Adding Conditional Control to Text-to-Image Diffusion Models

Сидоров Дмитрий

О чем этот доклад

Рисунок 1: Примеры работы ControlNet

План

- Напоминание про Latent Diffusion Models и Stable Diffusion
- Какие недостатки текущих подходов решает ControlNet
- Как работает ControlNet
- Как происходит обучение ControlNet
- Результаты

Latent Diffusion Models

1. Encoder кодирует изображение в латентное пространство меньшей размерности
2. Диффузионная модель работает в латентном пространстве
3. Диффузионная модель обучается, начиная с шума и постепенно превращая его в осмысленные латентные коды
4. Decoder преобразует латентные представления обратно в изображения

Latent Diffusion Models

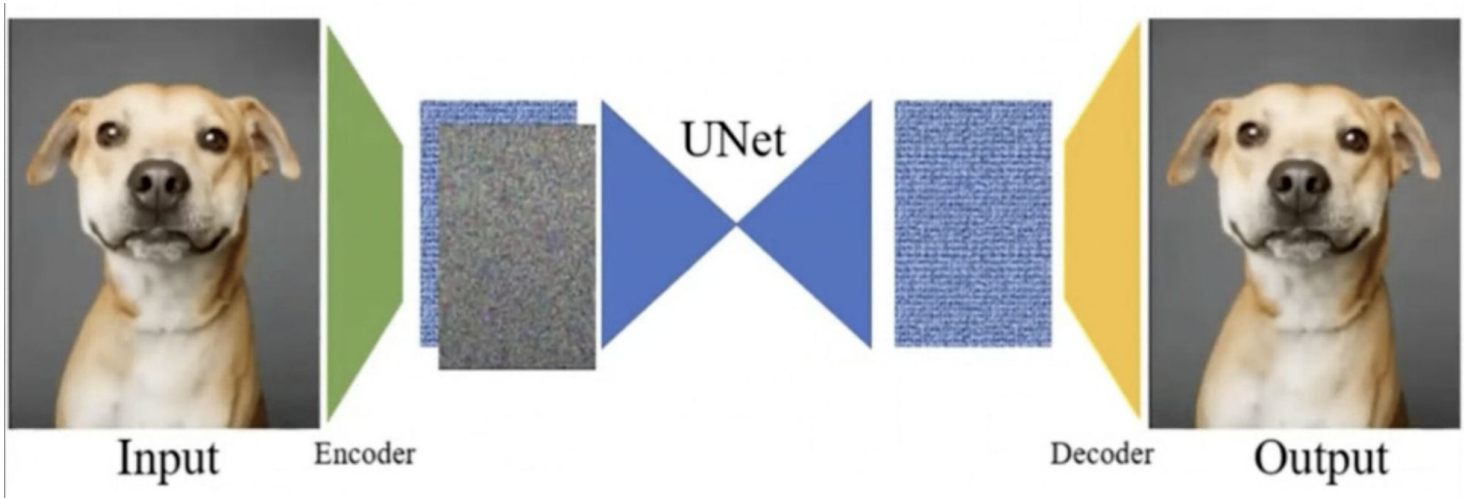

Рисунок 2: Схема работы LDM

Stable Diffusion

- Начинает с чистого изображения, к которому добавляется гауссовский шум
- Каждый шаг добавления шума зависит только от предыдущего состояния (марковский процесс)
- Для восстановления изображения используется обученная нейронная сеть, которая предсказывает и удаляет добавленный шум на каждом шаге
- На каждом шаге восстановления применяется корректировка

