Adding Conditional Control to Text-to-Image Diffusion Models

Зайцев Федор БПМИ213

Проблема

- Диффузионным моделям удобно передавать на вход абстрактные текстовые описания
- Неудобно задавать конкретные позы, формы или сложные композиции таргетируемого изображения

Решение

- Файнтюним предобученную диффузионную text-to-image модель, добавив к ней пространственные входные условия

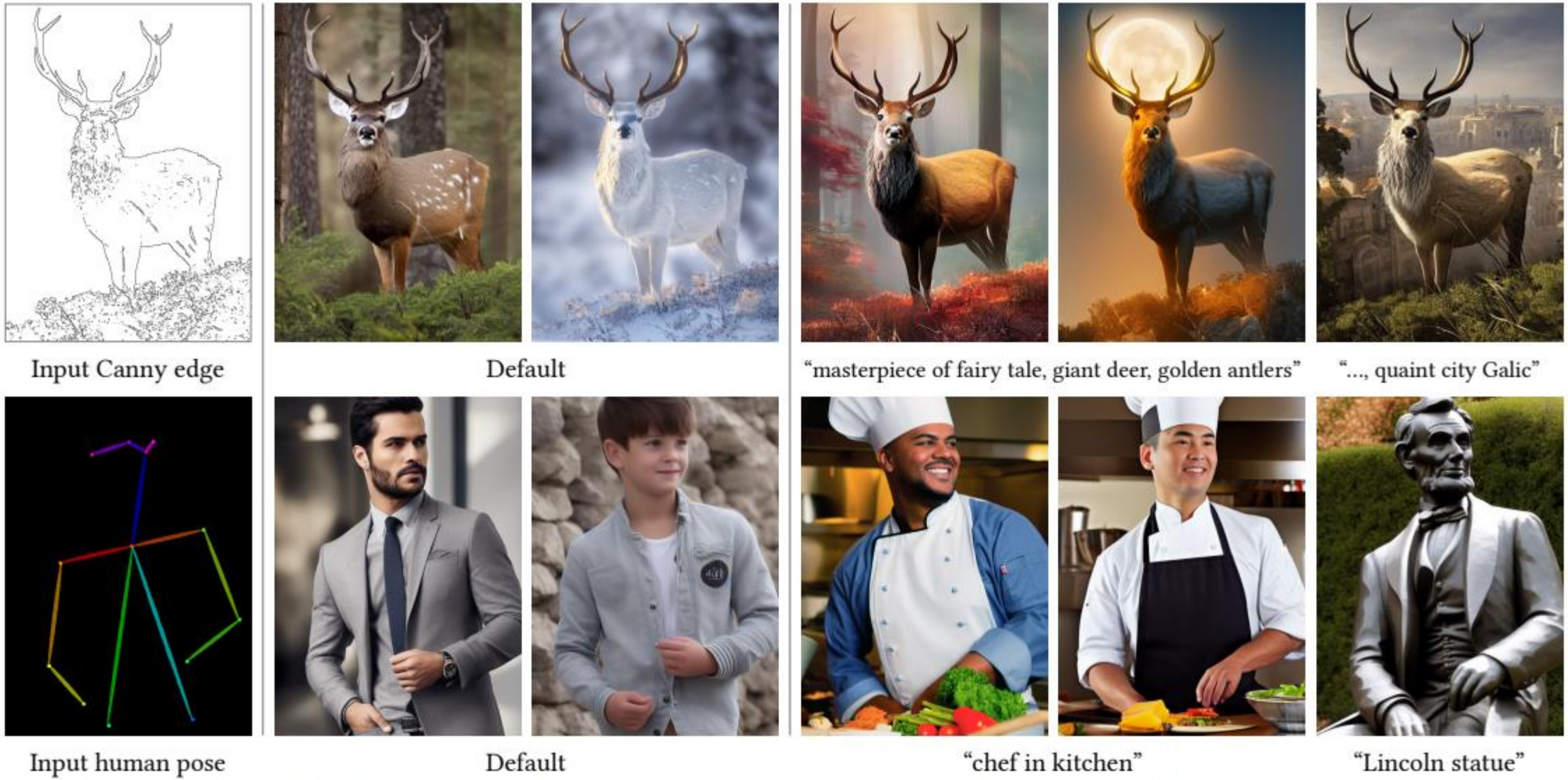
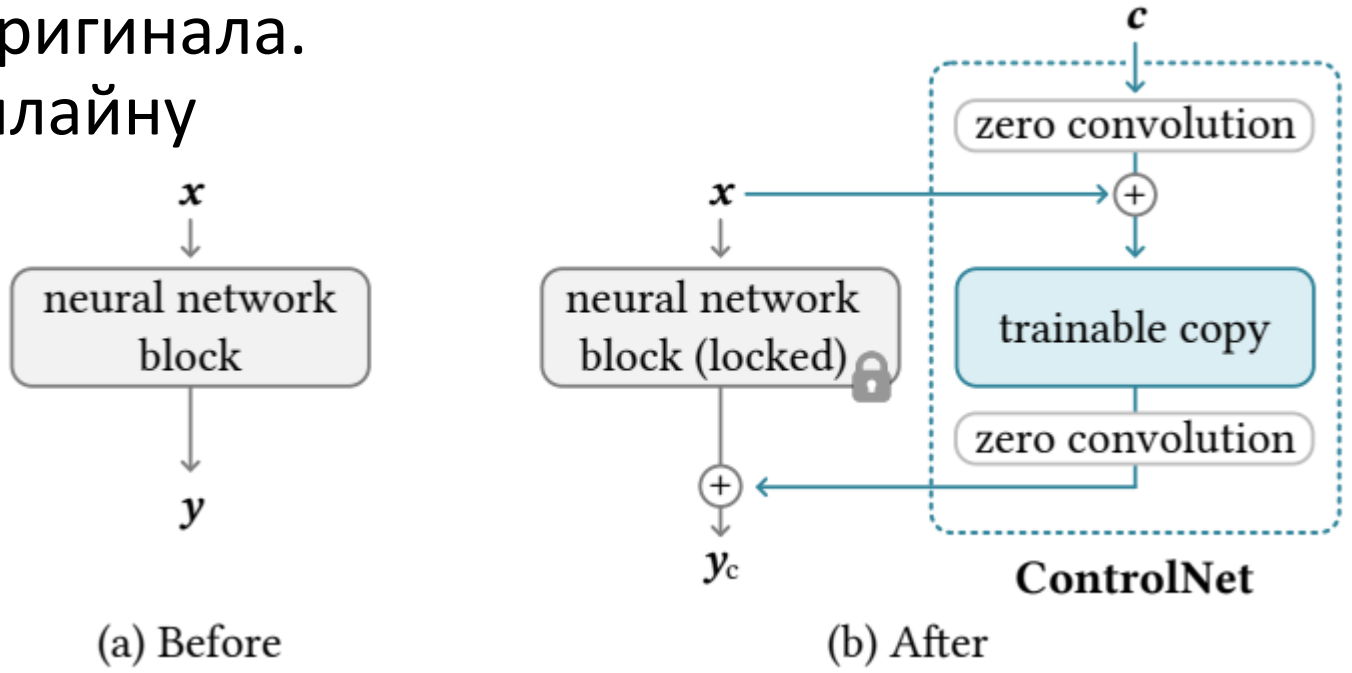


