, users can get results that do not follow the input shapes.

Figure 28 shows that when the semantic interpretation is wrong, the model may have difficulty generating correct contents.

Figure 29 shows all source images in this paper for edge detection, pose extraction, *etc*.

References

[1] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018.

[2] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[4] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

[5] G. Gu, B. Ko, S. Go, S.-H. Lee, J. Lee, and M. Shin. Towards light-weight and real-time line segment detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[6] J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, and H. Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. 2023.

[7] S. Kreiss, L. Bertoni, and A. Alahi. OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–14, March 2021.

[8] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 2022.

[9] A. Mercurio. Waifu diffusion, 2022.

[10] C. Mou, X. Wang, L. Xie, J. Zhang, Z. Qi, Y. Shan, and X. Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[12] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1623–1637, 2020.

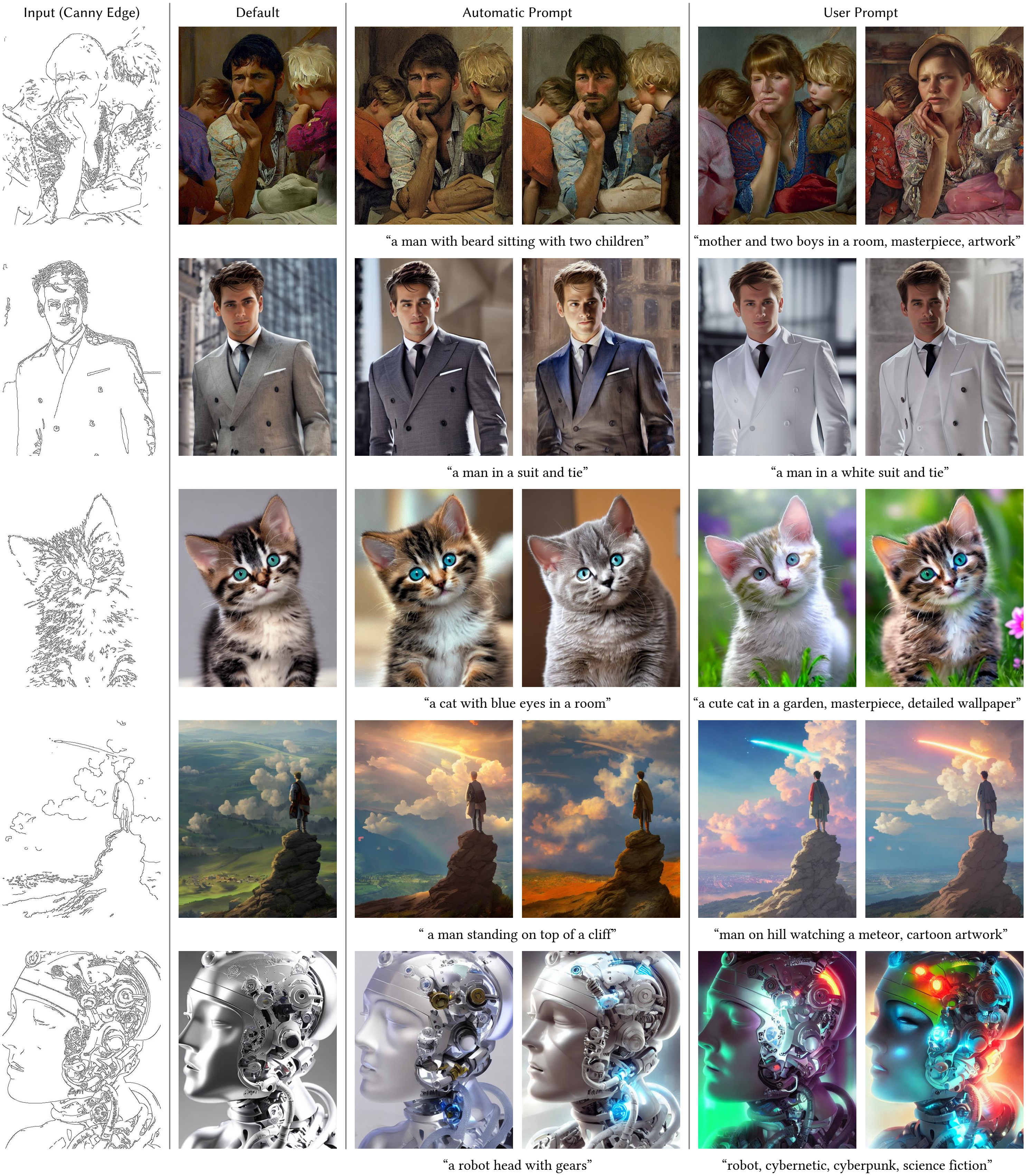

Input (Canny Edge)
Default
Automatic Prompt
User Prompt
“a man with beard sitting with two children”
“mother and two boys in a room, masterpiece, artwork”
“a man in a suit and tie”
“a man in a white suit and tie”
“a cat with blue eyes in a room”
“a cute cat in a garden, masterpiece, detailed wallpaper”
“ a man standing on top of a cliff”
“man on hill watching a meteor, cartoon artwork”
“a robot head with gears”
“robot, cybernetic, cyberpunk, science fiction”

Figure 8: Controlling Stable Diffusion with Canny edges. The “automatic prompts” are generated by BLIP based on the default result images without using user prompts. See also the Appendix for source images for canny edge detection.

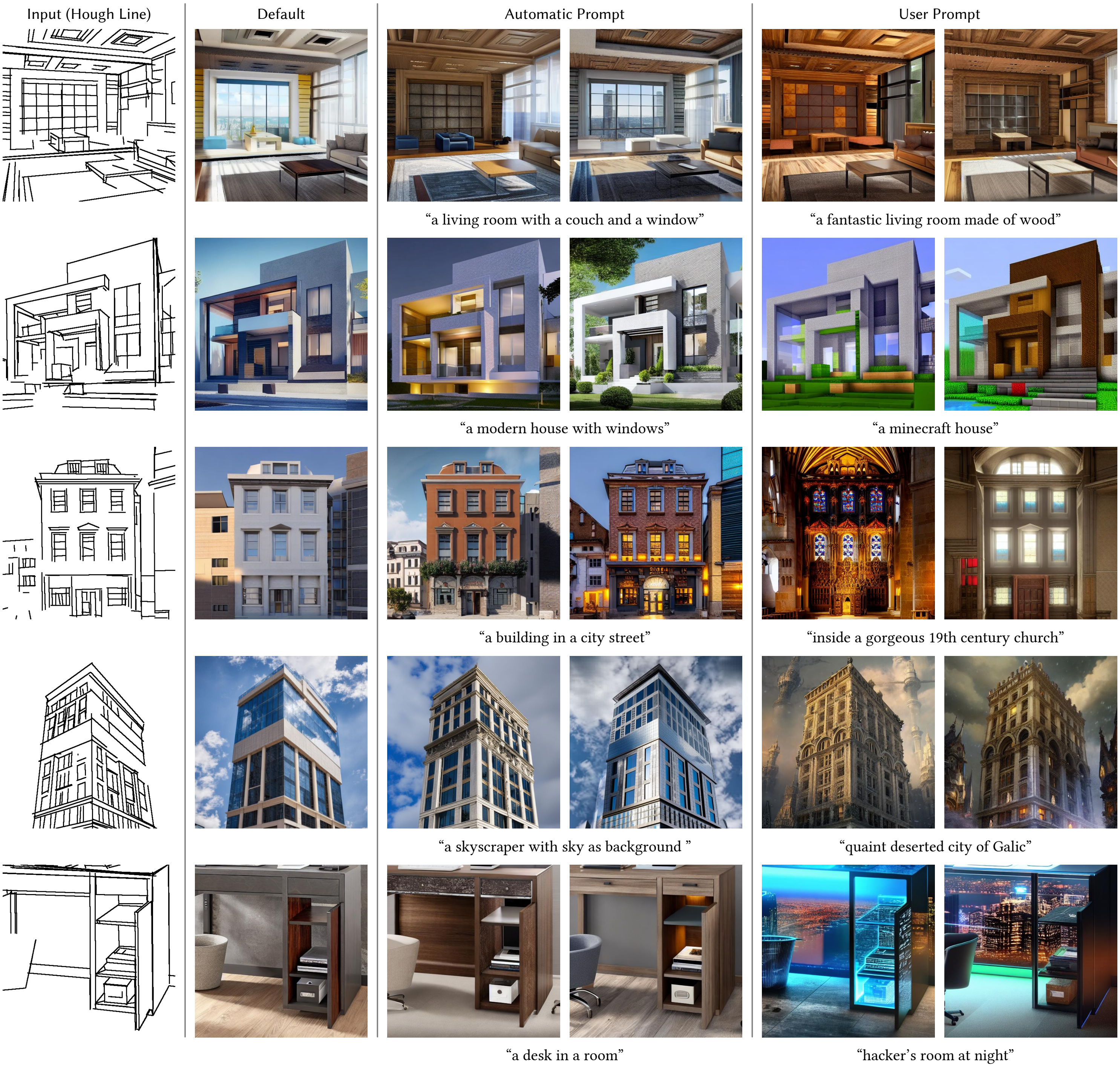
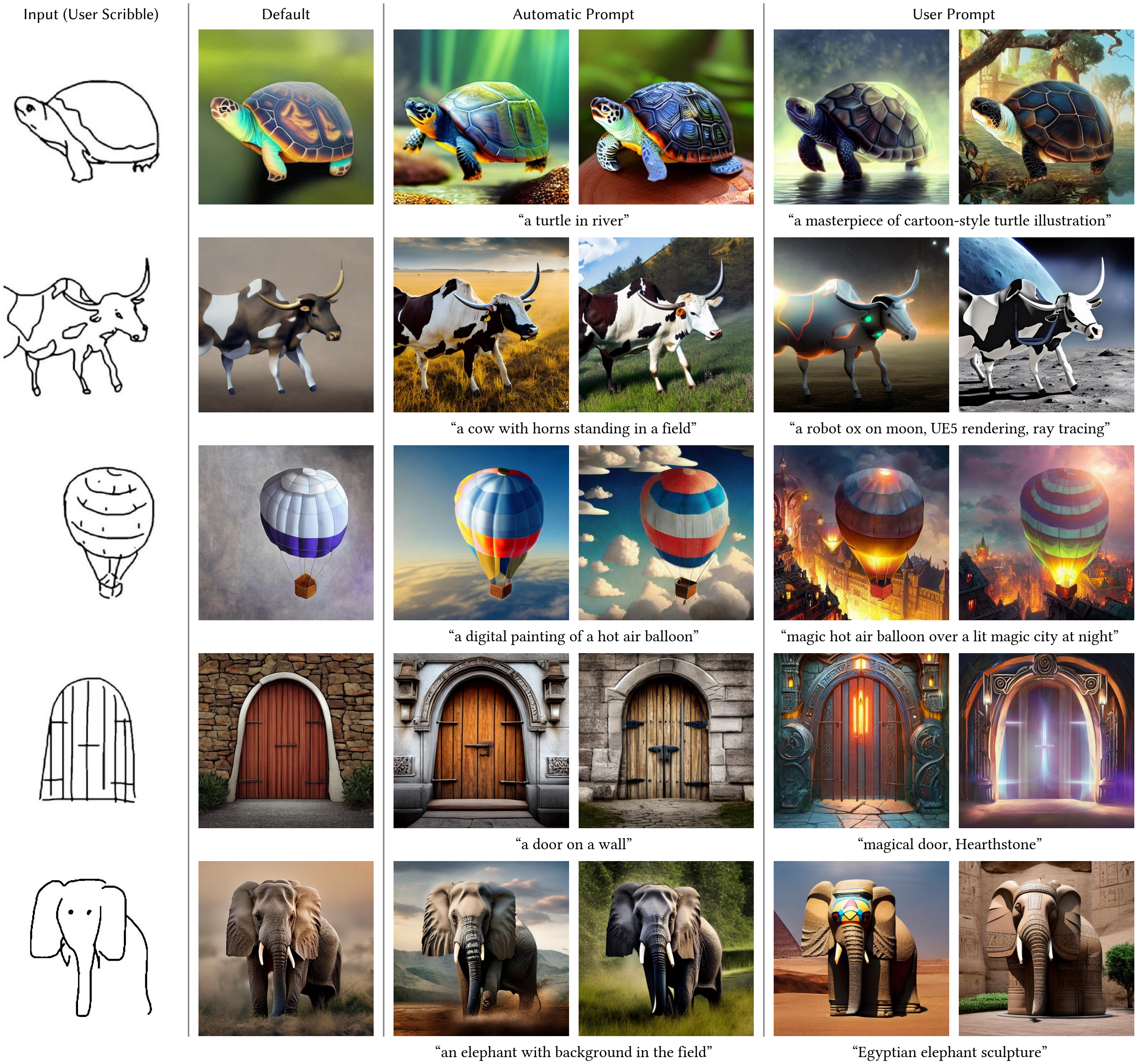

