ControlNet

by Klyuchnikova Ulyana

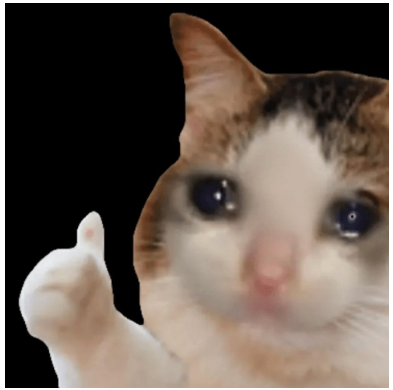
Motivation

Let’s say we have a text-to-image diffusion model but we want to add **conditional controls**

“cat crying”

“cat crying”

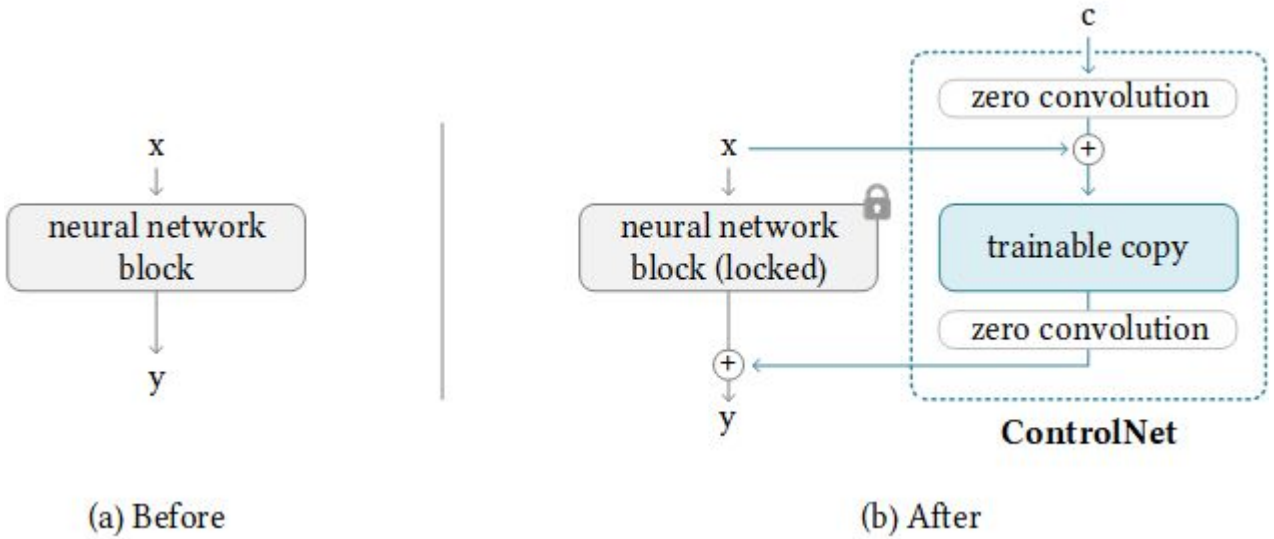
+



Generally speaking...

- control the final image generation through various techniques like pose, edge detection, depth maps, etc
- end-to-end architecture
- robust on small datasets
- as fast as fine-tuning but better
- can scale to large amount of data
- does not change the initial network
- allows to train a “home-made” model

Idea



1. Copy model weights to locked network and trainable copy
2. Add zero convolutions - those are basically Conv 1x1 initialized by 0, so at first iteration control net has zero input

