Adding Conditional Control to Text-to-Image Diffusion Models

Zhang, L., Rao, A., & Agrawala, M. (2023)

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*

Presenter: Tran Van Nhan

Abstract

- ControlNet, a neural network architecture to add spatial conditioning controls to large, pretrained text-to-image diffusion models.
- Reuses their deep and robust encoding layers pretrained with billions of images to learn a diverse set of conditional controls.
- Connected with “zero convolutions” (zero-initialized convolution layers) that progressively grow the parameters from zero and ensure that no harmful noise could affect the finetuning.
- Test various conditioning controls, e.g., edges, depth, segmentation, human pose, etc.
- The training of ControlNets is robust with small (<50k) and large (>1m) datasets.
- Extensive results show that ControlNet may facilitate wider applications to control image diffusion models.

Introduction

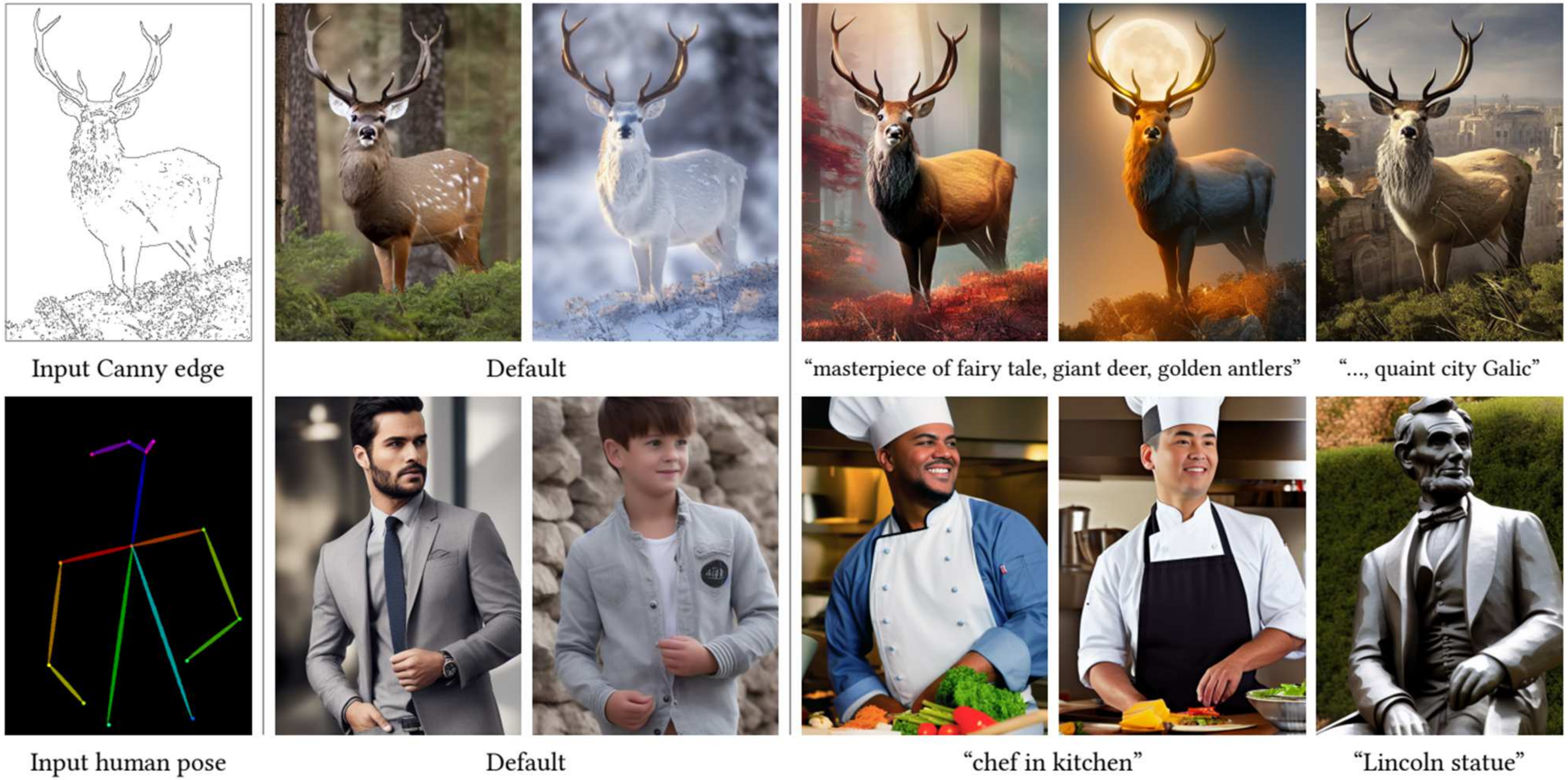
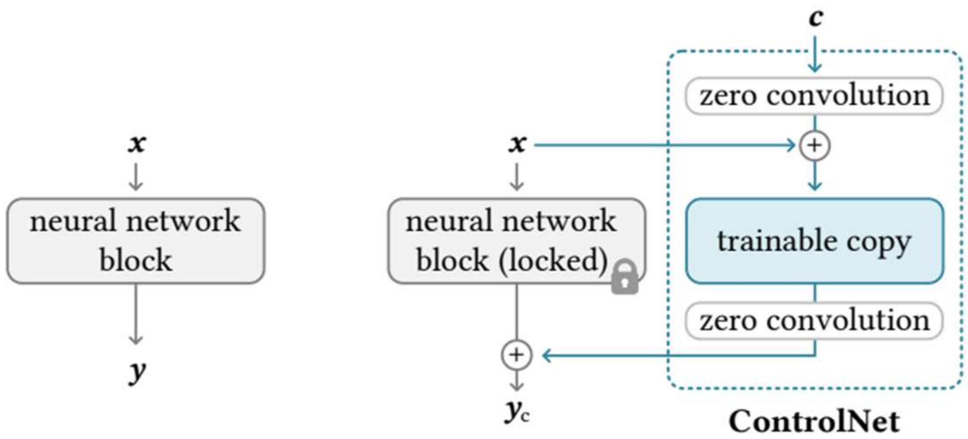