Figure 1: Controlling Stable Diffusion with learned conditions. ControlNet allows users to add conditions like Canny edges (top), human pose (bottom), *etc*., to control the image generation of large pretrained diffusion models. The default results use the prompt "a high-quality, detailed, and professional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

Файнтюнинг

- Обычный файнтюнинг на маленьком датасете может привести к катастрофическому забыванию или переобучению
- ControlNet копирует блок нейросети и добавляет дополнительные условия, замораживая веса оригинала.
  Копия присоединяется к пайплайну нулевыми свертками 1x1
- Нулевые свертки препятствуют зашумлению аутпута на ранних итерациях

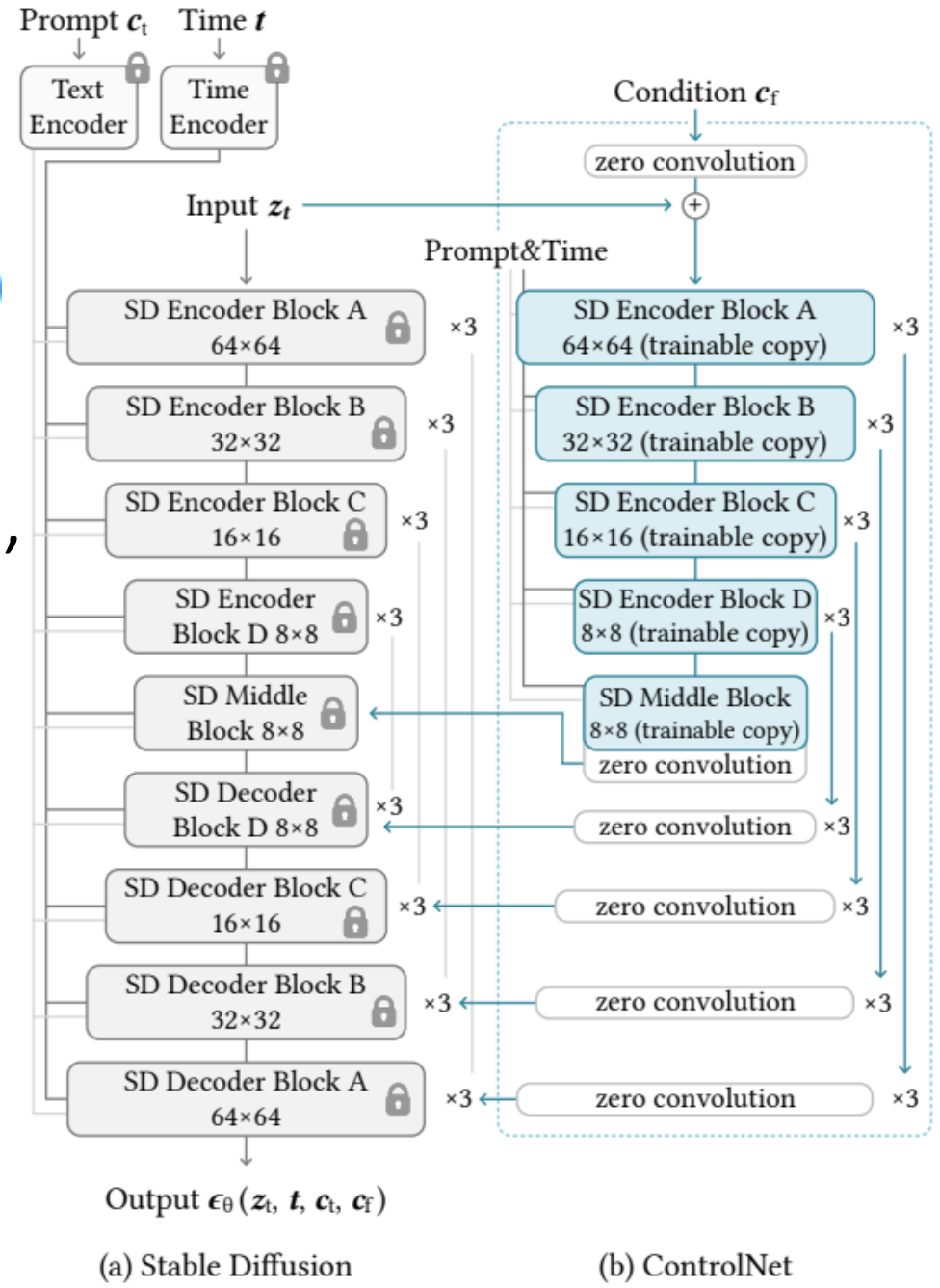


$$\boldsymbol{y}_c = \mathcal{F}(\boldsymbol{x}; \Theta) + \mathcal{Z}(\mathcal{F}(\boldsymbol{x} + \mathcal{Z}(\boldsymbol{c}; \Theta_{z1}); \Theta_c); \Theta_{z2})$$

Архитектура

- Векторизуем картинку-условие $c_f = \mathcal{E}(c_i)$
  c_i имеет размерность 512x512
  c_f имеет размерность 64x64 (как в SD)
$\mathcal{E}(\cdot)$ - 4 свертки с ядром 4x4 и страйдом 2x2, ReLU активация, каналов 16, 32, 64, 128

- На шаге оптимизации мы не считаем градиенты у оригинального SD
  → вычислительная сложность сравнима с обычным SD
  (+23% GPU memory, +34%time)



Обучение

- z_0 - исходное изображение
  z_t - зашумленное изображение
  t - количество итераций зашумления
  c_t - текстовые признаки
  c_f - пространственные признаки
  ϵ_θ - модель, предсказывающая добавленный шум