Stable Diffusion

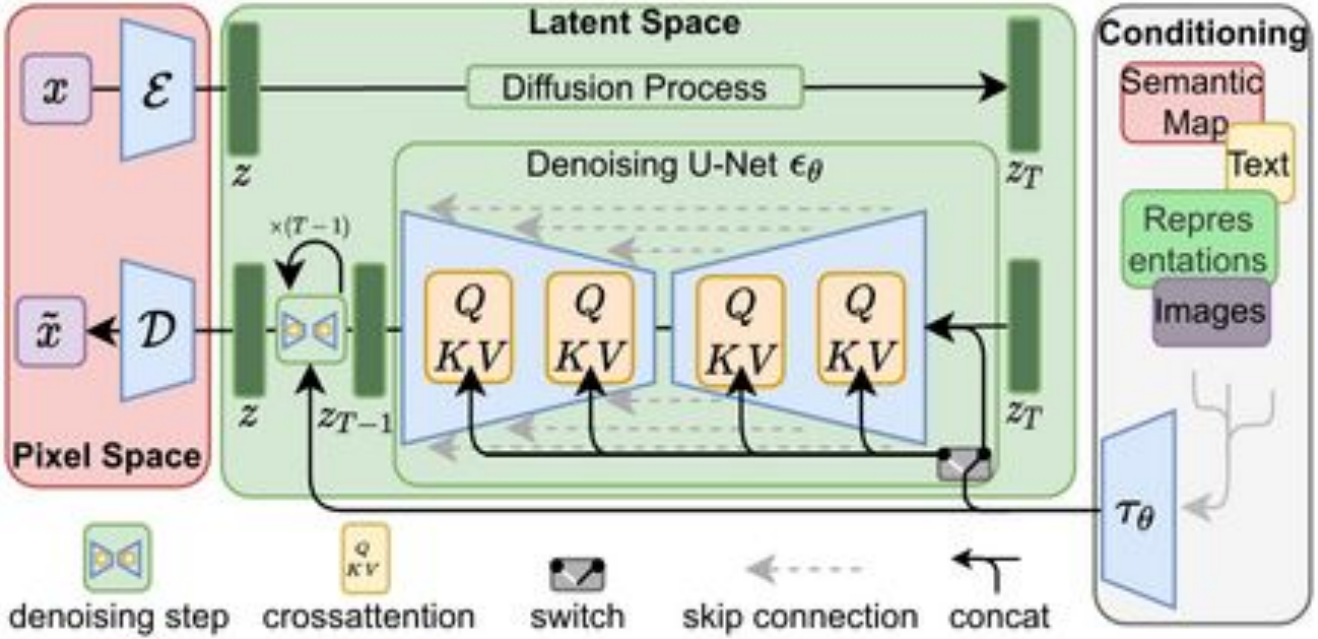

Рисунок 3: Схема работы Stable Diffusion

Преимущества ControlNet

1. Эффективно изучает входные условия даже при небольшом датасете
2. Нацелена на управление диффузионными моделями с учётом конкретных условий задачи, а не на изучение сопоставления между изображениями в разных доменах
3. Манипулирует входными условиями блоков нейронной сети и сохраняет исходные веса => **работает быстрее, чем обучение с нуля**
4. ControlNet можно обучить на одной NVIDIA RTX 3090Ti (обучение нейросети происходит так же быстро, как файнтюнинг диффузионной модели)

Как работает ControlNet

Сетевые блоки (network block)

Сетевой блок - набор нейронных слоев, которые обычно объединяются для формирования единого блока нейронной сети

$\mathcal{F}(\cdot;\Theta)$ – блок с параметрами Θ

$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x};\Theta)$ – преобразование feature map x

$\boldsymbol{x} \in \mathbb{R}^{h \times w \times c}$

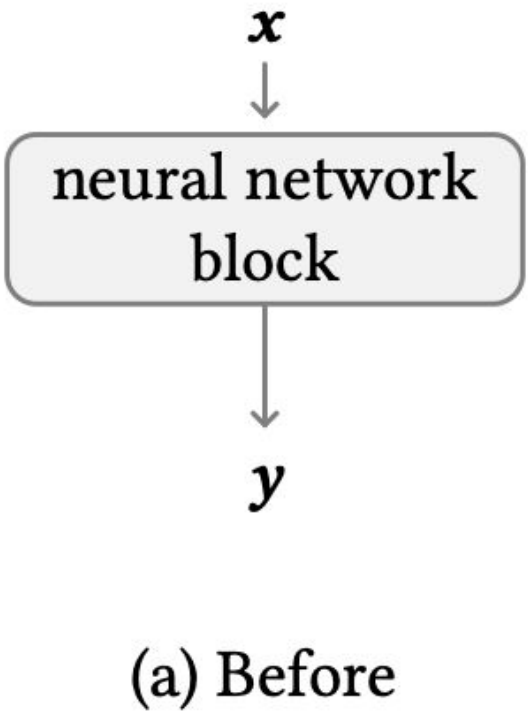

Рисунок 4: сетевой блок

Сетевые блоки (network block)

