Создание персонализированных генераций изображений

A Preprint

Кристина М. Казистова
ФПМИ
МФТИ
Долгопрудный
kazistova.km@phystech.edu

Степанов Илья Дмитриевич
ФПМИ
МФТИ
Долгопрудный
iliatut94@gmail.com

В генеративных моделях одной из наиболее актуальных задач является сложность создания высококачественных изображений конкретных людей с точностью, передающей их уникальную идентичность.Предлагается сфокусировать внимание на разработке моделей, способных генерировать изображения заданного человека в разнообразных вариациях и с высоким разрешением.Требуется обучить различные модификации метода IP-Adapter на модели Stable Diffusion с использованием множественных подсказок в виде картинок.

Keywords IP-Adapter(1) · Stable Diffusion (2)

1 Введение

В последние годы наблюдается быстрое развитие генеративных моделей, которые решают задачу преобразования текста в изображение. Существующие модели способны генерировать разнообразные изображения по текстовым описаниям с высокой точностью. Однако, в процессе работы с моделями генерации изображений возникают определенные проблемы, одной из которых является недостаточное соответствие сгенерированных изображений и исходным текстовым подсказкам. Наша задача заключается в повышении качества визуальных представлений за счет большего количества графических подсказок. В работе рассматриваются методы, которые позволяют решить вышеупомянутые проблемы, и затем сравниваются между собой. Все описанные далее подходы основаны на применении Stable Diffusion(2). Диффузионная модель представляет собой модель, состоящую из двух процессов: прямого и обратного. Во время прямого процесса к входным данным постепенно добавляется шум, а во время обратного процесса модель постепенно восстанавливает данные из шума. Эта модель позволяет создавать высококачественные изображения на основе текстовых и графических подсказок, открывая новые возможности в области синтеза изображений.

Первый представленный метод — это DreamBooth(3).Он принимает на вход несколько изображений одного объекта вместе с соответствующим названием класса и возвращает специальный токен, идентифицирующий объект, который затем встраивается в текстовую подсказку, по которой генерируется желаемое изображение. Проблемы данного метода заключаются в слабой адаптивности, отсутствии обобщения и необходимости обучать всю диффузионную модель.

Второй метод — это IP-Adapter(1). Он состоит из двух частей: энкодера для извлечения признаков изображения, текста и адаптированных модулей с механизмом перекрестного внимания. Метод принимает на вход только одно изображение объекта. Однако одной картинки может быть мало, для того чтобы модель могла уловить все необходимые зависимости.

В работе предлагается третий метод, представляющий собой модификацию IP-Adapter. На вход подаются несколько изображений вместо одного, причем каждому изображению соответствует своя текстовая подсказка. В процессе обучения модели одно изображение удаляется равновероятно, и модель учится восстанавливать это удаленное изображение, опираясь на текстовое описание и другие имеющиеся изображения. К этим имеющимся изображениям применяется агрегирующая функция. За счет по-

дачи нескольких изображений добиваемся лучшей передачи идентичности. Рассмотренные методы сравниваются между собой по критериям качества генерации и разнообразия, критериям идентичности. Исследование проводится на выборке из датасета LFW Deep Funneled — датасете изображений знаменитостей в высоком разрешении.

2 Постановка задачи

Определим датасет как $\mathfrak{D} = \{(\mathbf{x}_i, \tau_i) : i = 1, \ldots, n\}$, $\mathbf{x}_i$ — латентное представление изображения, τ_i – текстовая подсказка. На этапе обучения на каждом шаге из $\mathfrak{D}$ удаляется изображение $\mathbf{x}_j, j \sim \mathcal{U}\{1, \ldots, n\}$ и решается следующая оптимизационная задача:

$$\epsilon_\theta^* = \arg\min_{\epsilon_\theta} \mathcal{L}(\epsilon, \epsilon_\theta), \tag{1}$$

Определим функцию потерь:

$$\mathcal{L}(\epsilon, \epsilon_\theta) = \mathbb{E}_{\epsilon \sim N(0,I), \mathbf{c}_\tau, G(\mathbf{c}_i \backslash \{\mathbf{c}^j\}), t, \mathbf{x}_t^j} \|\epsilon - \epsilon_\theta(\mathbf{c}_\tau, G(\mathbf{c}_i \backslash \{\mathbf{c}^j\}), t, \mathbf{x}_t^j)\|^2, \tag{2}$$