In formulas

x - input like $\mathbb{R}^{h \times w \times c}$

y - output

c - conditional vector

Θ - original parameters

Θc - trainable copy

Z - zero 1x1 conv

Was: $$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}; \Theta)$$

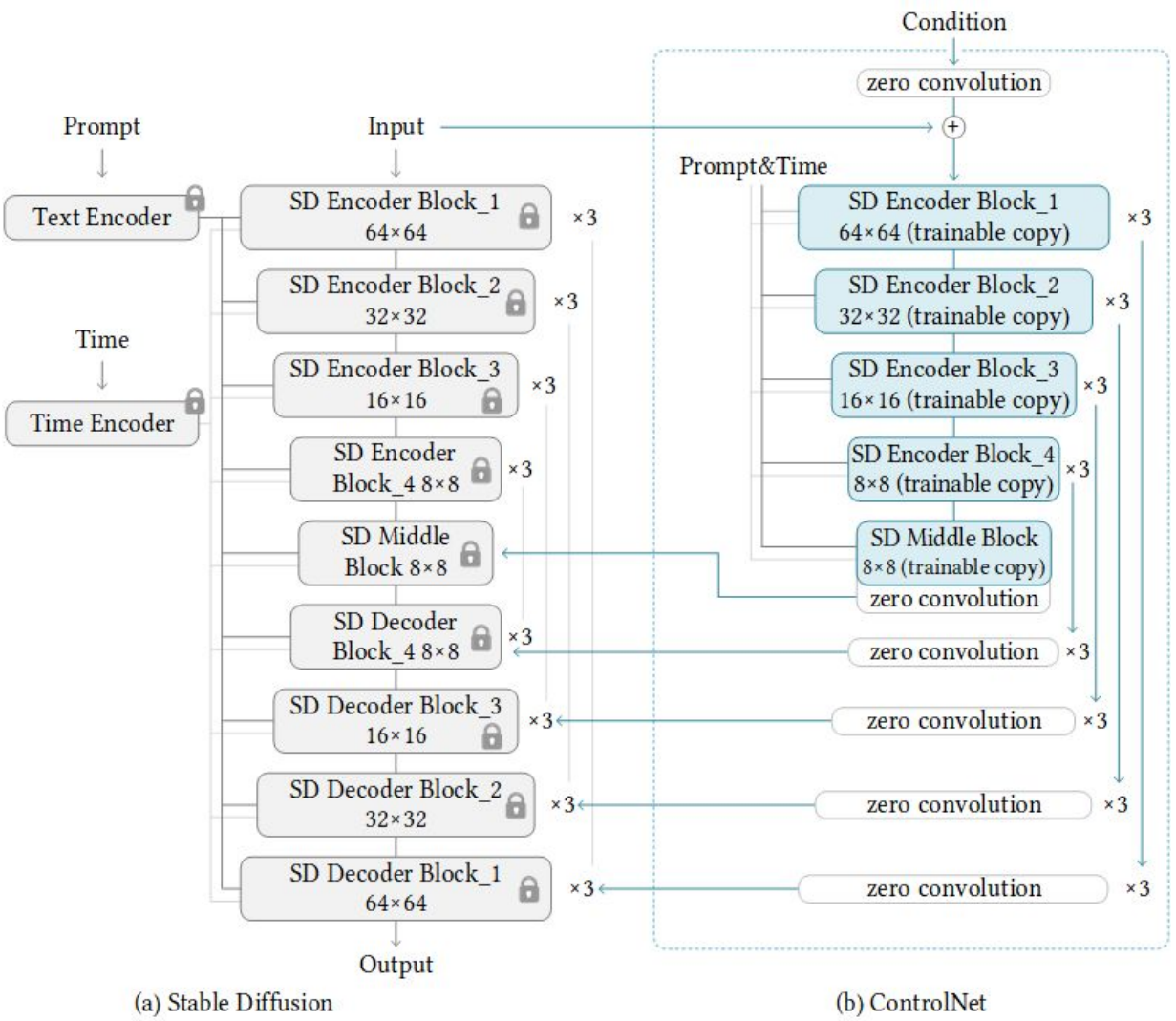
Became: $$\boldsymbol{y}_\mathrm{c} = \mathcal{F}(\boldsymbol{x}; \Theta) + \mathcal{Z}(\mathcal{F}(\boldsymbol{x} + \mathcal{Z}(\boldsymbol{c}; \Theta_{\mathrm{z1}}); \Theta_\mathrm{c}); \Theta_{\mathrm{z2}})$$