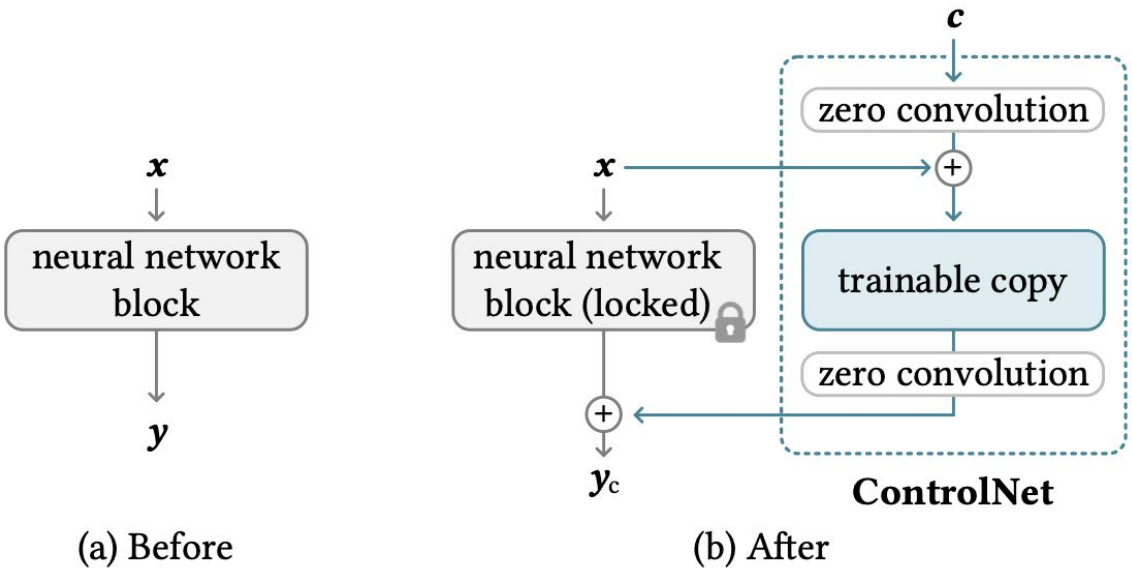

Рисунок 5: блокировка сетевого блока

Блокируем исходный блок, создаем обучаемую копию и соединяем их вместе, используя слои нулевой свертки (свертка 1×1 с нулевым весом и смещением, инициализированными нулем)

Обучаемая копия

- Обучаемая копия соединена с помощью слоев «нулевой свёртки»

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

Θ_{z1}, Θ_{z2} – параметры двух нулевых сверток

$\mathcal{Z}(\cdot;\cdot)$ – нулевая свертка

- Шум не может повлиять на скрытые состояния слоев нейронной сети в обучаемой копии
- Копия сохраняет функционал изначальной модели

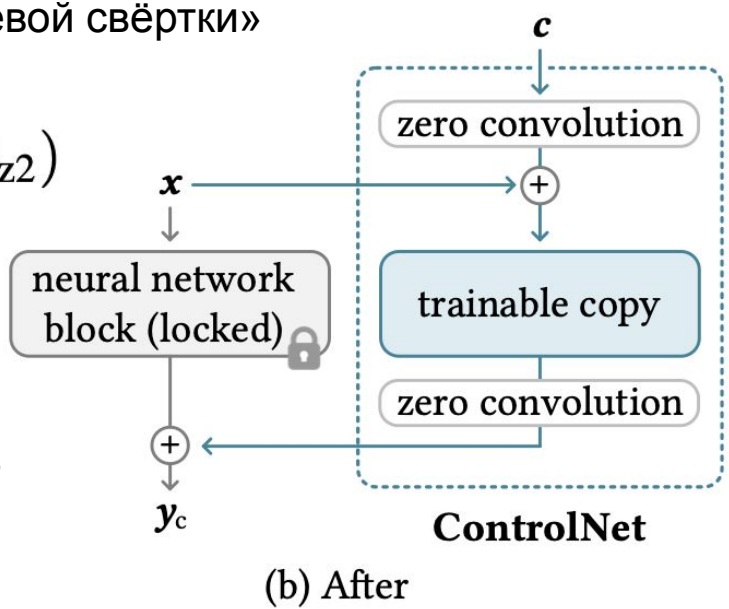

Рисунок 6: обучаемая копия

ControlNet и Stable Diffusion

- Замораживаем параметры начальной модели => не вычисляем градиент 2 раз
- Для добавления ControlNet в Stable Diffusion нужно также преобразовать изображения с условиями в латентное пространство