Input (Hough Line)
Default
Automatic Prompt
User Prompt
“a living room with a couch and a window”
“a fantastic living room made of wood”
“a modern house with windows”
“a minecraft house”
“a building in a city street”
“inside a gorgeous 19th century church”
“a skyscraper with sky as background ”
“quaint deserted city of Galic”
“a desk in a room”
“hacker’s room at night”

Figure 9: Controlling Stable Diffusion with Hough lines (M-LSD). The “automatic prompts” are generated by BLIP based on the default result images without using user prompts. See also the Appendix for source images for line detection.



"a turtle in river"

"a masterpiece of cartoon-style turtle illustration"

"a cow with horns standing in a field"

"a robot ox on moon, UE5 rendering, ray tracing"

"a digital painting of a hot air balloon"

"magic hot air balloon over a lit magic city at night"

"a door on a wall"

"magical door, Hearthstone"

"an elephant with background in the field"

"Egyptian elephant sculpture"

Figure 10: Controlling Stable Diffusion with Human scribbles. The "automatic prompts" are generated by BLIP based on the default result images without using user prompts. These scribbles are from [19].

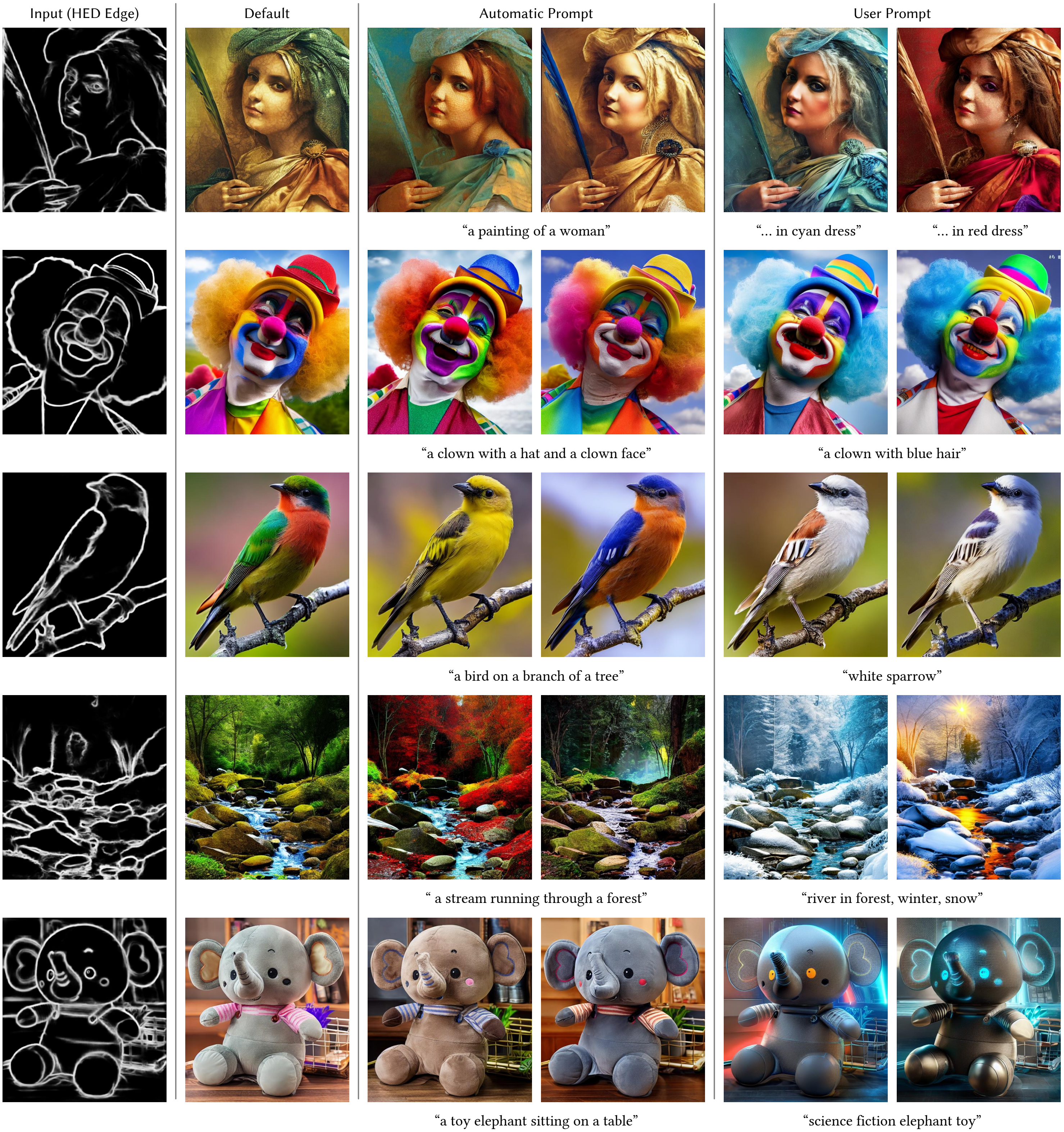

Input (HED Edge)
Default
Automatic Prompt
User Prompt
“a painting of a woman”
“... in cyan dress”
“... in red dress”
“a clown with a hat and a clown face”
“a clown with blue hair”
“a bird on a branch of a tree”
“white sparrow”
“ a stream running through a forest”
“river in forest, winter, snow”
“a toy elephant sitting on a table”
“science fiction elephant toy”

Figure 11: Controlling Stable Diffusion with HED boundary map. The “automatic prompts” are generated by BLIP based on the default result images without using user prompts. See also the Appendix for source images for HED boundary detection.