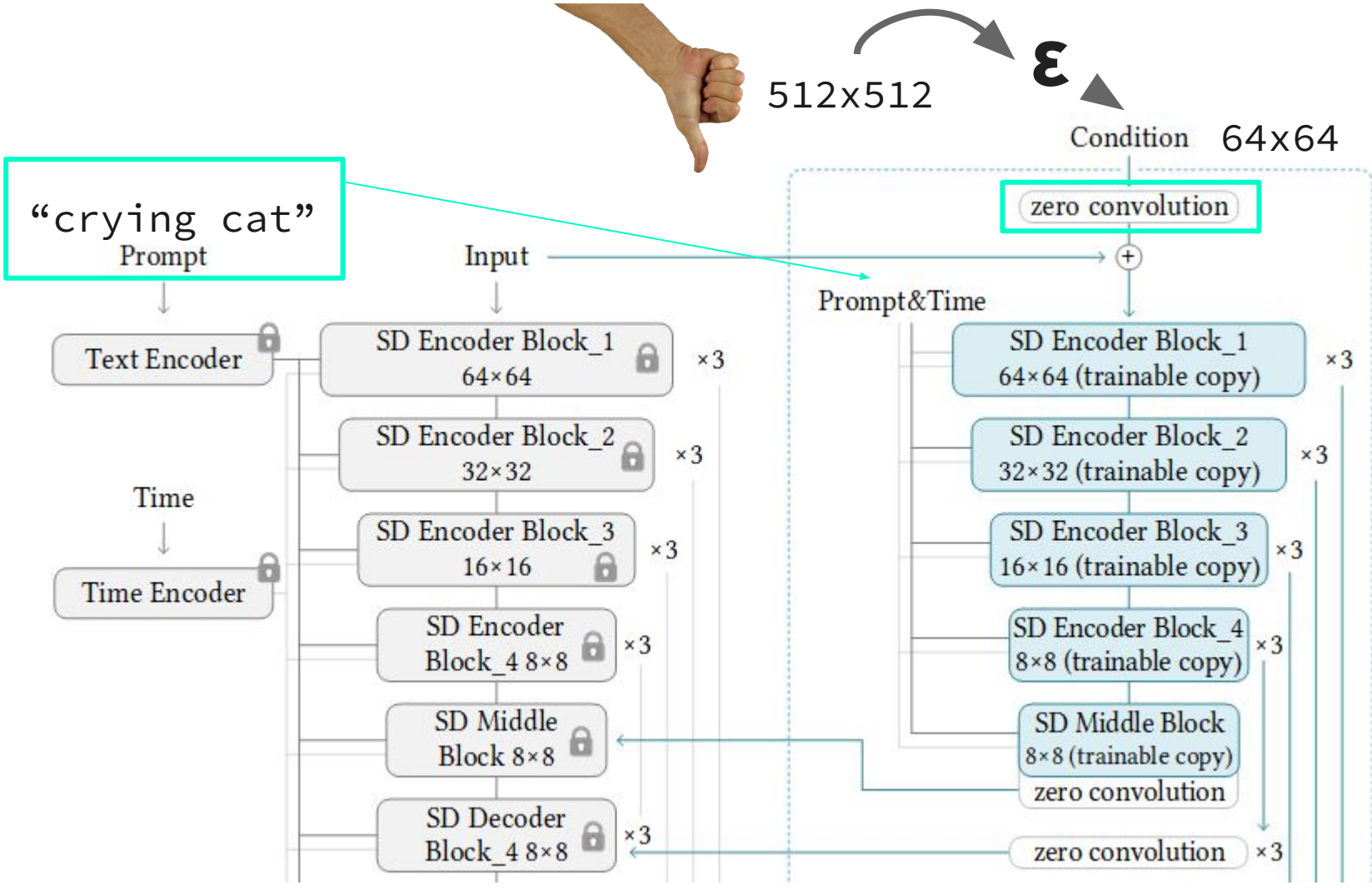
Text2Image

Stable Diffusion is essentially a U-Net

12 encoder blocks
+ 1 middle
+ 12 decoder

images as well as condition are originally 512x512 but processed into 64x64




512x512
ε
Condition
64x64
"crying cat"
Prompt
Input
zero convolution
Prompt&Time
Text Encoder
SD Encoder Block_1
64×64
×3
SD Encoder Block_1
64×64 (trainable copy)
×3
SD Encoder Block_2
32×32
×3
SD Encoder Block_2
32×32 (trainable copy)
×3
Time
SD Encoder Block_3
16×16
×3
SD Encoder Block_3
16×16 (trainable copy)
×3
Time Encoder
SD Encoder
Block_4 8×8
×3
SD Encoder Block_4
8×8 (trainable copy)
×3
SD Middle
Block 8×8
SD Middle Block
8×8 (trainable copy)
zero convolution