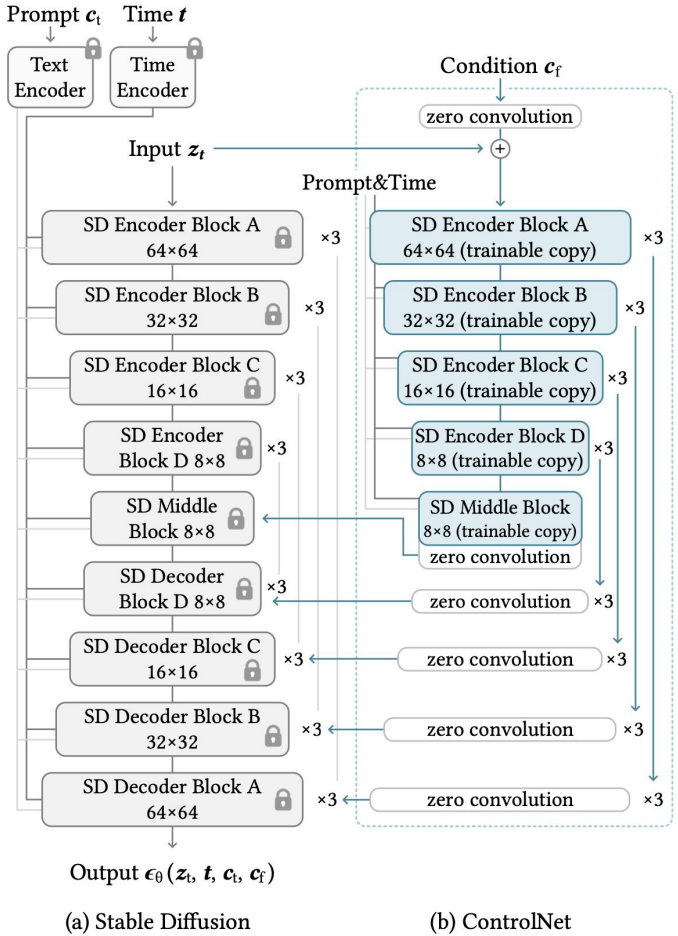

Рисунок 7: применение ControlNet на примере Stable Diffusion

Обучение ControlNet

Обучение LDM

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

К исходному изображению $\boldsymbol{z}_0$ постепенно добавляем шум и получаем $\boldsymbol{z}_t$

$\boldsymbol{z}_t$ – зашумленное изображение

$\boldsymbol{c}_t$ – текстовый промпт

$\boldsymbol{c}_\mathrm{f}$ – условие задачи

ϵ_θ – обучаемая сеть для предсказания добавленного шума

Обучение ControlNet

- Случайно заменяем 50% текстовых промптов на пустые строки
- Модель усваивает условия контроль **внезапно**

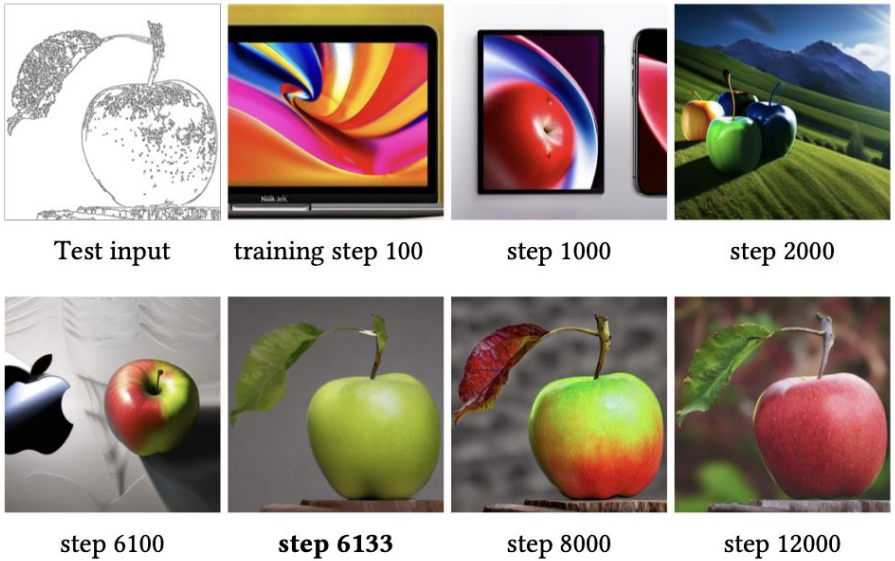

Рисунок 8: Модель внезапно учится следовать входному условию

Inference ControlNet

Classifier-Free Guidance (CFG)