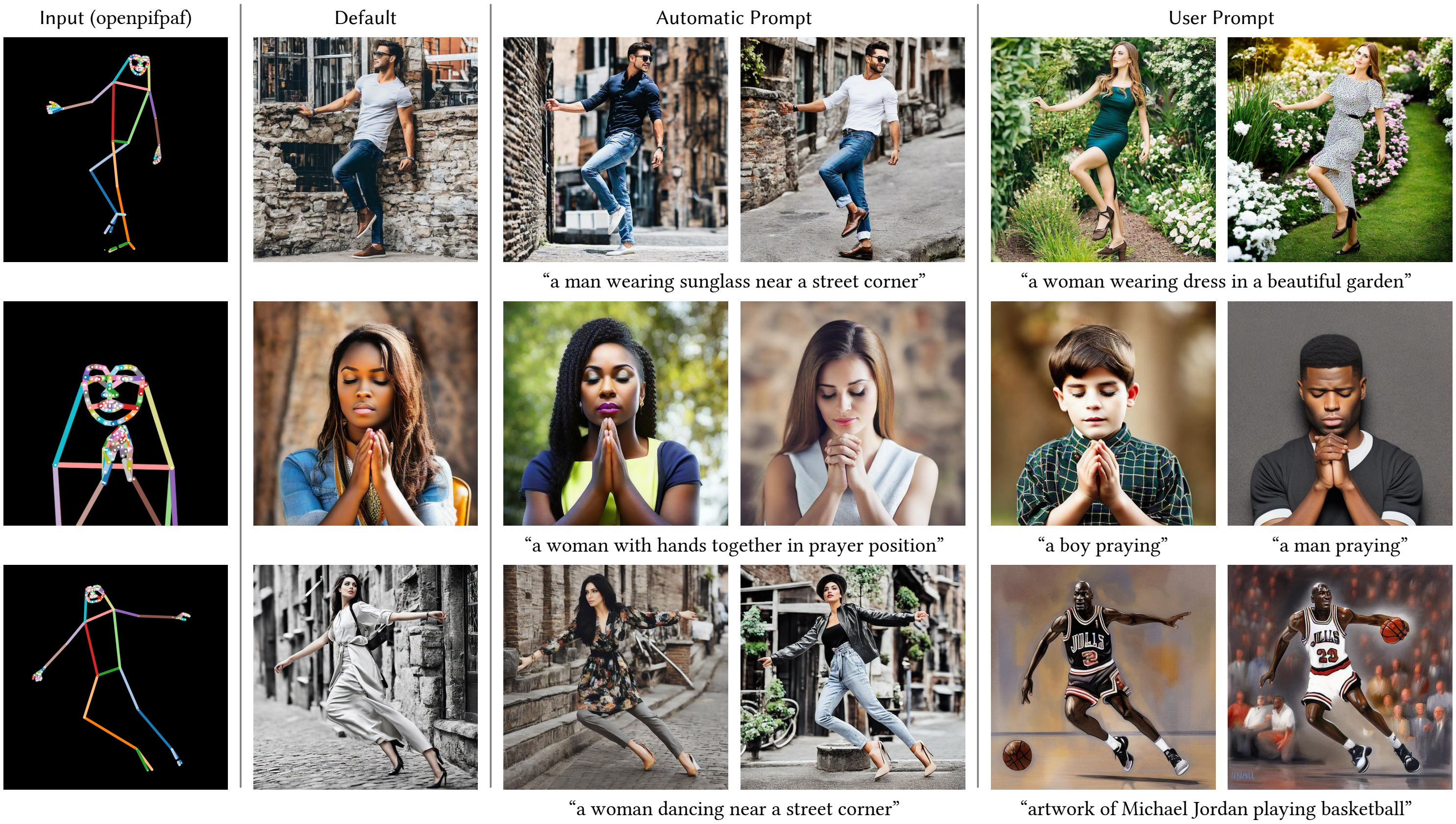

Input (openpifpaf)
Default
Automatic Prompt
User Prompt
"a man wearing sunglass near a street corner"
"a woman wearing dress in a beautiful garden"
"a woman with hands together in prayer position"
"a boy praying"
"a man praying"
"a woman dancing near a street corner"
"artwork of Michael Jordan playing basketball"

Figure 12: Controlling Stable Diffusion with Openpifpaf pose. See also the Appendix for source images for Openpifpaf pose detection.

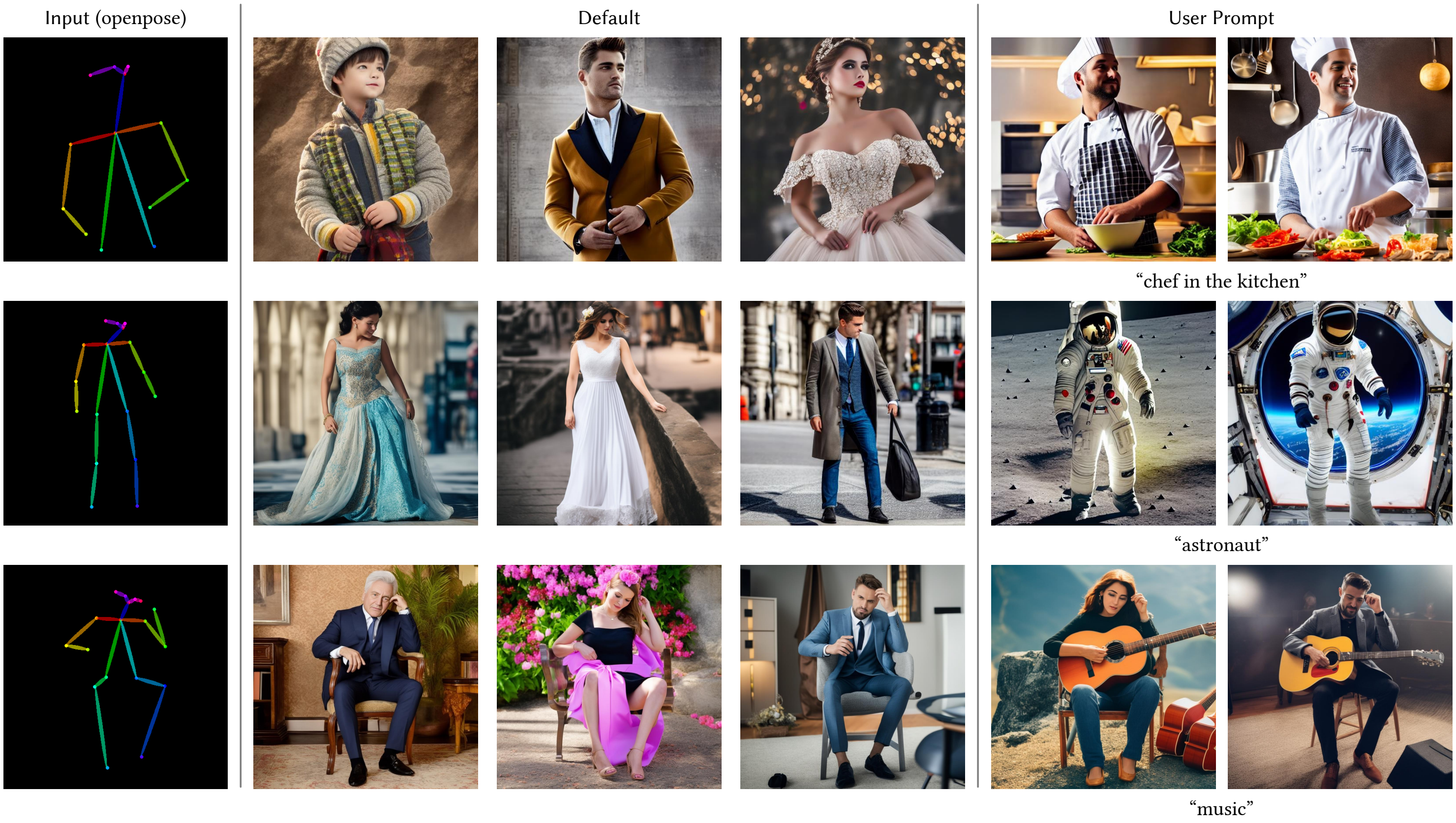


Figure 13: Controlling Stable Diffusion with Openpose. See also the Appendix for source images for Openpose pose detection.

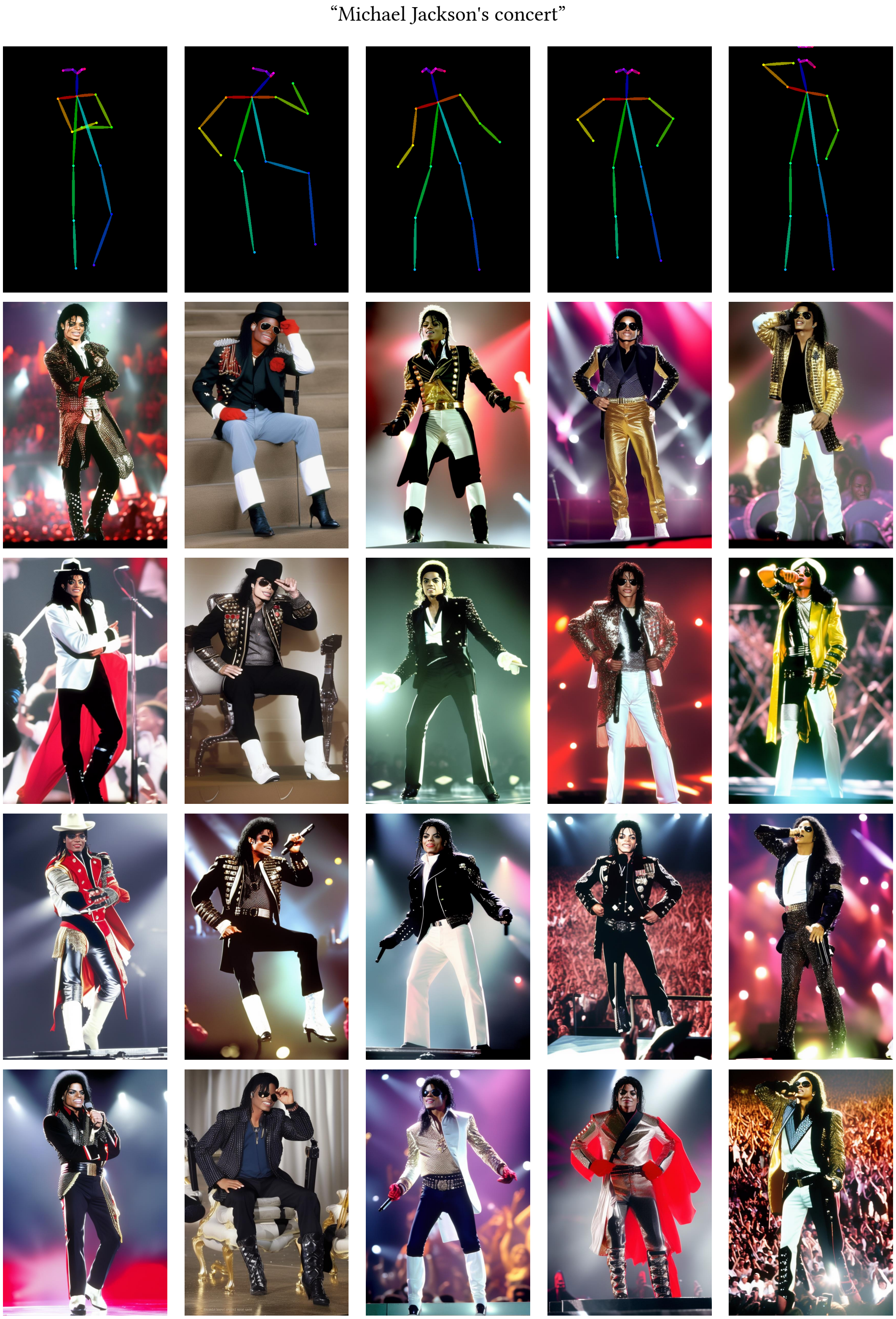


Figure 14: Controlling Stable Diffusion with human pose to generate different poses for a same person (“Michael Jackson’s concert”). Images are not cherry picked. See also the Appendix for source images for Openpose pose detection.