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

- При обучении рандомно выкидываем 50% текстовых промптов
- Модель резко «выучивает» заданное пространственное условие на одной из итераций (чаще всего до 10к)

Test input

training step 100

step 1000

step 2000

step 6100

step 6133

step 8000

step 12000

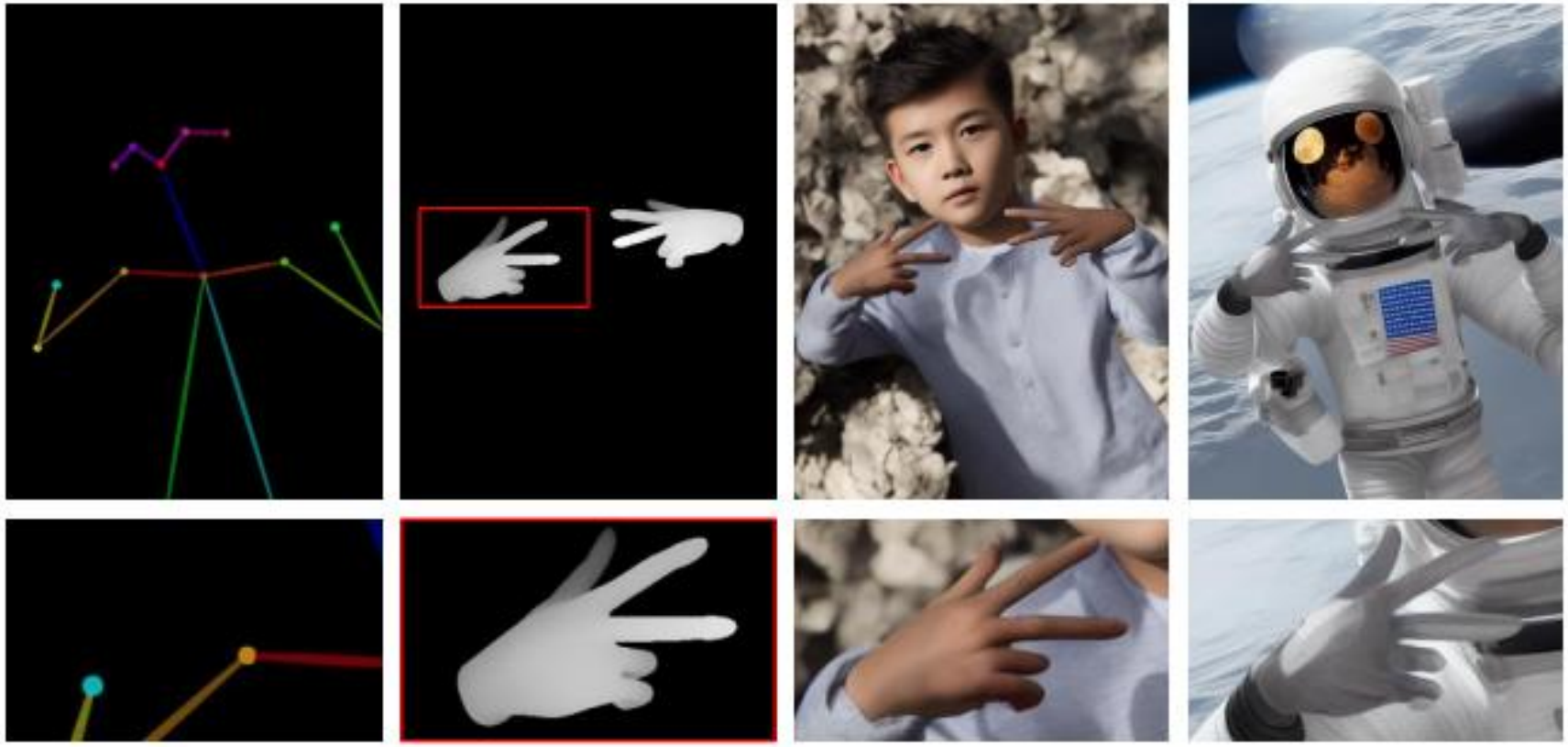
Инференс: Classifier-free guidance

- SD использует CFG: $\epsilon_{\text{prd}} = \epsilon_{\text{uc}} + \beta_{\text{cfg}}(\epsilon_{\text{c}} - \epsilon_{\text{uc}})$
- Куда отнести пространственное условие?
- Добавим его в ϵ_{c}, но умножим все входы из CN в SD на вес, обратно пропорциональный разрешению блока