где G — агрегирующая функция, применяемая ко входным данным; $\mathbf{c}_\tau$ — текстовые признаки удаленного изображения; $\mathbf{c}_i$ — признаки изображений; $\mathbf{c}^j$ — признаки удаленного изображения; $t \in [0, T]$ — временной шаг диффузионного процесса; $\mathbf{x}_t^j = \alpha_t \mathbf{x}^j + \sigma_t \epsilon$ — зашумленные данные удаленного изображения на шаге t; α_t, σ_t — предопределенные функции от t, определяющие диффузионный процесс; ϵ_θ — цель обучения модели диффузии.

Также в данной работе регулируется условие изображения с помощью константы w, чтобы обеспечить управление без использования классификатора на этапе вывода.

$$\hat{\epsilon}_\theta(\mathbf{c}_\tau, G(\mathbf{c}_i \backslash \{\mathbf{c}^j\}), t, \mathbf{x}_t^j) = w\epsilon_\theta(\mathbf{c}_\tau, G(\mathbf{c}_i \backslash \{\mathbf{c}^j\}), t, \mathbf{x}_t^j) + (1 - w)\epsilon_\theta(\mathbf{c}_\tau, t, \mathbf{x}_t^j) \tag{3}$$

Поскольку перекрестное внимание к тексту и перекрестное внимание к изображению разделены, мы также можем настроить вес условия изображения на этапе вывода:

$$\mathbf{Z}^{new} = Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \lambda \cdot Attention(\mathbf{Q}, \mathbf{K}^{'}, \mathbf{V}^{'}), \tag{4}$$

где λ — весовой коэффициент, $\mathbf{Z}$ — признаки запроса, $\mathbf{Q} = \mathbf{Z}\mathbf{W}_q$, $\mathbf{K} = \mathbf{c}_t\mathbf{W}_k, \mathbf{K}^{'} = G(\mathbf{c}_i \backslash \{\mathbf{c}^j\})\mathbf{W}_k^{'}, \mathbf{V} = \mathbf{c}_t\mathbf{W}_v, G(\mathbf{c}_i \backslash \{\mathbf{c}^j\})\mathbf{W}_v^{'}$ — матрицы запросов, ключей и значений механизмов внимания для текста и изображений соответственно, а $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_{k_i}^{'}, \mathbf{W}_v, \mathbf{W}_{v_i}^{'}, \mathbf{W}_{kj}^{'}, \mathbf{W}_{vj}^{'}$ — соответствующие весовые матрицы.

Метрики качества:

Frechet Inception Distance (FID), Inception Score (IS) — это метрики качества, которые используются для оценки качества сгенерированных изображений

$$FID = ||\mu_p - \mu_q||^2 + Tr(\mathbf{\Sigma_p} + \mathbf{\Sigma_q} - 2(\mathbf{\Sigma_p}\mathbf{\Sigma_q})^{1/2}) \tag{5}$$

где μ_p и μ_q — средние значения признаков в реальных и сгенерированных изображениях соответственно, $\mathbf{\Sigma_p}$ и $\mathbf{\Sigma_q}$ — ковариационные матрицы для распределений признаков на реальных и сгенерированных изображениях соответственно.

$$IS(x) = \exp(\mathbb{E}_x \left[D_{KL}(p(y|x)||p(y))\right]) \tag{6}$$

Где D_{KL} - дивергенция Кульбака-Лейблера для двух распределений $p(y|x)$ - вероятность класса y для изображения x и $p(y)$ - равномерное распределение на множестве классов, $\mathbb{E}_x$ - математическое ожидание по всем изображениям x.

3 Планирование эксперимента

В эксперименте рассматривается задача генерации изображений с помощью существующих моделей DreamBooth, IP-Adapter, а также модификации IP-Adapter на датасете LFW Deep Funneled.

3.1 DreamBooth

Как уже отмечалось ранее, данная модель принимает на вход несколько изображений одного объекта вместе с соответствующим названием класса и возвращает специальный токен, идентифицирующий объект. Затем этот токен встраивается в текстовую подсказку, по которой генерируется желаемое изображение. Параллельно применяется функция потерь сохранения класса, основанная на семантическом контексте модели относительно класса, что стимулирует генерацию разнообразных экземпляров, принадлежащих классу субъекта. Вычисление метрик FID и IS производится на всем датасете.