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

ϵ_{prd} – итоговый выход модели ϵ_{uc} – unconditional output

β_{cfg} – вес (гиперпараметр) ϵ_{c} – conditional output

Вес умножается на каждое соединение между Stable Diffusion и ControlNet

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

Рисунок 9: Сравнение CFG и CFG Resolution Weighting

Несколько ControlNet

Не требуется никакого дополнительного взвешивания!

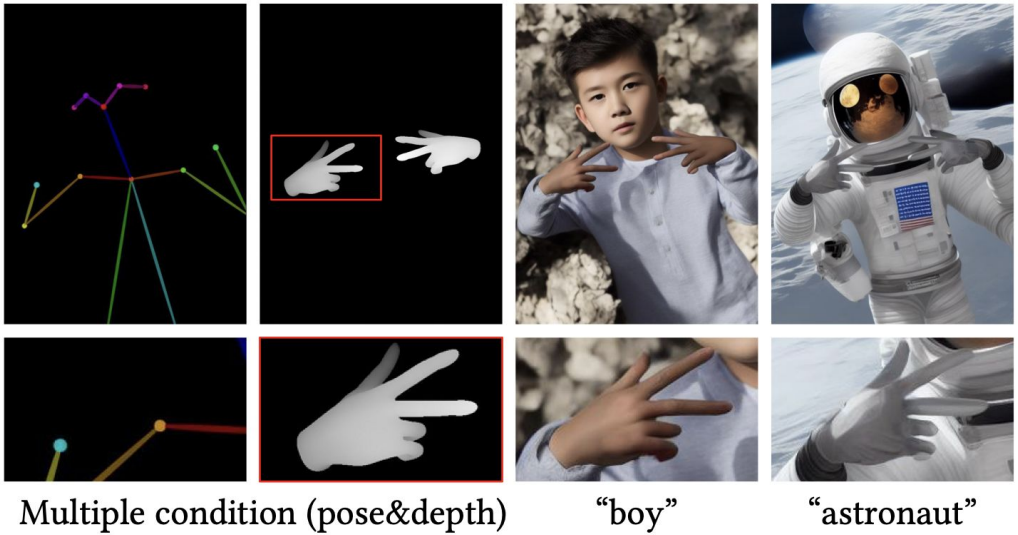

Рисунок 10: Применение нескольких условий

Эксперименты

Генерация без промптов