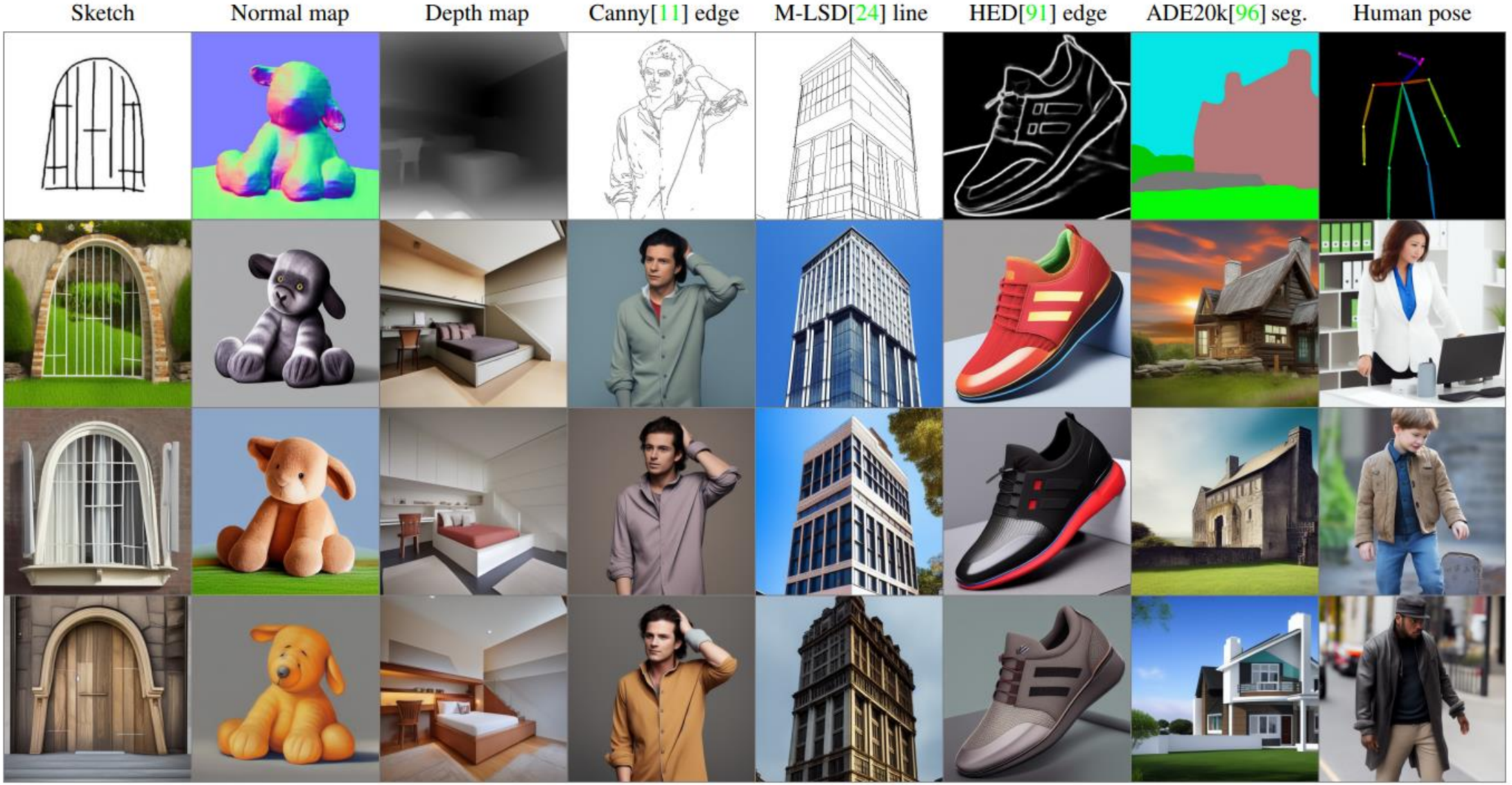
(a) Input Canny map (b) W/o CFG (c) W/o CFG-RW (d) Full (w/o prompt)