Определим функцию потерь для модели DreamBooth:

$$\mathcal{L}(\epsilon, \epsilon_\theta, \epsilon_\theta^{pr}) = \mathbb{E}_{\epsilon\sim N(0,I),\mathbf{c}_\tau,t,\mathbf{x}_t}\|\epsilon - \epsilon_\theta(\mathbf{c}_\tau, t, \mathbf{x}_t)\|^2 + \lambda \cdot \mathbb{E}_{\epsilon^{pr}\sim N(0,I),\mathbf{c}_\tau^{pr},t,\mathbf{x}_t^{pr}}\|\epsilon^{pr} - \epsilon_\theta(\mathbf{c}_\tau^{pr}, t, \mathbf{x}_t^{pr})\|^2, \quad (7)$$

$\mathbf{c}_\tau$ — текстовые признаки изображений с токеном; $\mathbf{c}_\tau^{pr}$ — текстовые признаки класса изображений; $t \in [0, T]$ — временной шаг диффузионного процесса; $\mathbf{x}_t = \alpha_t\mathbf{x}+\sigma_t\epsilon$ — зашумленные данные изображения на шаге t; $\mathbf{x}_t^{pr} = \alpha_t^{pr}\mathbf{x}^{pr} + \sigma_t^{pr}\epsilon^{pr}$ — зашумленные данные класса изображений на шаге t; α_t, σ_t, $\alpha_t^{pr}, \sigma_t^{pr}$ — предопределенные функции от t, определяющие диффузионный процесс; ϵ_θ, ϵ_θ^{pr} — цели обучения модели диффузии; λ — весовой коэффицент.

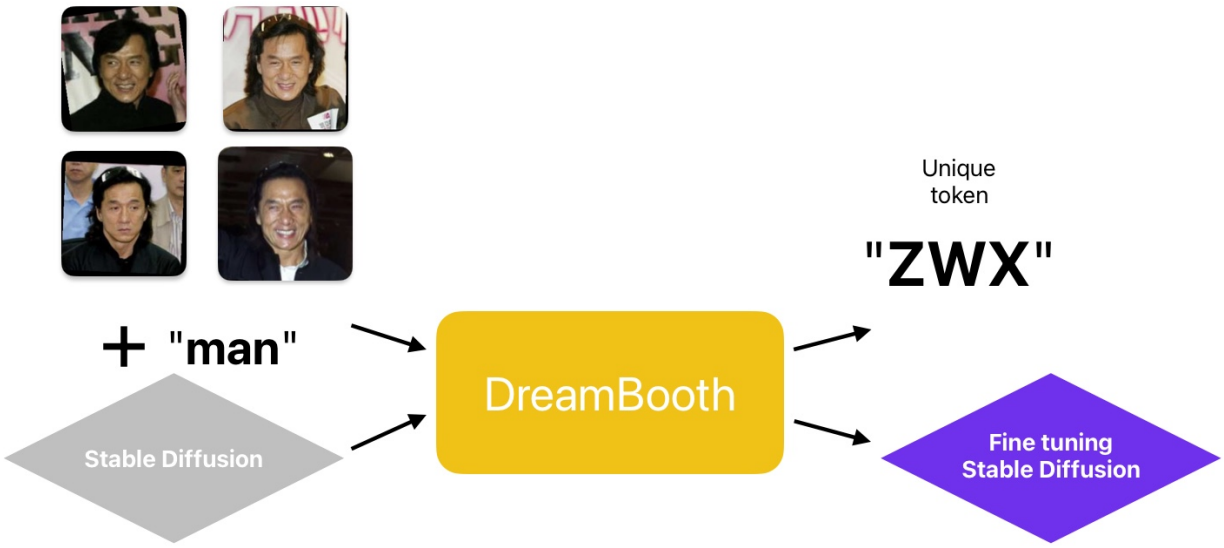

Рис. 1: DreamBooth

3.2 IP-Adapter

IP-Adapter состоит из двух основных компонентов: энкодера, который извлекает признаки изображения и текста, и модулей адаптации с механизмом перекрестного внимания. Он принимает на вход одно изображение. По сравнению с моделью DreamBooth, IP-Adapter обладает большей адаптивностью. Данный подход включает свои модули в предварительно обученную диффузионную модель, что позволяет обучать только энкодер и механизм перекрестного внимания. Вычисление метрик FID и IS производится на всем датасете.