Figure 1: Controlling Stable Diffusion with learned conditions. ControlNet allows users to add conditions like Canny edges (top), human pose (bottom), *etc*., to control the image generation of large pretrained diffusion models. The default results use the prompt "a high-quality, detailed, and professional image". Users can optionally give prompts like the "chef in kitchen".

ControlNet

- ControlNet is a neural network architecture that can enhance large pretrained text-to-image diffusion models with spatially localized.
- ControlNet injects additional conditions into the blocks of a neural network.
- *network block* to refer to a set of neural layers that are commonly put together to form a single unit of a neural network.
- Suppose $F(\cdot;\Theta)$ is such a trained neural block, with parameters Θ, that transforms an input feature map ***x***, into another feature map ***y*** as

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}; \Theta). \quad (1)$$



(a) Before (b) After

Figure 2: A neural block takes a feature map x as input and outputs another feature map y, as shown in (a). To add a ControlNet to such a block we lock the original block and create a trainable copy and connect them together using zero convolution layers, *i.e.*, 1×1 convolution with both weight and bias initialized to zero. Here c is a conditioning vector that we wish to add to the network, as shown in (b).

ControlNet

- lock (freeze) the parameters Θ of the original block and simultaneously clone the block to a *trainable copy* with parameters Θc.
- Applied to large models Stable Diffusion, the locked parameters preserve the production-ready model trained with billions of images.
- The trainable copy is connected to the locked model with *zero convolution* layers, denoted *Z*(·; ·). To build up a ControlNet, we use two instances of zero convolutions with parameters Θz1 and Θz2 respectively. The complete ControlNet then computes