Инференс: комбинация условий

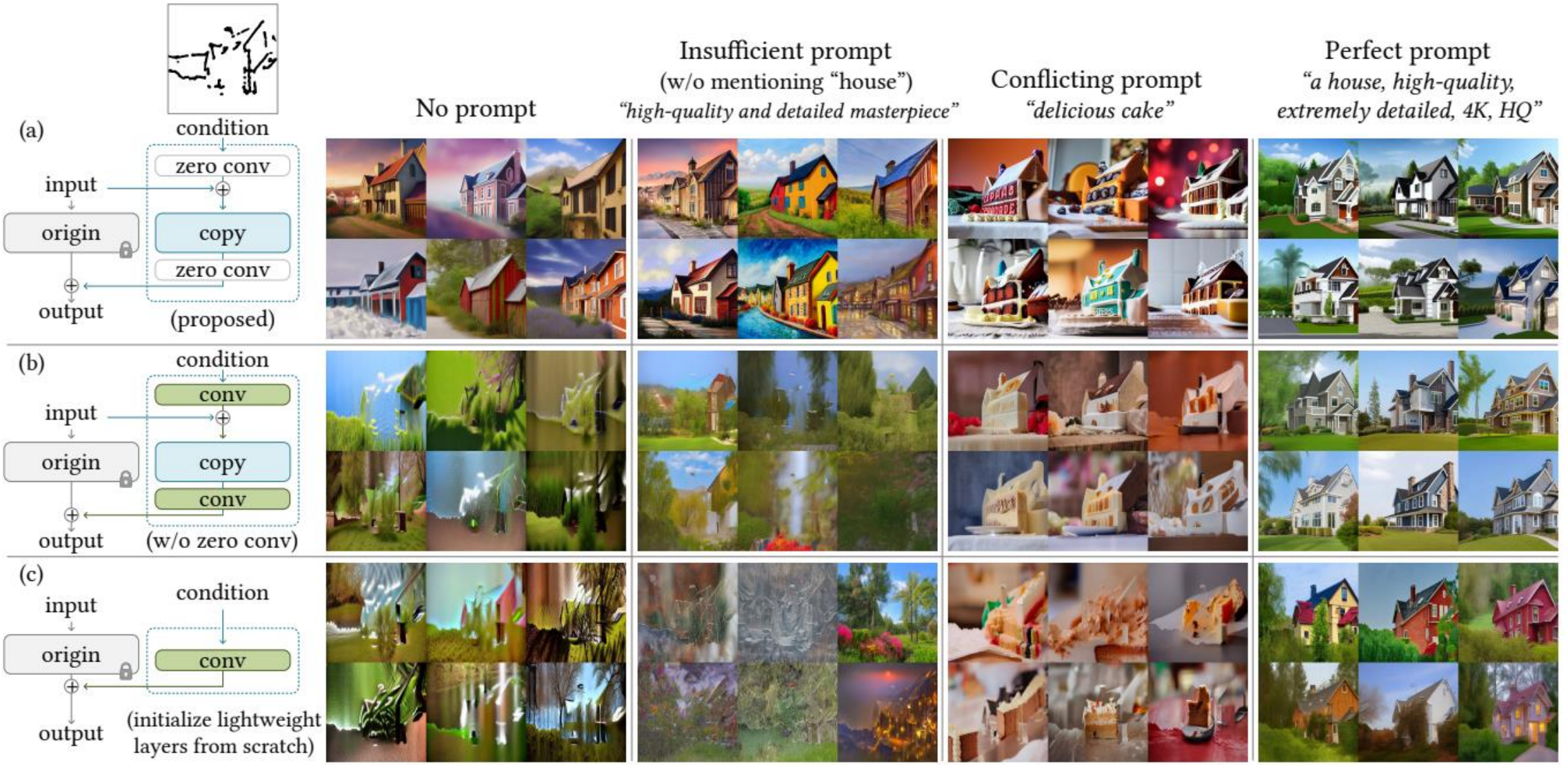
- Чтобы скомбинировать несколько условий просто складываем выходы всех соответствующих им ControlNet с SD вместо одного