SD Decoder
Block_4 8×8
×3
zero convolution
×3

Training details

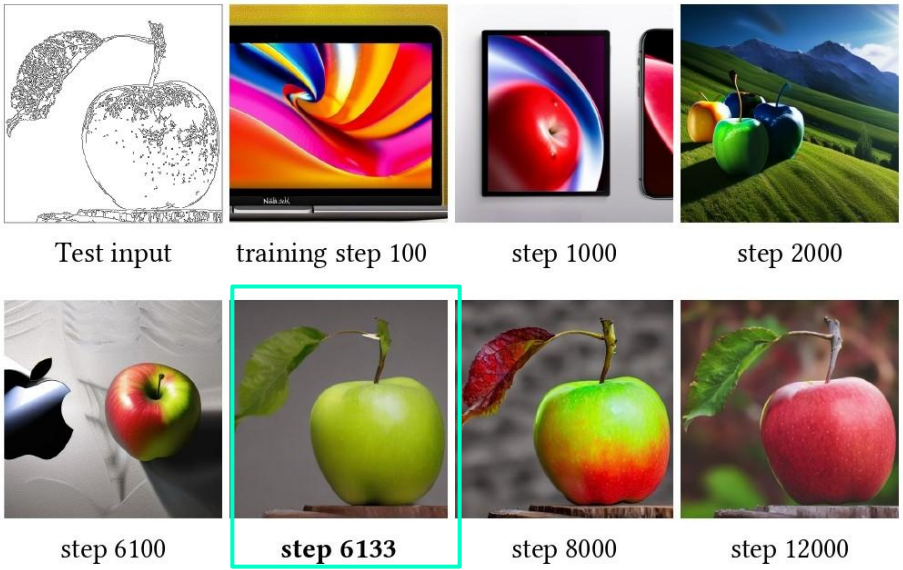
noisy image

condition

network

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

- 50% of prompts are replaced by empty strings to encourage control net to learn information from condition
- sudden convergence phenomenon



Classifier-free guidance

final output

conditional output

unconditional output

$$\epsilon_{prd} = \epsilon_{uc} + \beta_{cfg}(\epsilon_c - \epsilon_{uc})$$

c

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

CFG Resolution Weighting

- add conditioning to ec
- multiply connection weights by w = 64/h where h is the size of the block (8, 16, …, 64)