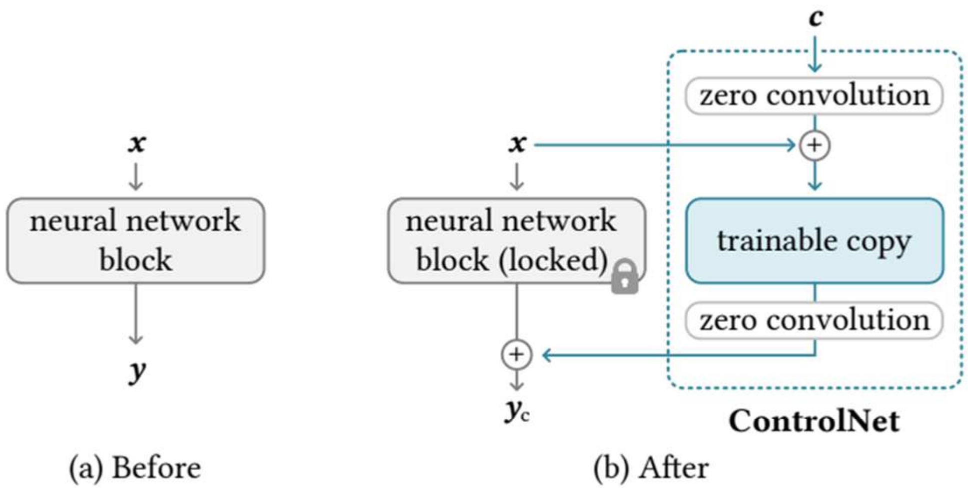
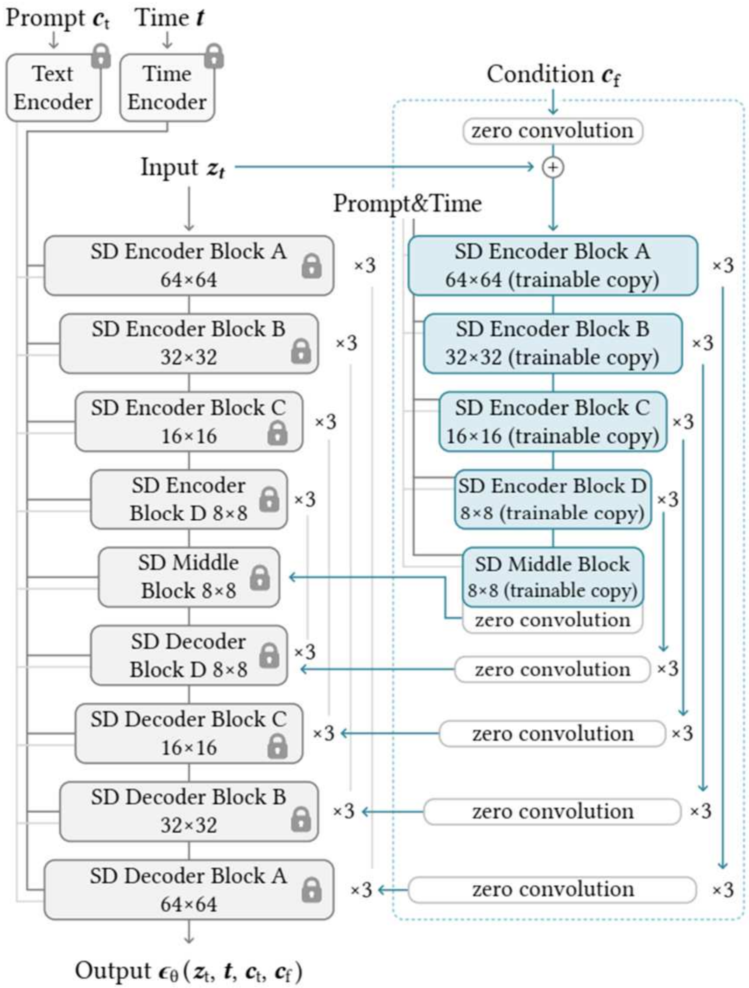


Figure 2: A neural block takes a feature map x as input and outputs another feature map y, as shown in (a). To add a ControlNet to such a block we lock the original block and create a trainable copy and connect them together using zero convolution layers, *i.e.*, 1×1 convolution with both weight and bias initialized to zero. Here c is a conditioning vector that we wish to add to the network, as shown in (b).

$$\boldsymbol{y}_c = \mathcal{F}(\boldsymbol{x}; \Theta) + \mathcal{Z}(\mathcal{F}(\boldsymbol{x} + \mathcal{Z}(\boldsymbol{c}; \Theta_{z1}); \Theta_c); \Theta_{z2}), \quad (2)$$

ControlNet for Text-to-Image Diffusion



(a) Stable Diffusion (b) ControlNet

Both the encoder and decoder contain 12 blocks, and the full model contains 25 blocks, including the middle block.