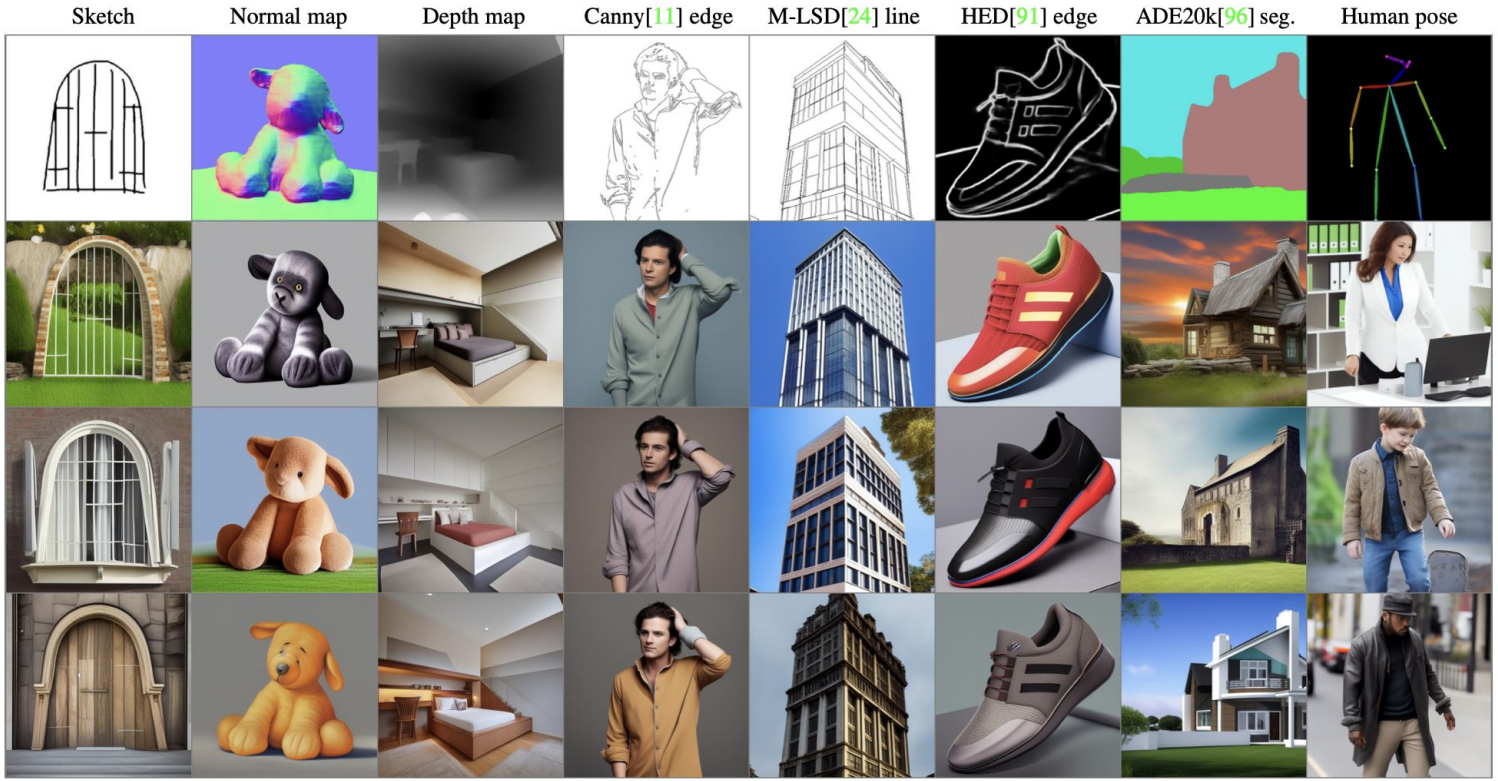

Рисунок 11: Управление Stable Diffusion без промптов

Вариации ControlNet

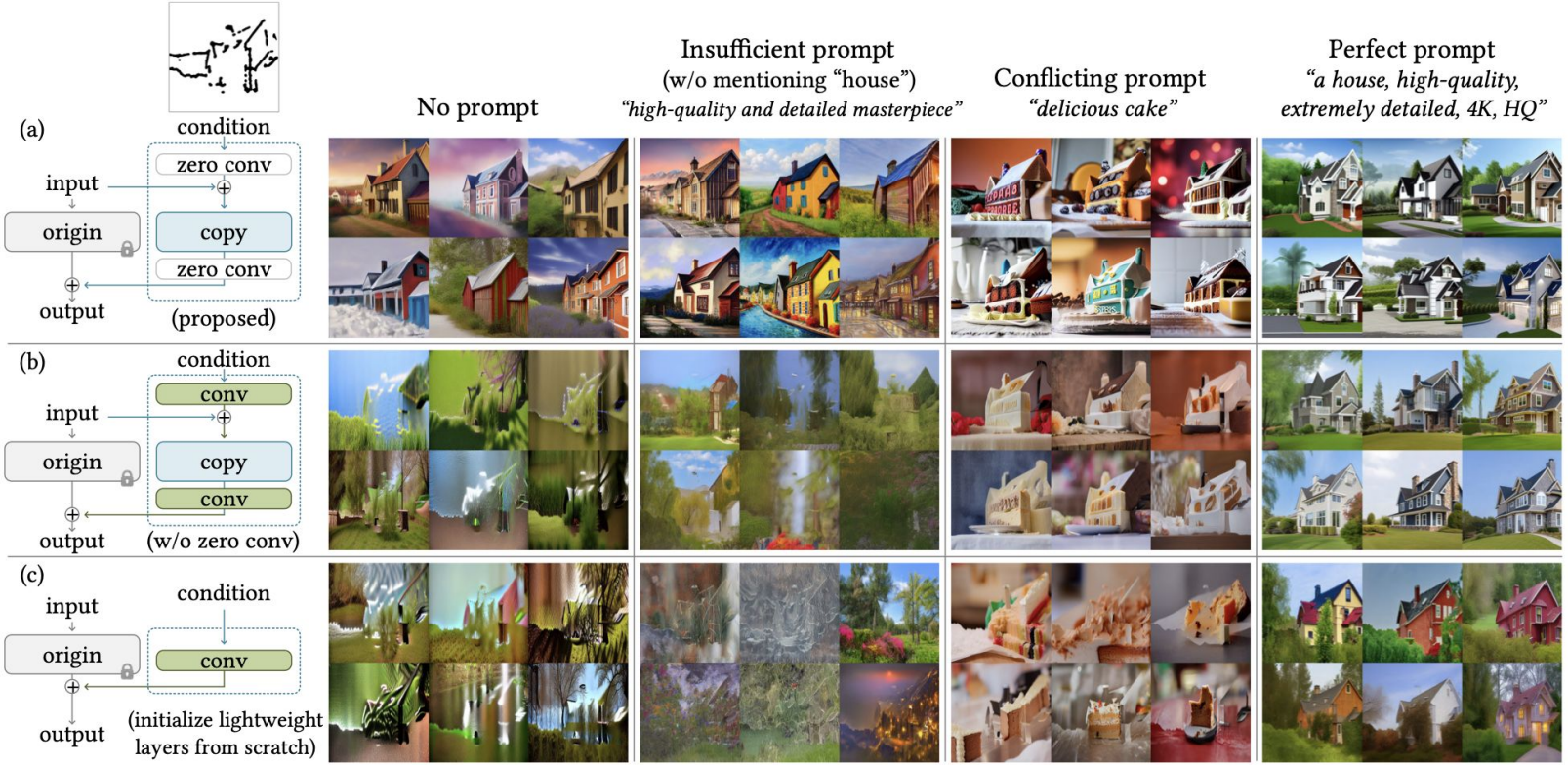

Рисунок 12: Сравнение различных архитектур ControlNet

Сравнение метрик

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

Рисунок 13: Сравнение людьми на основе соотношения текста и изображения и эстетической оценке

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

Рисунок 14: Сравнение на основе автоматизированных метрик

Сравнение с предыдущими подходами