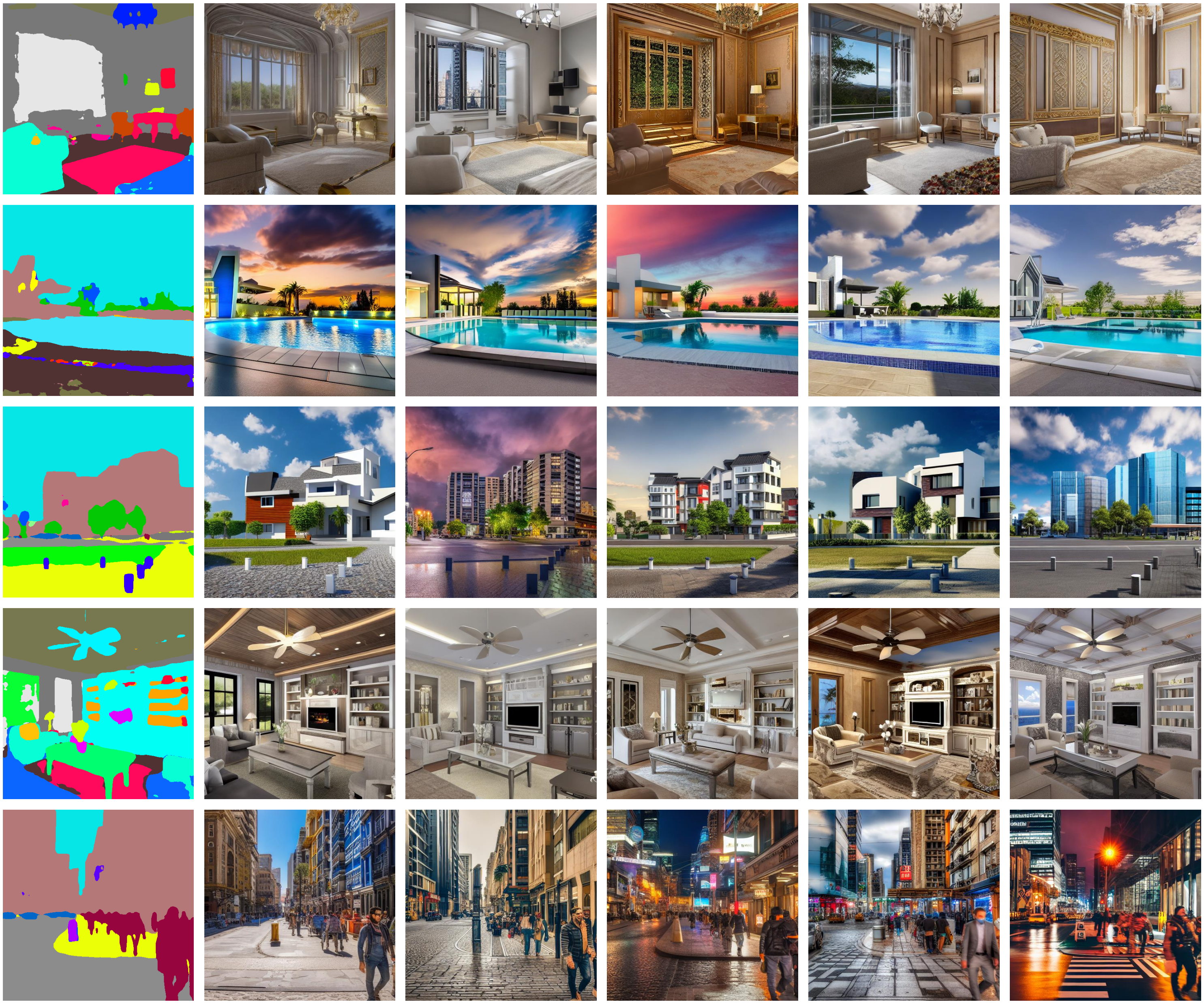


Figure 15: Controlling Stable Diffusion with ADE20K segmentation map. All results are achieved with default prompt. See also the Appendix for source images for semantic segmentation map extraction.

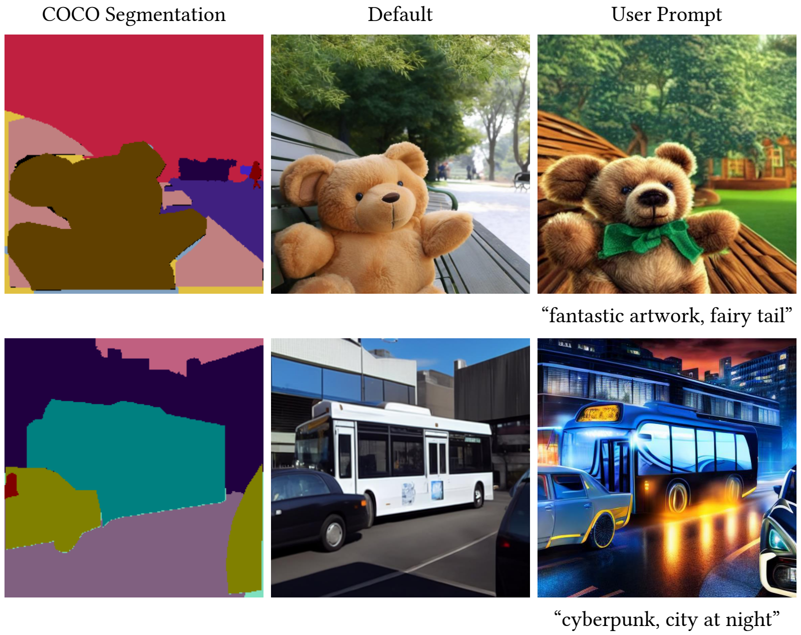


Figure 16: Controlling Stable Diffusion with COCO-Stuff [1] segmentation map.

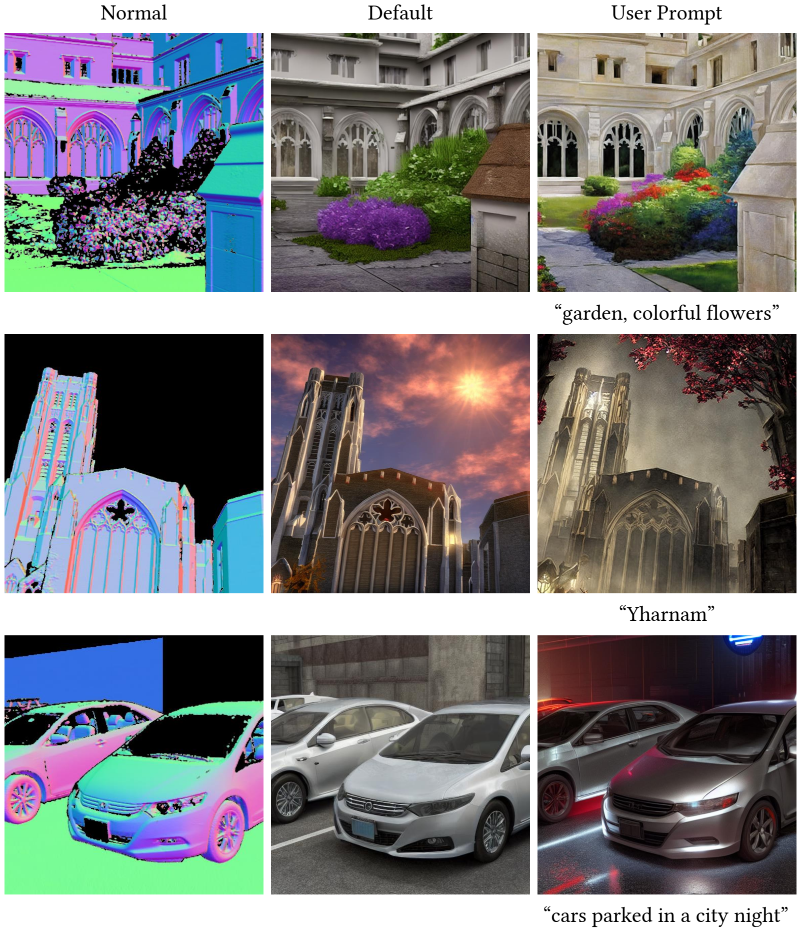


Figure 17: Controlling Stable Diffusion with DIODE [18] normal map.