Of the 25 blocks, 8 blocks are down-sampling or up-sampling convolution layers, while the other 17 blocks are main blocks that each contain 4 resnet layers and 2 Vision Transformers (ViTs)

“SD Encoder Block A” contains 4 resnet layers and 2 ViTs

Figure 3: Stable Diffusion’s U-net architecture connected with a ControlNet on the encoder blocks and middle block. The locked, gray blocks show the structure of Stable Diffusion V1.5 (or V2.1, as they use the same U-net architecture). The trainable blue blocks and the white zero convolution layers are added to build a ControlNet.

Training

- Given an input image ***z***0, image diffusion algorithms progressively add noise to the image and produce a noisy image ***zt***, where *t* represents the number of times noise is added. Given a set of conditions including time step ***t***, text prompts ***ct***, as well as a task-specific condition ***c***f, image diffusion algorithms learn a network ϵ to predict the noise added to the noisy image ***zt*** with

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{z}_0, \boldsymbol{t}, \boldsymbol{c}_t, \boldsymbol{c}_\mathrm{f}, \epsilon \sim \mathcal{N}(0,1)} \Big[\| \epsilon - \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{t}, \boldsymbol{c}_t, \boldsymbol{c}_\mathrm{f})) \|_2^2 \Big],$$

- Where L is the overall learning objective of the entire diffusion model.
- Randomly replace 50% text prompts ct with empty strings. This approach increases ControlNet’s ability to directly recognize semantics