Определим функцию потерь для модели IP-Adapter:

$$\mathcal{L}(\epsilon, \epsilon_\theta) = \mathbb{E}_{\epsilon\sim N(0,I),\mathbf{c}_\tau,\mathbf{c}_i,t,\mathbf{x}_t}\|\epsilon - \epsilon_\theta(\mathbf{c}_\tau, \mathbf{c}_i, t, \mathbf{x}_t)\|^2, \quad (8)$$

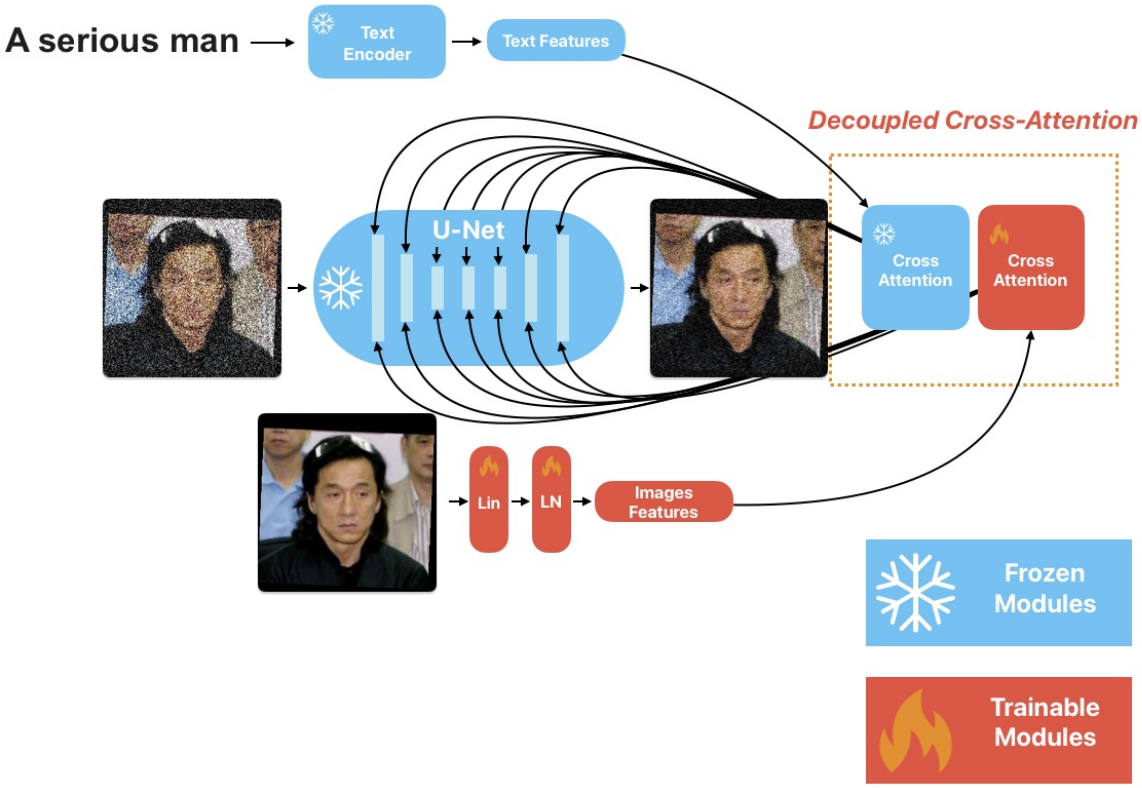

Рис. 2: IP-Adapter

3.3 IP-AdapterMAX и IP-AdapterAVG

Данная модификация метода IP-Adapter включает в себя обработку нескольких изображений, к которым применяются аггрегирующие функции MAXpooling или AVGpooling для их латентных представлений. На вход подаются изображения людей, вычисляются эмбеддинги данных изображений, после чего к эмбеддингам применяются упомянутые ранее функции аггрегации. В данном случае полученное латентное представление интегрируется в полностью предобученную модель IP-Adapter. Вычисление метрик производится на всем датасете.