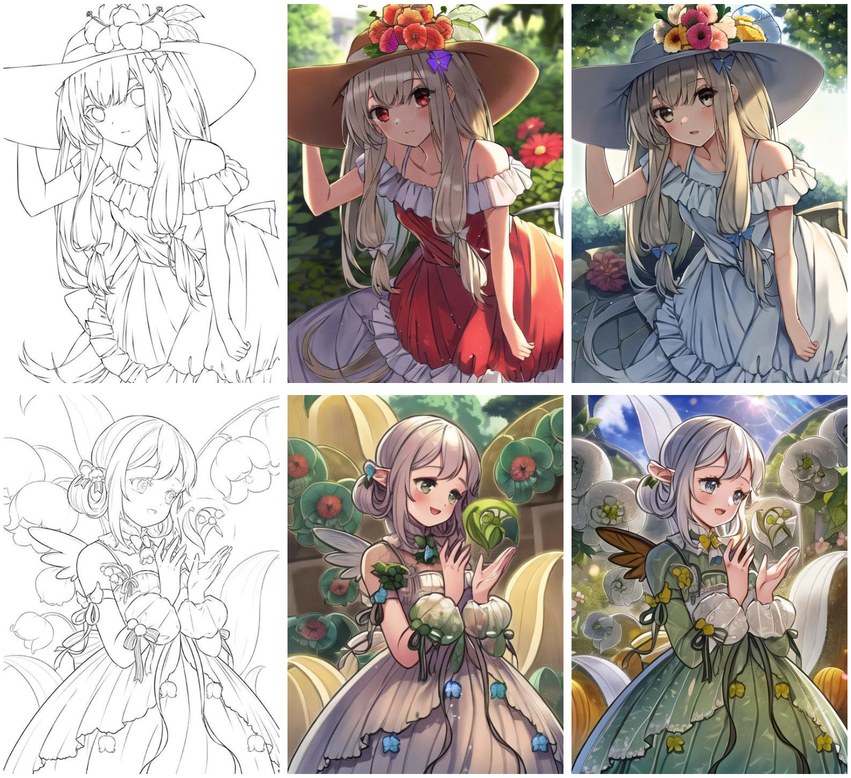

Cartoon line drawing
“1girl, masterpiece, best quality, ultra-detailed, illustration”

Figure 18: Controlling Stable Diffusion (anime weights) with cartoon line drawings. The line drawings are inputs and there are no corresponding “ground truths”. This model may be used in artistic creation tools.

[13] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

[14] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH ’22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393379.

[15] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

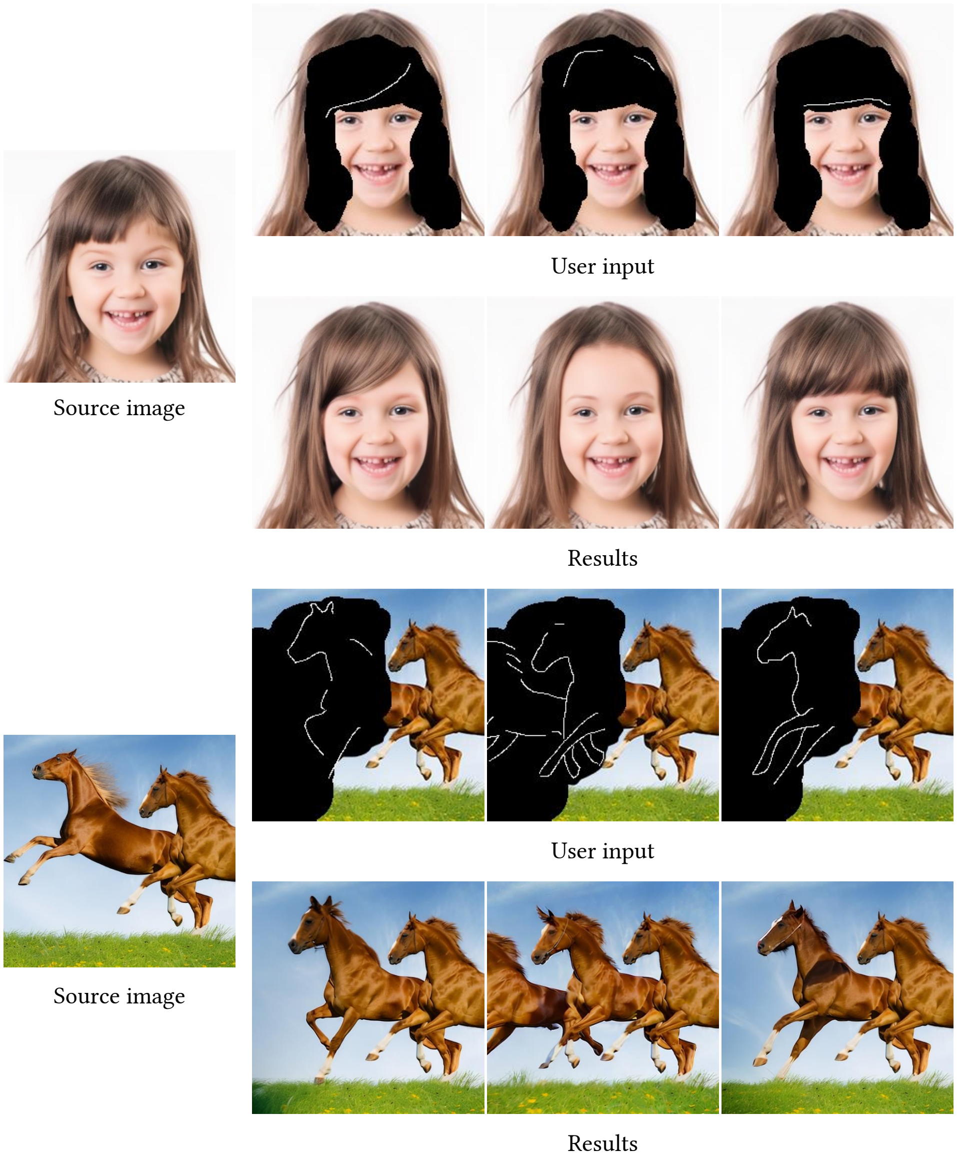
[16] Stability. Stable diffusion v1.5 model card, https://huggingface.co/runwayml/stable-diffusion-v1-5, 2022.

[17] Stability. Stable diffusion v2 model card, stable-diffusion-2-depth, https://huggingface.co/stabilityai/stable-diffusion-2-depth, 2022.

[18] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

[19] A. Voynov, K. Abernan, and D. Cohen-Or. Sketch-guided text-to-image diffusion models. 2022.

[20] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen. Pretraining is all you need for image-to-image translation. 2022.



User input

Source image

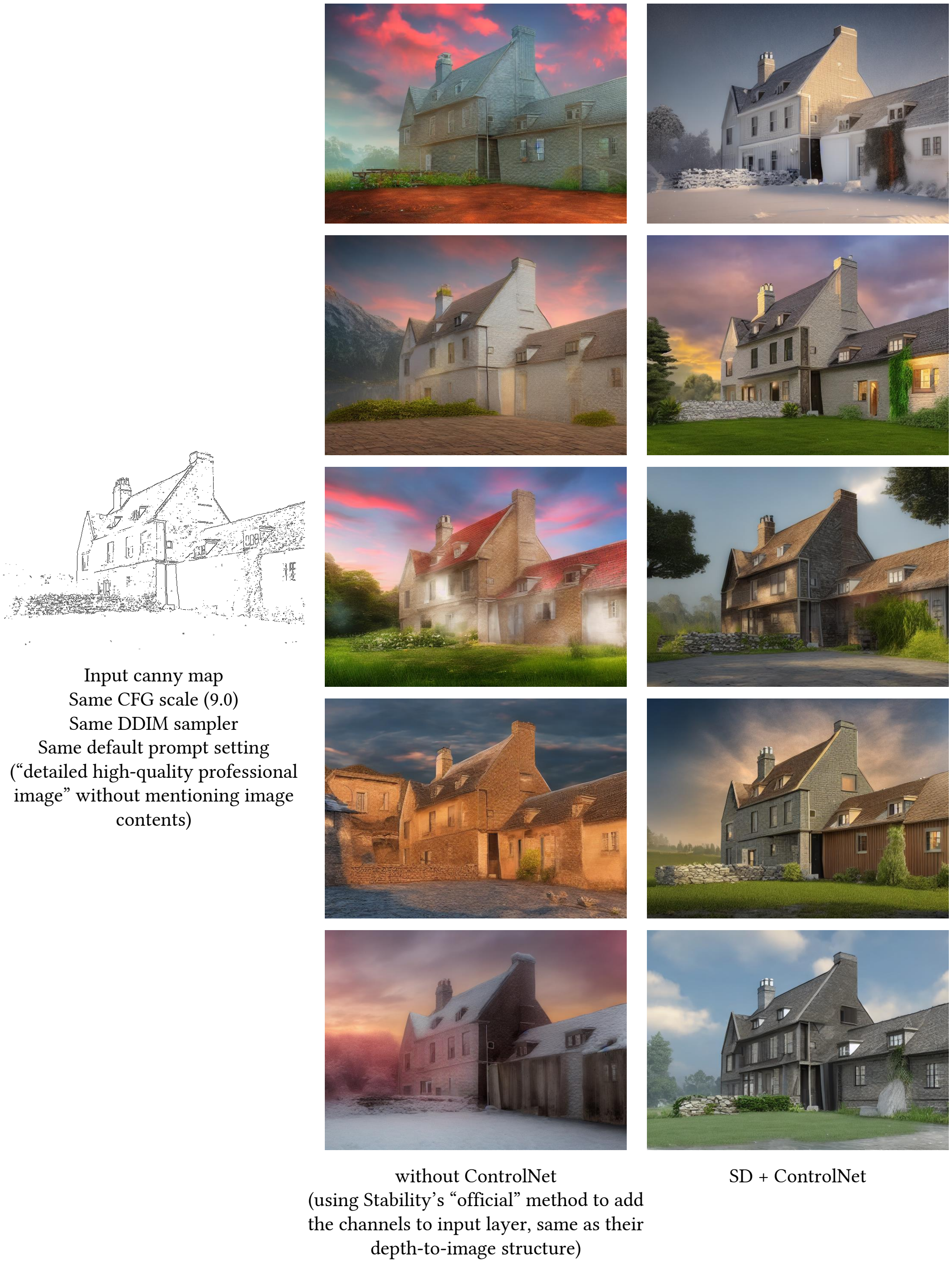
Results

User input

Source image

Results

Figure 19: Masked Diffusion. By diffusing images in masked areas, the Canny-edge model can be used to support pen-based editing of image contents. Since all diffusion models naturally support masked diffusion, the other models are also likely to be used in manipulating images.