Реализованные модели



Эксперименты

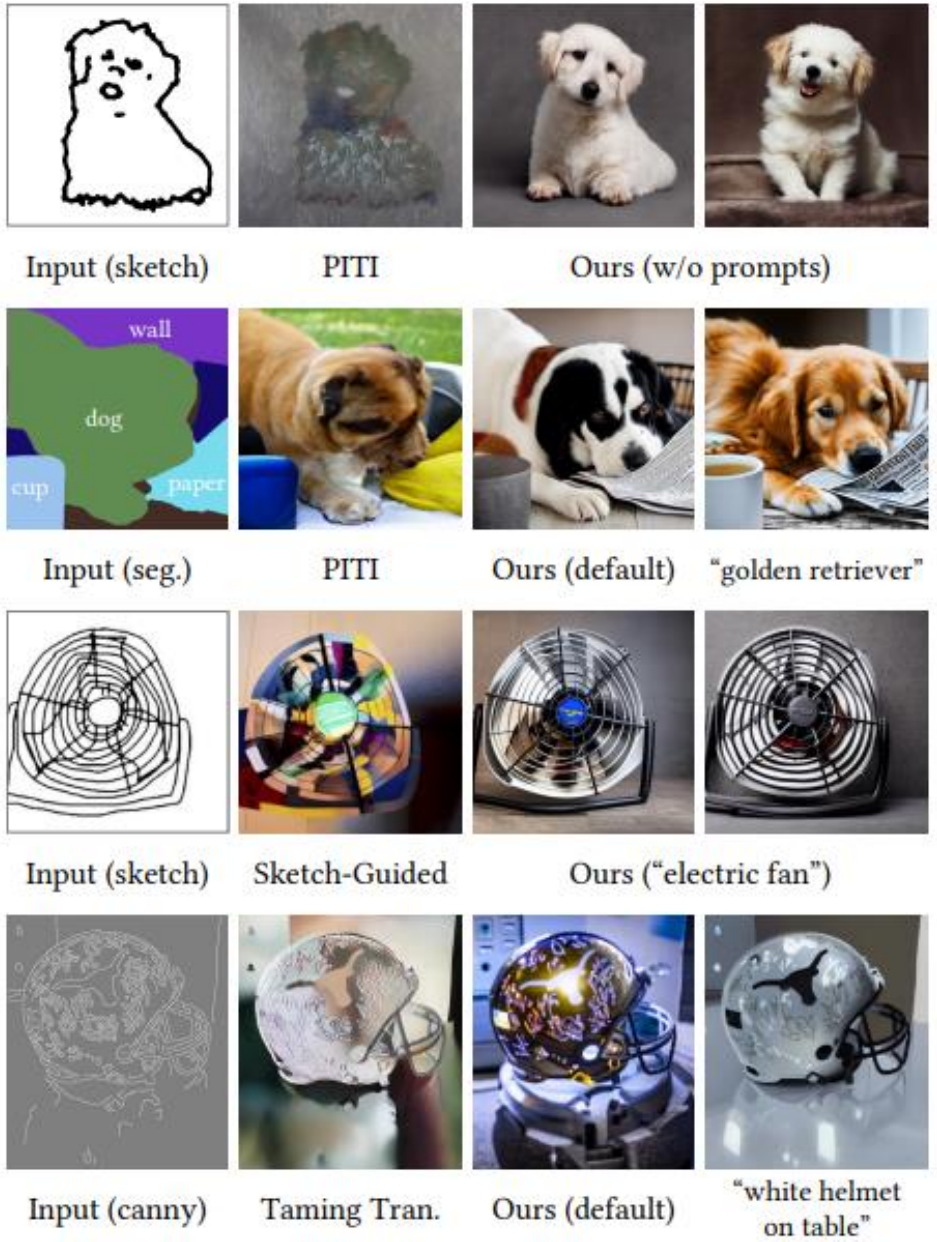


Результаты

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	$\mathbf{4.22 \pm 0.43}$	$\mathbf{4.28 \pm 0.45}$

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	$\mathbf{0.35 \pm 0.14}$

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



Результаты

Figure 10: The influence of different training dataset sizes.

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.

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

Заключение

- ControlNet – архитектура, позволяющая добавлять к большим предобученным Stable Diffusion моделям пространственные условия
- ControlNet обучается в ходе файнтюнинга копий блоков SD модели, в то же время храня в себе замороженный оригинал и используя его в качестве основы для генерации изображений
- ControlNet дает устойчивый результат при обучении как на маленьких, так и на больших датасетах
- ControlNet достигает SOTA результатов среди аналогов, сохраняя при этом качество изображения SD

Спасибо за внимание!

- Источники - https://arxiv.org/abs/2302.05543

- Еще прикольные картинки - https://journal.tinkoff.ru/controlnet/