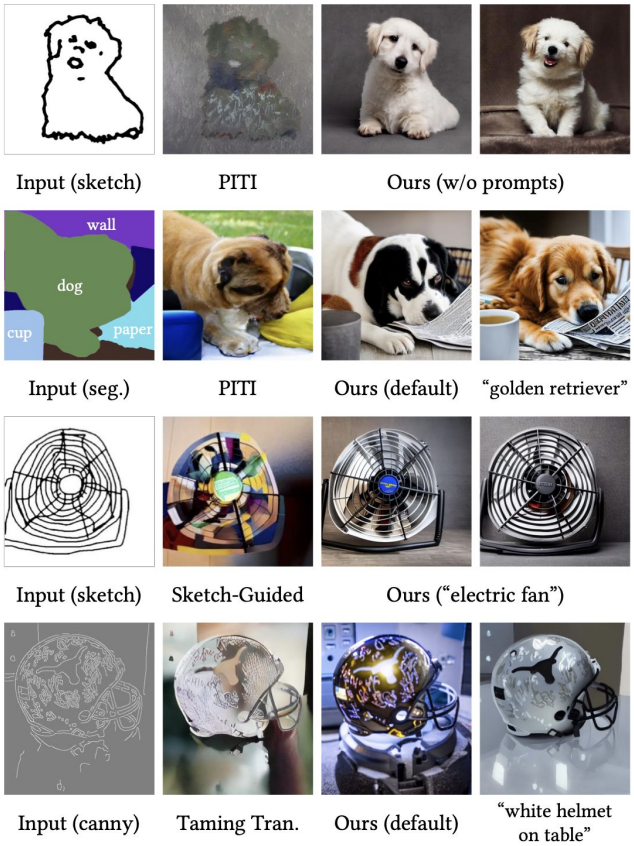

Рисунок 15: Сравнение генераций различных моделей

Рисунок 16: Зависимость качества от размера датасета

Рисунок 17: Результаты при разных предобученных моделях

Рисунок 18: Различная интерпретация исходного изображения

Источники

- https://arxiv.org/pdf/2302.05543
- https://habr.com/ru/companies/ruvds/articles/719348/
- https://learnopencv.com/controlnet/#How-ControlNet-Works?
- https://journal.tinkoff.ru/controlnet/?ysclid=lvtz038zmk209578071

Как попробовать: fast_stable_diffusion_AUTOMATI...