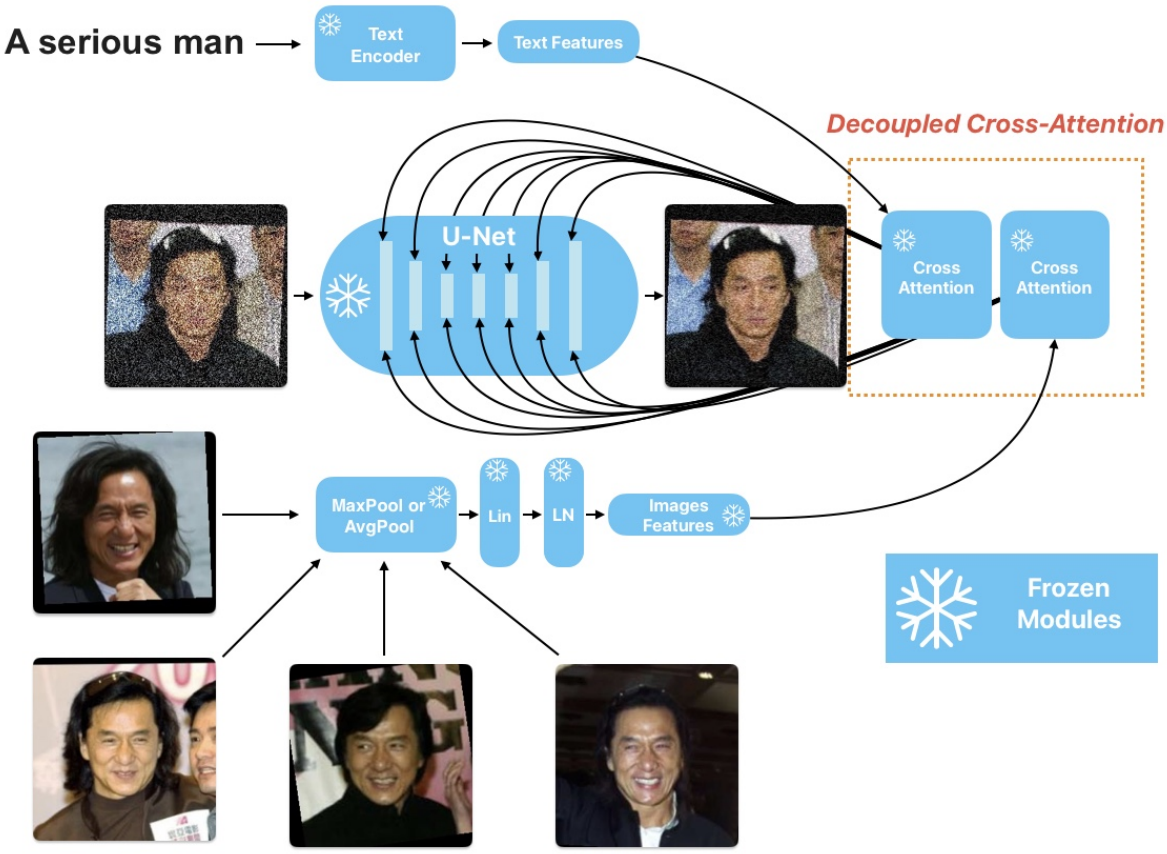

Рис. 3: IP-Adapter with Pooling

3.4 IP-AdapterSelf-Attention

Предложенная модификация метода IP-Adapter включает в себя обработку нескольких изображений, к которым применяется алгоритм Self-Attention(4) для их латентных представлений. Исходный датасет разделяется на тренировочную и тестовую выборки в соотношении 2 : 1. Тренировочная выборка содержит набор персон, каждая из которых обладает 10 изображениями, к каждому из которых прилагается текстовая подсказка. Обучение происходит на 9 изображениях: в ходе обучения осуществляется попытка предсказать 10-е изображение, используя текстовую подсказку и предварительно обработанные эмбеддинги 9 изображений. Выбор удаленного изображения осуществляется равновероятно.

Algorithm 1 Self-Attention

procedure Self-Attention($\mathbf{x}$)
$\mathbf{Q} \leftarrow \mathbf{x} \cdot \mathbf{W}_q$
$\mathbf{K} \leftarrow \mathbf{x} \cdot \mathbf{W}_k$
$\mathbf{V} \leftarrow \mathbf{x} \cdot \mathbf{W}_v$
$\mathbf{Z} \leftarrow softmax\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_k}}\right) \cdot \mathbf{V}$
return $\mathbf{Z} \cdot \mathbf{W}_{out}$
end procedure

После завершения этапа модуля Self-Attention последуют модули IP-Adapter без изменений. В данном случае обучаются модули Self-Attention, Linear и Cross-Attention. Поскольку модификация Self-Attention

обучается на 9 изображениях, то если от пользователя поступит большее или меньшее число изображений, в первом случае лишние изображения просто удаляются, а во втором выполняется процедура бутстрэпа до достижения нужного количества картинок.