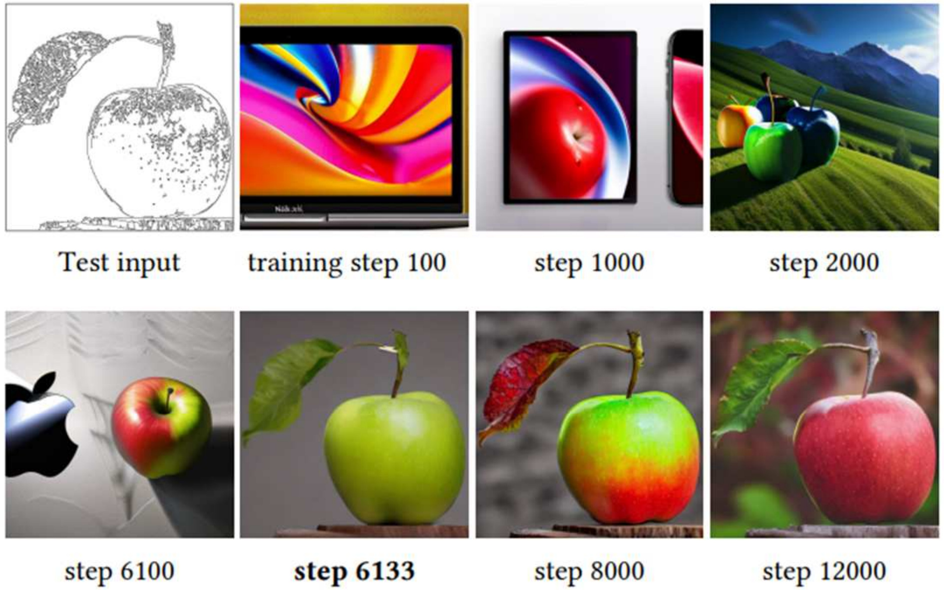


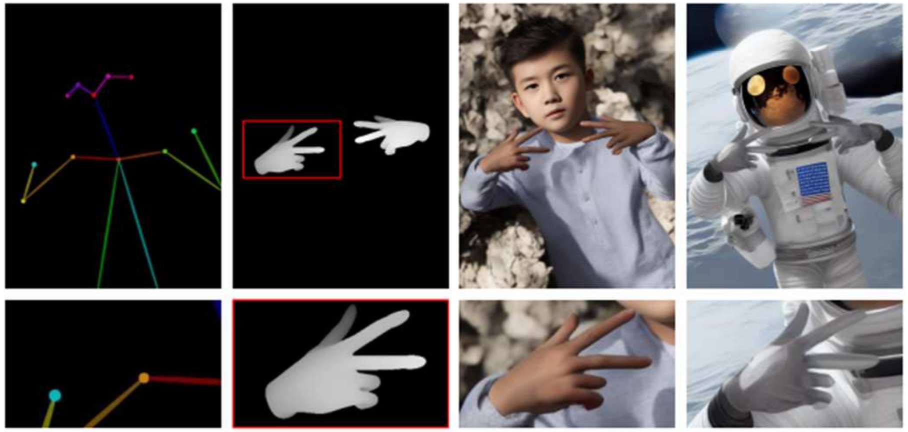
Figure 4: The sudden convergence phenomenon. Due to the zero convolutions, ControlNet always predicts high-quality images during the entire training. At a certain step in the training process (*e.g.*, the 6133 steps marked in bold), the model suddenly learns to follow the input condition.

Inference

- We can further control how the extra conditions of ControlNet affect the denoising diffusion process in several ways.
- **Classifier-free guidance resolution weighting.** CFG is formulated as $\epsilon_{prd} = \epsilon_{uc} + \beta_{cfg}(\epsilon_c - \epsilon_{uc})$ where ϵ_{prd}, ϵ_{uc}, ϵ_c, β_{cfg} are the model's final output, unconditional output, conditional output, and a user-specified weight respectively.
- **Composing multiple ControlNets.** To apply multiple conditioning images (*e.g.*, Canny edges, and pose) to a single instance of Stable Diffusion

(a) Input Canny map (b) W/o CFG (c) W/o CFG-RW (d) Full (w/o prompt)

Figure 5: Effect of Classifier-Free Guidance (CFG) and the proposed CFG Resolution Weighting (CFG-RW).



Multiple condition (pose&depth) "boy" "astronaut"

Figure 6: Composition of multiple conditions. We present the application to use depth and pose simultaneously.