Experiments

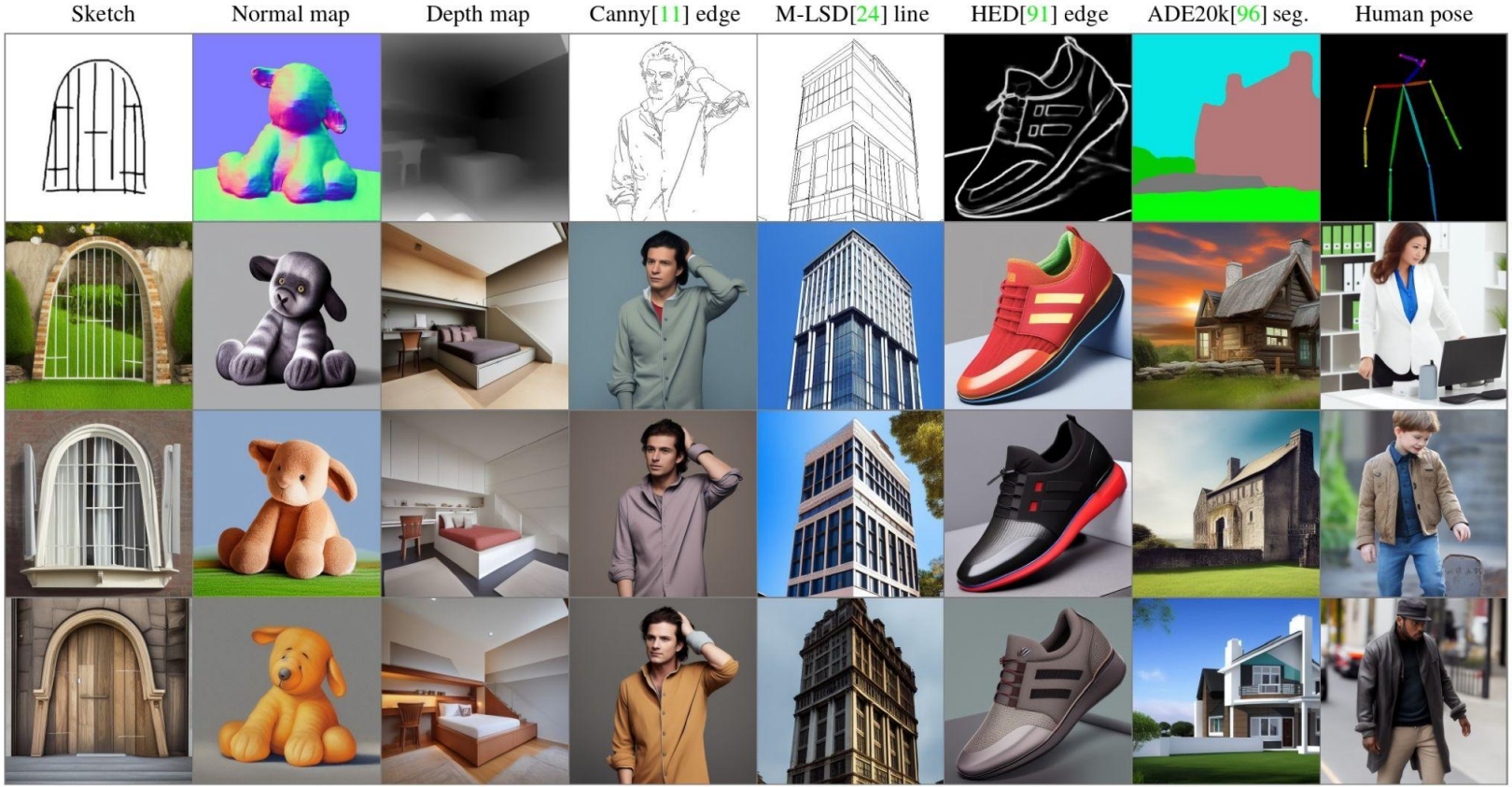
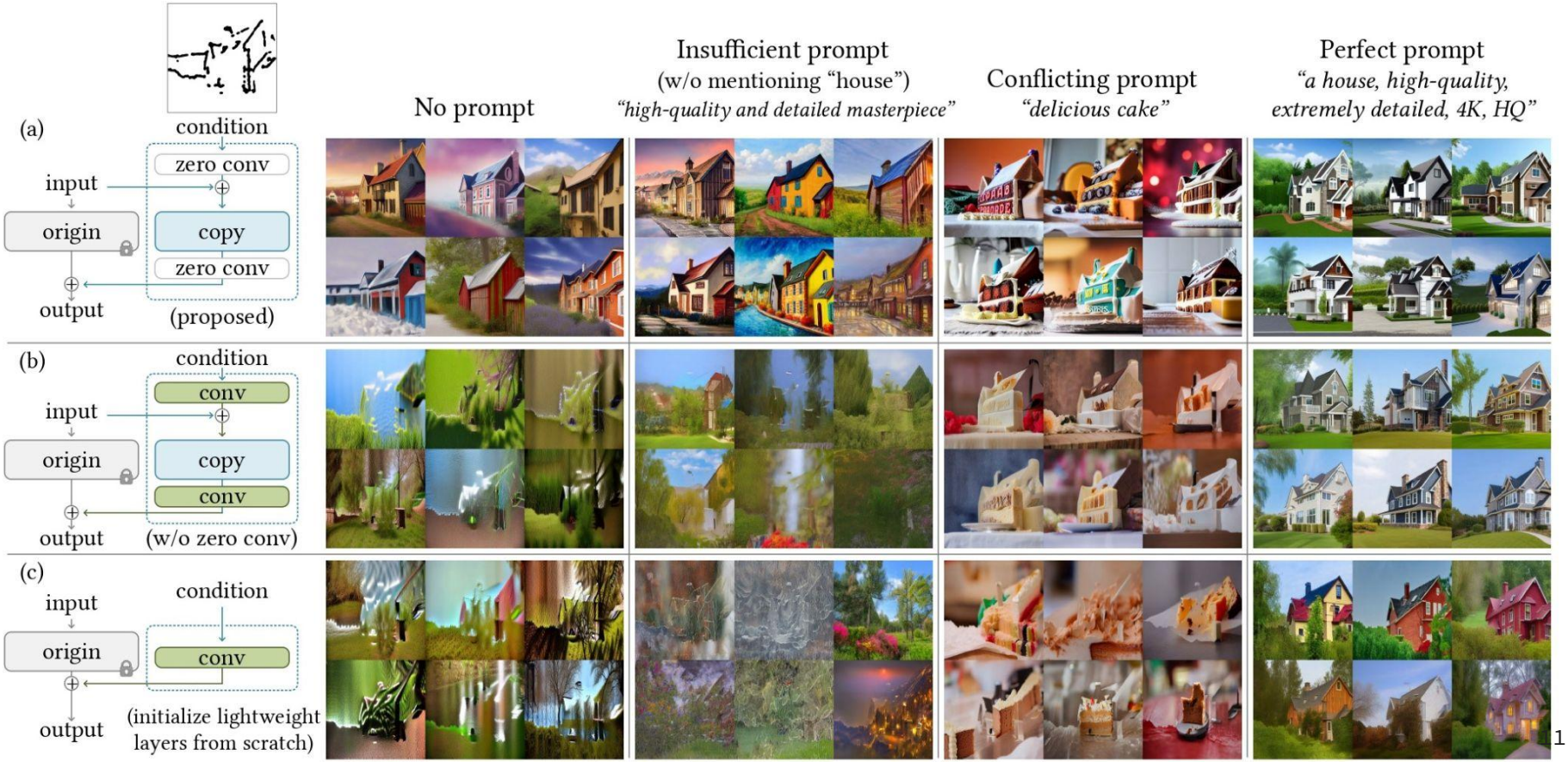


Figure 7: Controlling Stable Diffusion with various conditions **without prompts**. The top row is input conditions, while all other rows are outputs. We use the empty string as input prompts. All models are trained with general-domain data. The model has to recognize semantic contents in the input condition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

Ablative study



Evaluation of results

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.

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.

Input (sketch) PITI Ours (w/o prompts)

Input (seg.) PITI Ours (default) “golden retriever”

Input (sketch) Sketch-Guided Ours (“electric fan”)

Input (canny) Taming Tran. Ours (default) “white helmet on table”

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

“Lion” 1k images 50k images 3m images

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

Input “a high-quality and extremely detailed image”

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.

“house” SD 1.5 Comic Diffusion Protogen 3.4

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

Useful links

paper: https://arxiv.org/pdf/2302.05543

paper2: https://arxiv.org/pdf/2302.05543v1

video: https://youtu.be/WgrmCVa35ws?feature=shared

hugging face: https://huggingface.co/spaces/hysts/ControlNet

github: https://github.com/lllyasviel/ControlNet

how to run in google collab:
https://www.youtube.com/watch?v=Uq9N0nqUYqc

Thanks for your attention!