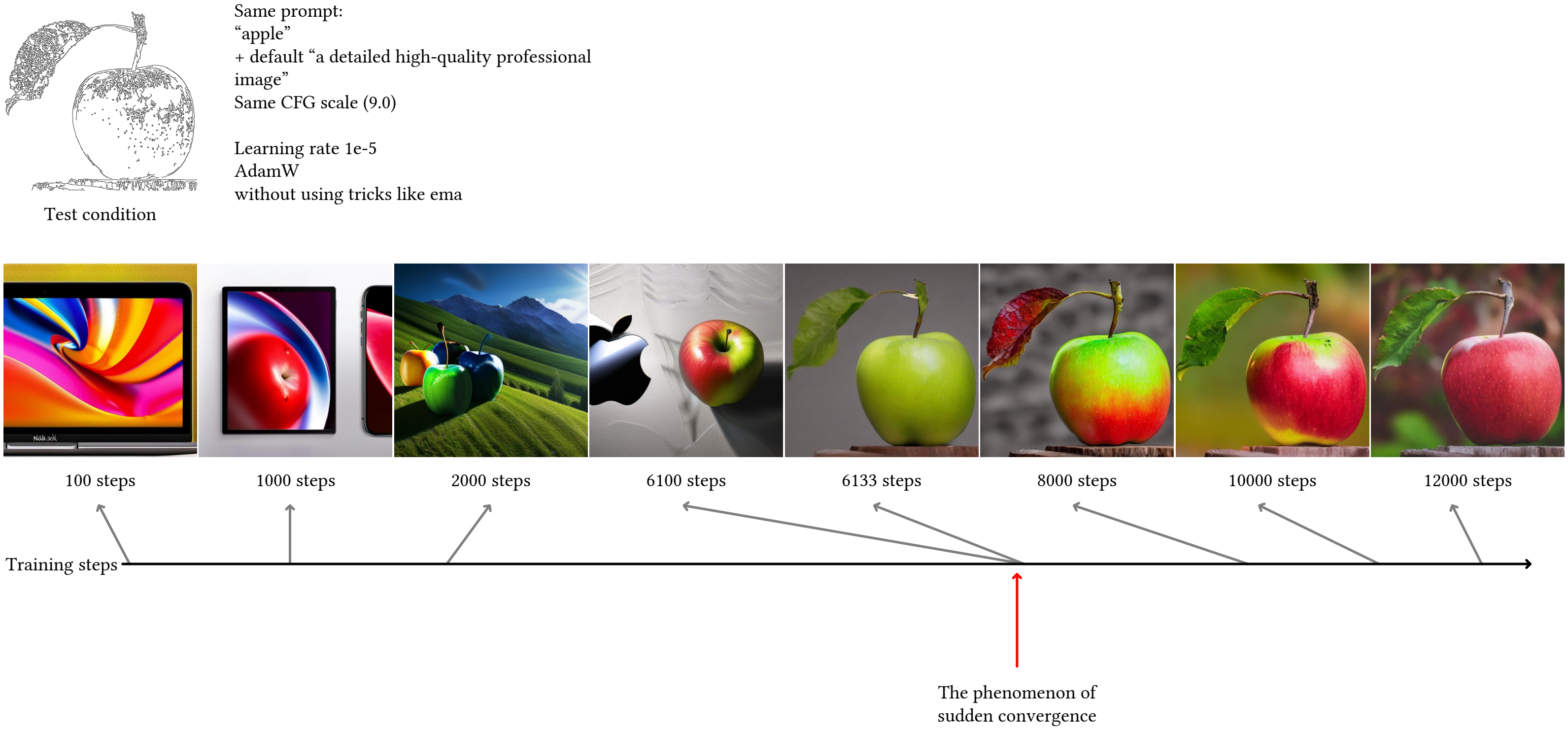


Input canny map
Same CFG scale (9.0)
Same DDIM sampler
Same default prompt setting
(“detailed high-quality professional image” without mentioning image contents)

without ControlNet
(using Stability’s “official” method to add the channels to input layer, same as their depth-to-image structure)

SD + ControlNet

Figure 20: Ablative study. We compare the ControlNet structure with a standard method that Stable Diffusion uses as default way to add conditions to diffusion models.



Same prompt:
"apple"
+ default "a detailed high-quality professional image"
Same CFG scale (9.0)

Learning rate 1e-5
AdamW
without using tricks like ema

Figure 21: The sudden converge phenomenon. Because we use zero convolutions, the neural network always predicts high-quality images during the entire training. At a certain point in training steps, the model suddenly learns to adapt to the input conditions. We call this "sudden converge phenomenon".

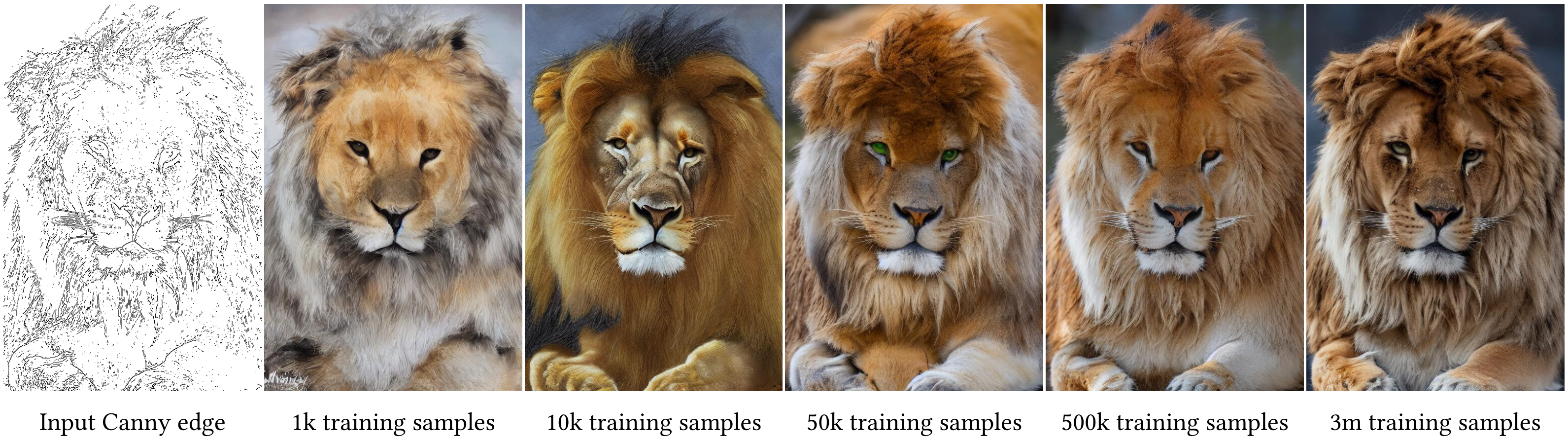
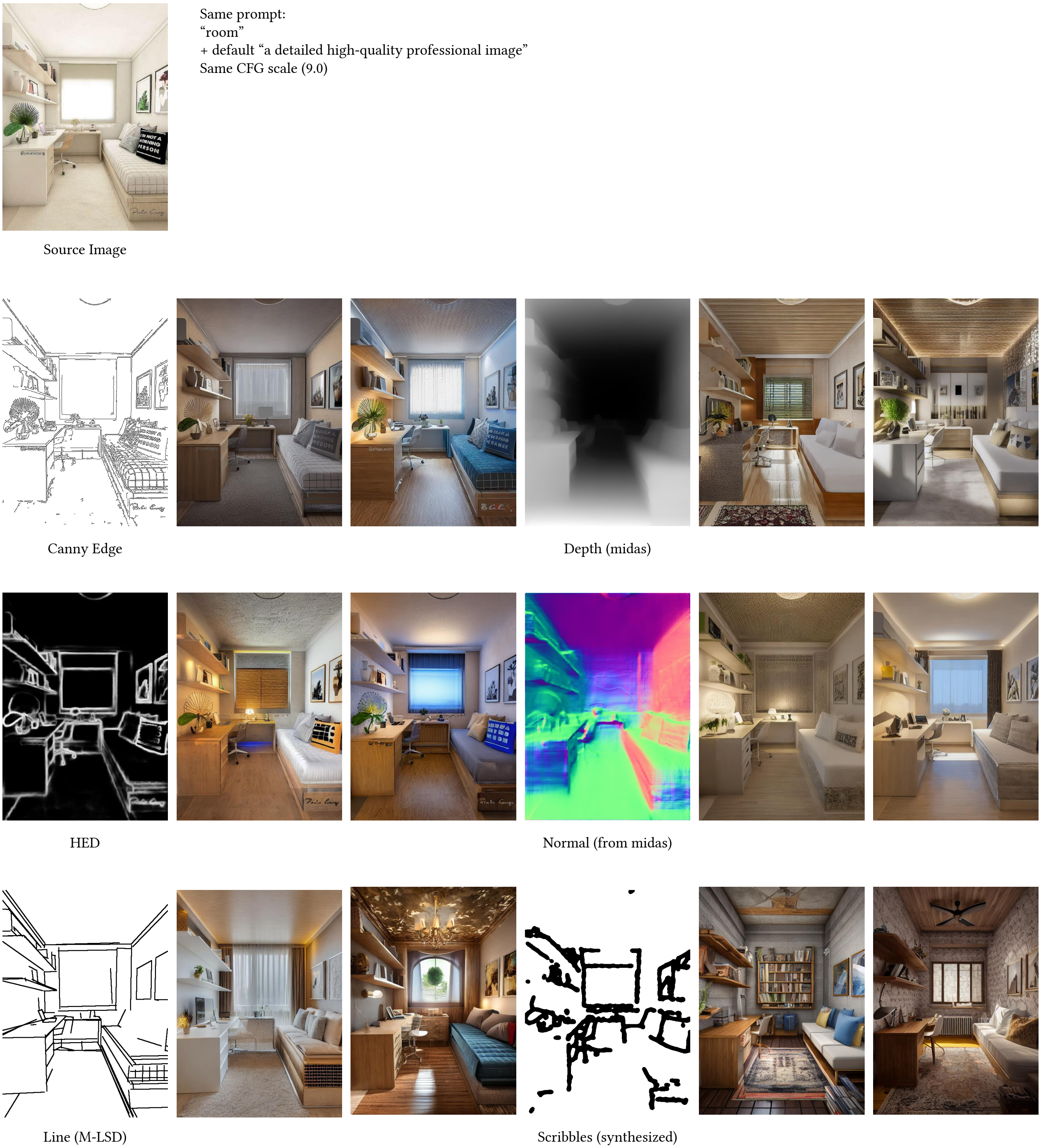


Figure 22: Training on different dataset sizes. We show the Canny-edge-based ControlNet trained on different experimental settings with various dataset size.