Inference

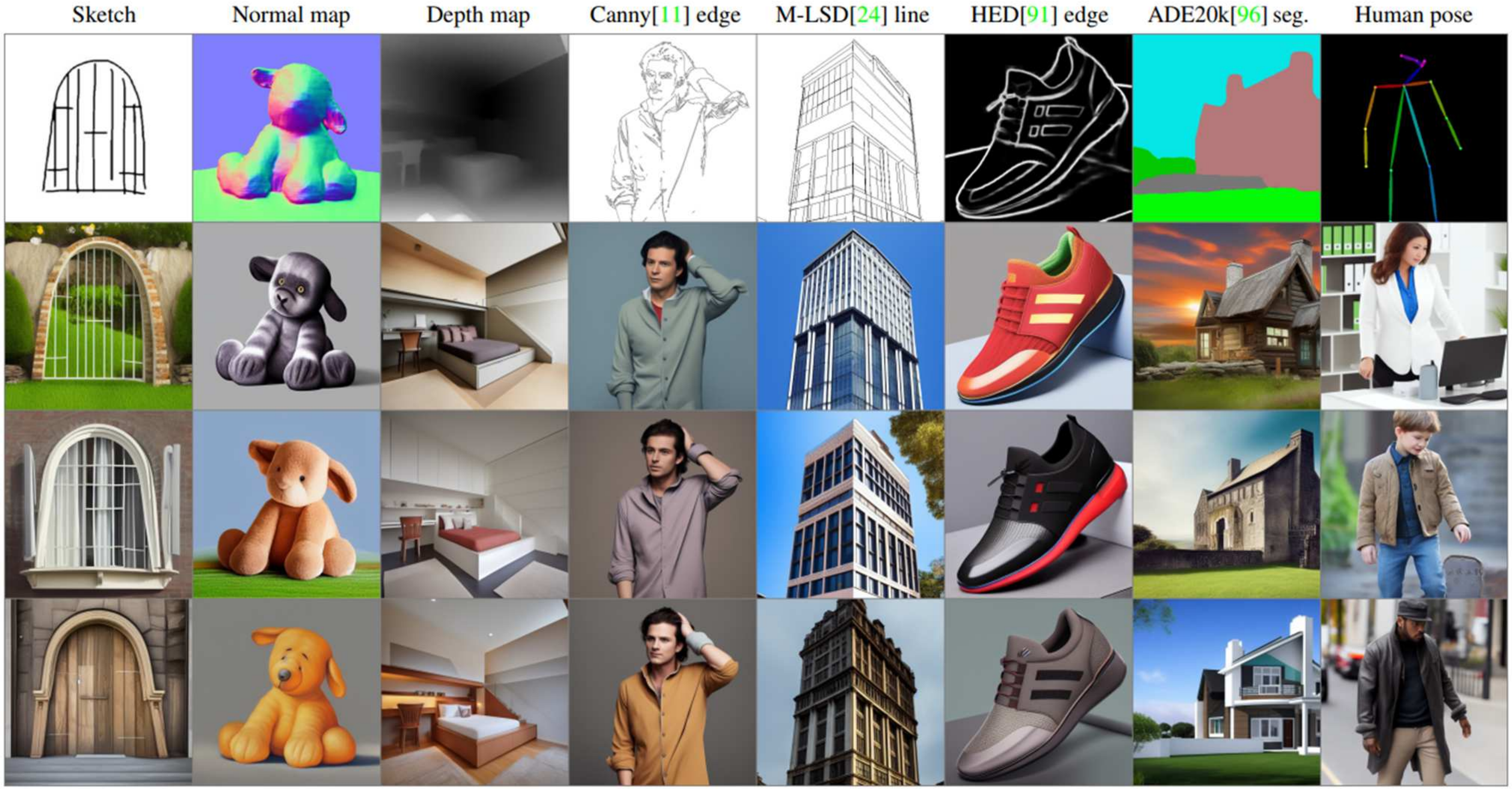


Figure 7: Controlling Stable Diffusion with various conditions **without prompts**. The top row is input conditions, while all other rows are outputs. We use the empty string as input prompts. All models are trained with general-domain data. The model has to recognize semantic contents in the input condition images to generate images.

Experiment

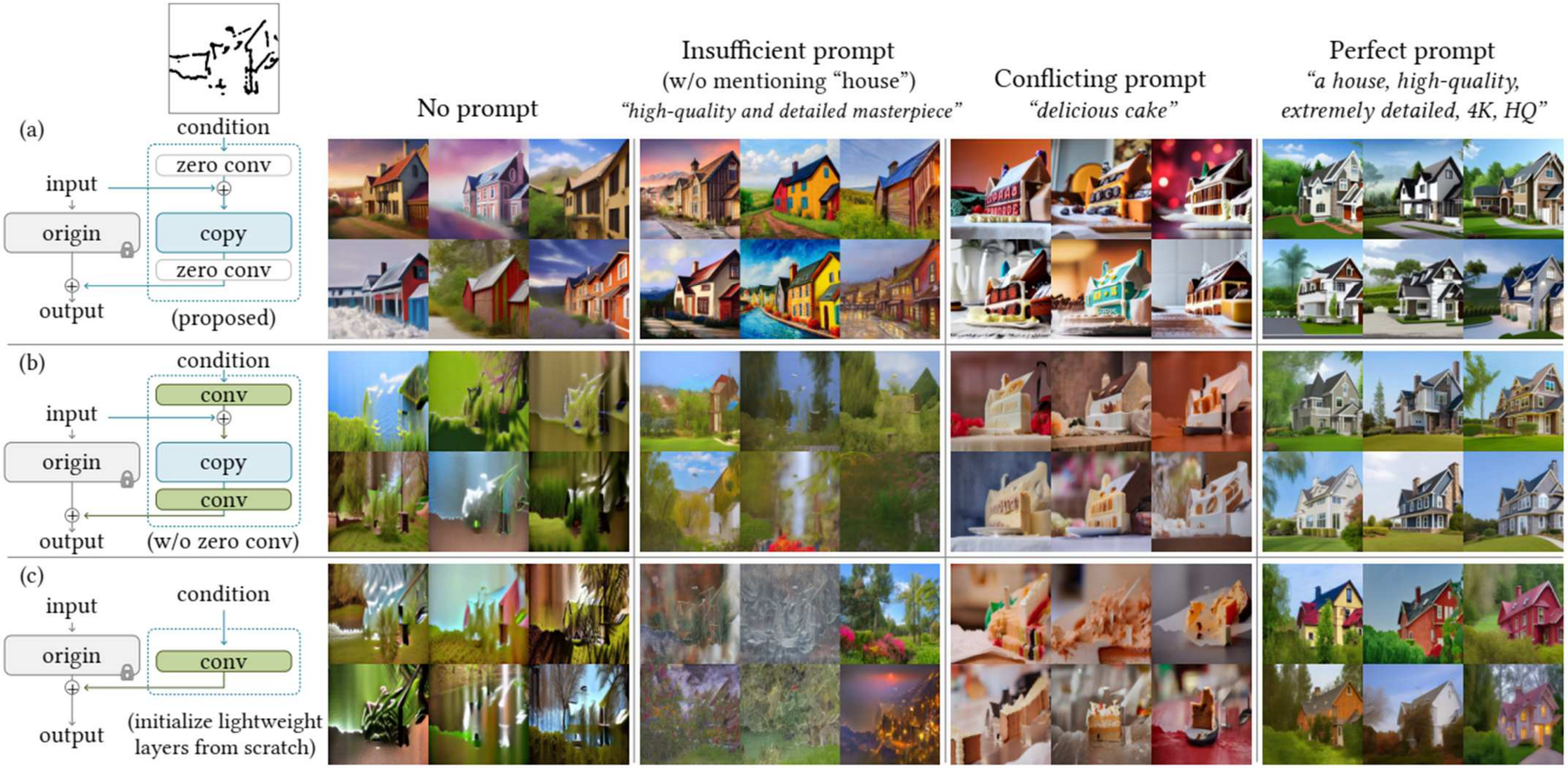


Figure 8: Ablative study of different architectures on a sketch condition and different prompt settings. For each setting, we show a random batch of 6 samples without cherry-picking. Images are at 512 × 512 and best viewed when zoomed in. The green "conv" blocks on the left are standard convolution layers initialized with Gaussian weights.

Experiment

Method	Result Quality ↑	Condition Fidelity ↑
PITI [89](sketch)	1.10 ± 0.05	1.02 ± 0.01
Sketch-Guided [88] ($\beta = 1.6$)	3.21 ± 0.62	2.31 ± 0.57
Sketch-Guided [88] ($\beta = 3.2$)	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 1: 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.