Same prompt:
“room”
+ default “a detailed high-quality professional image”
Same CFG scale (9.0)

Figure 23: Comparison of six detection types and the corresponding results. The scribble map is extracted from the HED map with morphological transforms.

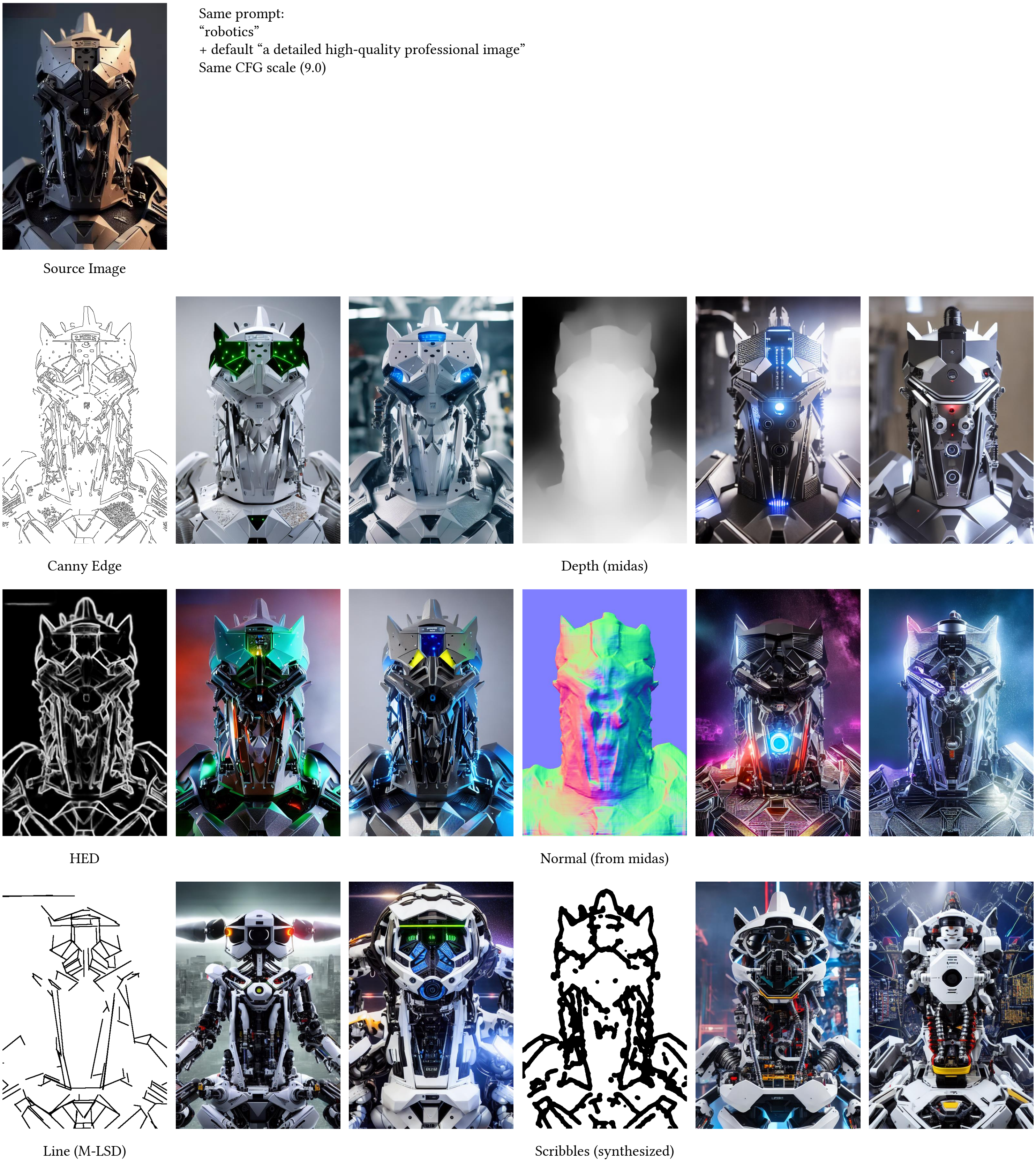
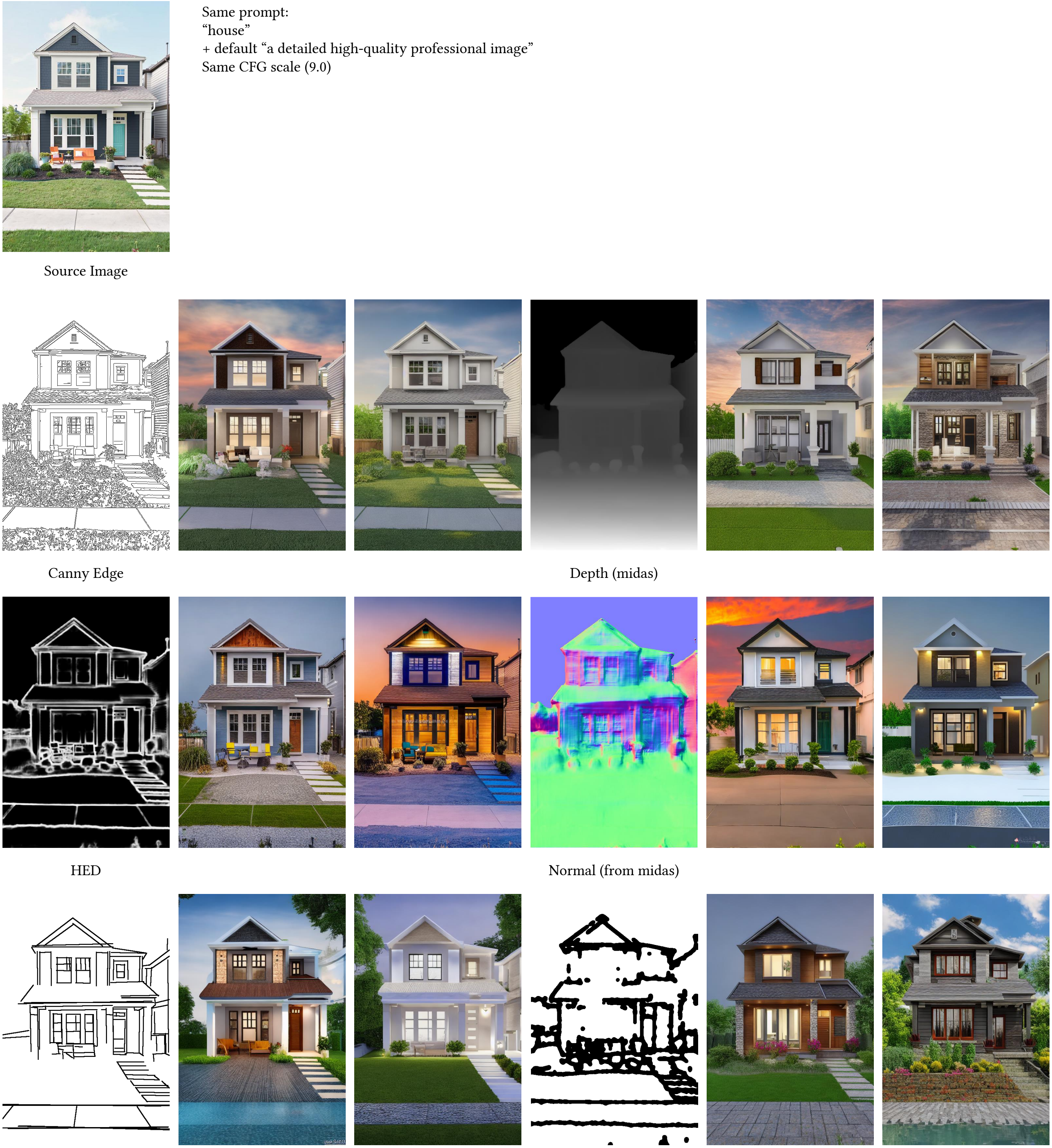


Figure 24: (Continued) Comparison of six detection types and the corresponding results. The scribble map is extracted from the HED map with morphological transforms.



Same prompt:
“house”
+ default “a detailed high-quality professional image”
Same CFG scale (9.0)

Source Image

Canny Edge

Depth (midas)

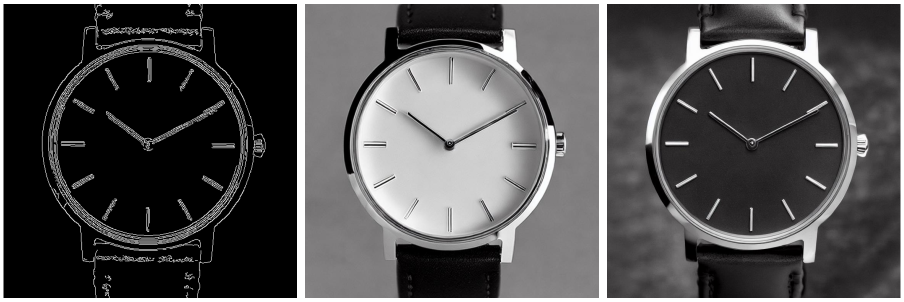
HED

Normal (from midas)

Line (M-LSD)

Scribbles (synthesized)

Figure 25: (Continued) Comparison of six detection types and the corresponding results. The scribble map is extracted from the HED map with morphological transforms.