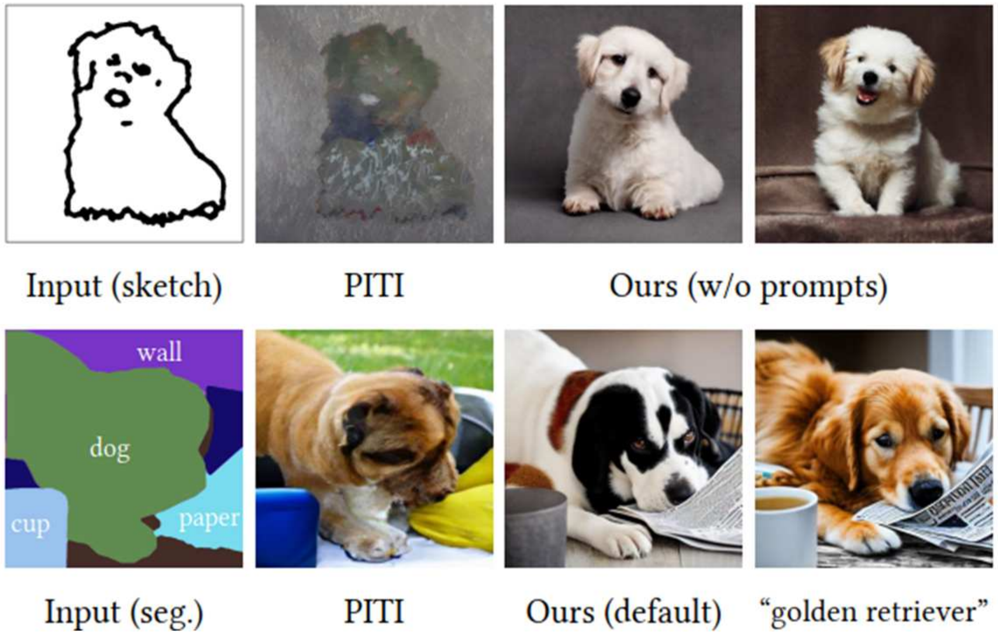
ADE20K (GT)	VQGAN [19]	LDM [72]	PITI [89]	ControlNet-lite	ControlNet
0.58 ± 0.10	0.21 ± 0.15	0.31 ± 0.09	0.26 ± 0.16	0.32 ± 0.12	$\mathbf{0.35 \pm 0.14}$

Table 2: Evaluation of semantic segmentation label reconstruction (ADE20K) with Intersection over Union (IoU ↑).

Method	FID ↓	CLIP-score ↑	CLIP-aes. ↑
Stable Diffusion	6.09	0.26	6.32
VQGAN [19](seg.)*	26.28	0.17	5.14
LDM [72](seg.)*	25.35	0.18	5.15
PITI [89](seg.)	19.74	0.20	5.77
ControlNet-lite	17.92	0.26	6.30
ControlNet	15.27	0.26	6.31

Table 3: 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.

Experiment



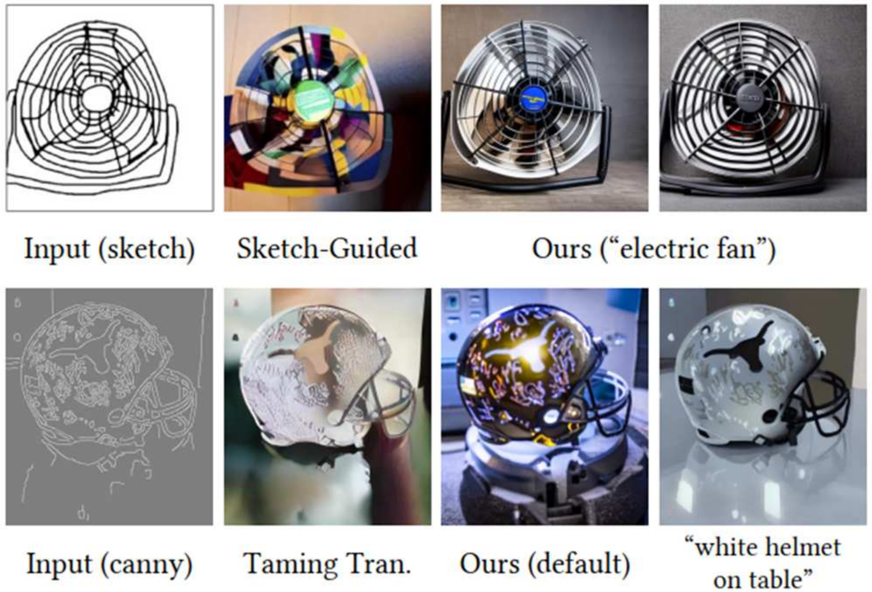


Figure 9: Comparison to previous methods. We present the qualitative comparisons to PITI [89], Sketch-Guided Diffusion [88], and Taming Transformers [19].

Experiment

Figure 10: The influence of different training dataset sizes. See also the supplementary material for extended examples.

Figure 11: Interpreting contents. If the input is ambiguous and the user does not mention object contents in prompts, the results look like the model tries to interpret input shapes.

Experiment

Figure 12: Transfer pretrained ControlNets to community models [16, 61] without training the neural networks again.

THANK YOU

Question & Answer