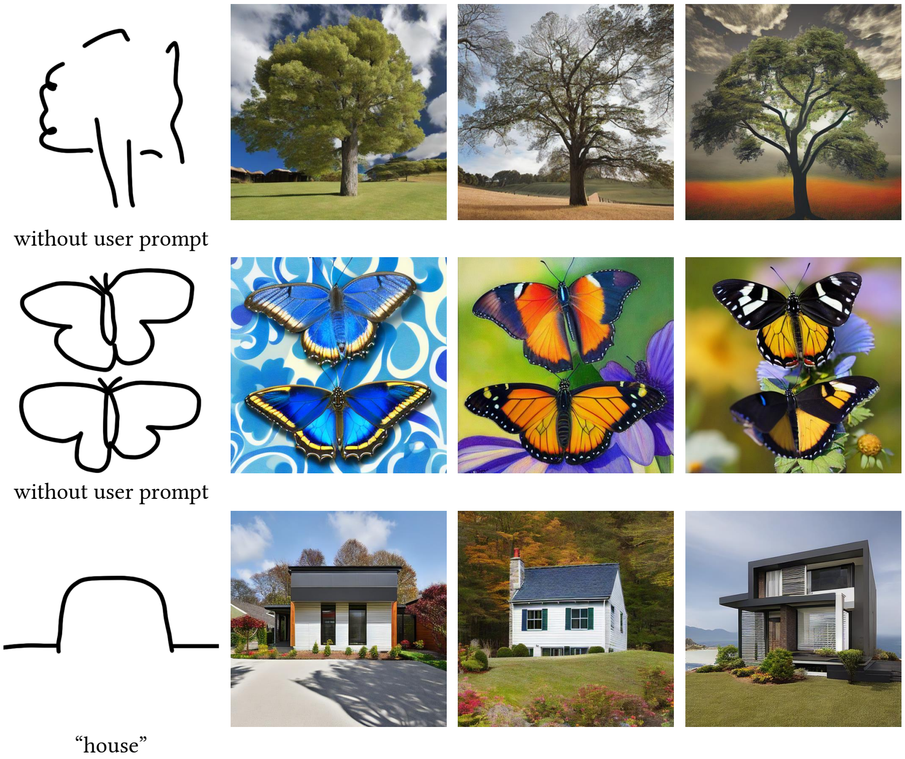


Input Canny edge

Output without prompt

“a watch in black color”

Figure 26: Example of simple object. When the diffusion content is relatively simple, the model can achieve very accurate control to manipulate the content materials.

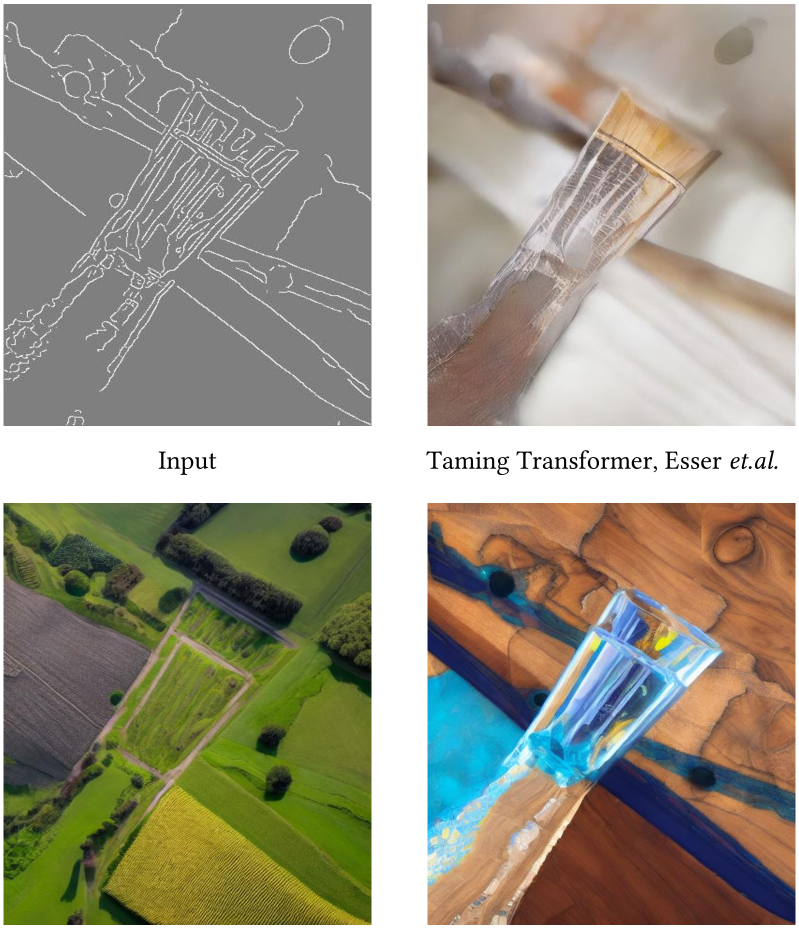


without user prompt

without user prompt

“house”

Figure 27: Coarse-level control. When users do not want their input shape to be preserved in the images, we can simply replace the last 50% diffusion iterations with standard SD without ControlNet. The resulting effect is similar to image retrieval but those images are generated.



Input

Taming Transformer, Esser *et.al.*

Ours default
(Seems to be interpreted as a bird's eye view of an agricultural field)

Ours “a glass of water”
(Seems unable to eliminate the effects of mistaken recognitions)

Figure 28: Limitation. When the semantic of input image is mistakenly recognized, the negative effects seem difficult to be eliminated, even if a strong prompt is provided.

[21] X. Xiang, D. Liu, X. Yang, Y. Zhu, and X. Shen. Anime2sketch: A sketch extractor for anime arts with deep networks, https://github.com/mukosame/anime2sketch, 2021.

[22] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403, 2015.

[23] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[24] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017.

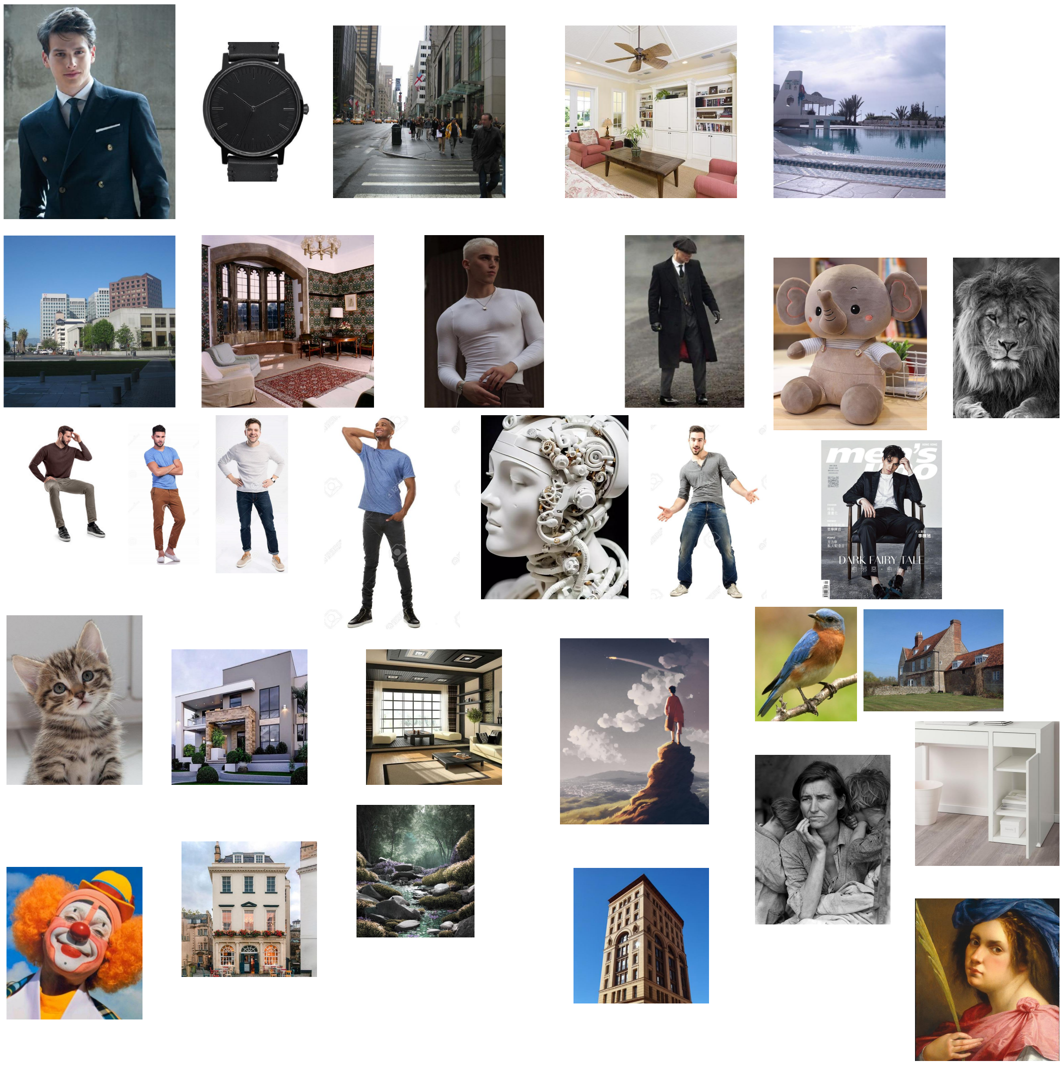


Figure 29: Appendix: all original source images for edge detection, semantic segmentation, pose extraction, *etc*. Note that some images may have copyrights.