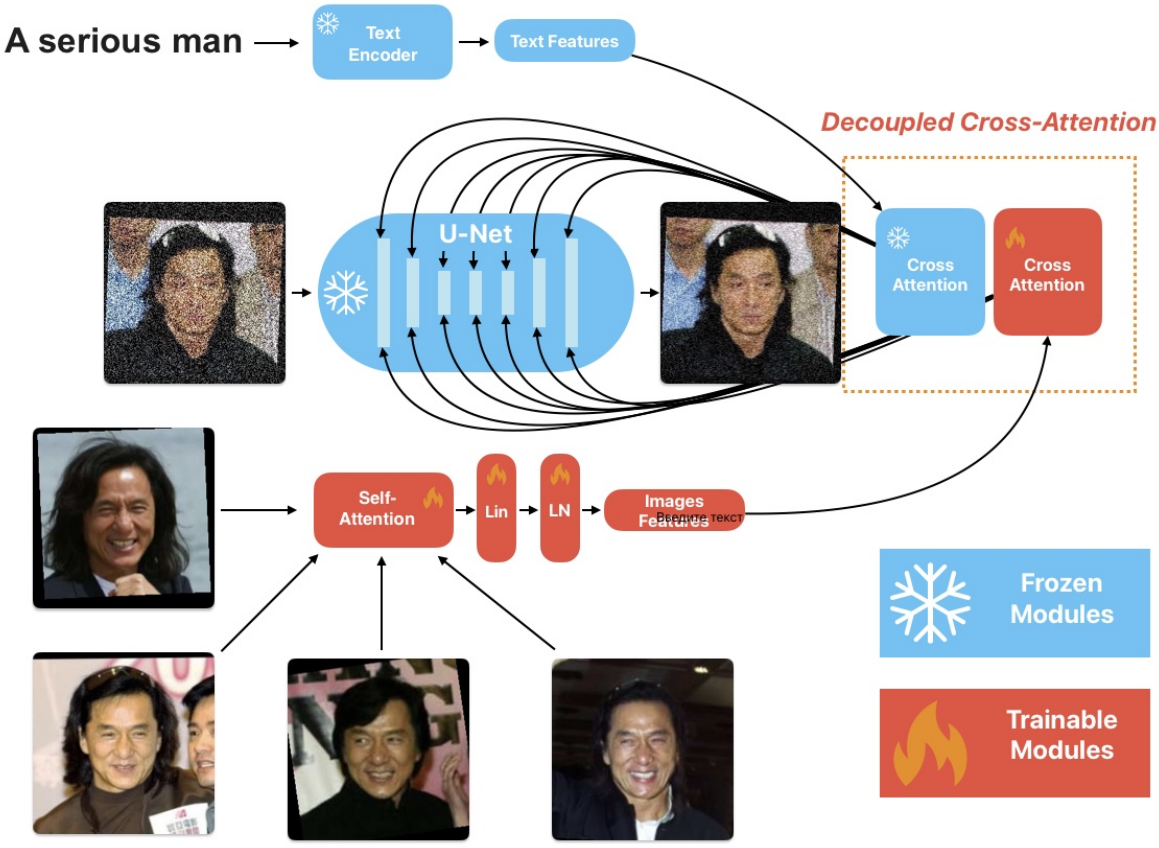

Рис. 4: IP-Adapter with Self-Attention

4 Результаты эксперимента

Метод	IS	FID
IP-Adapter	15.37	8.92
DreamBooth	17.64	9.61
IP-AdapterMAX	14.12	10.10
IP-AdapterAVG	-	-
IP-AdapterSelf-Attention	-	-

5 Заключение

Список литературы

[1] "IP-Adapter"https://arxiv.org/pdf/2308.06721.pdf.

[2] "Latent Stable Diffusion"https://arxiv.org/abs/2112.10752.pdf.

[3] "DreamBooth"https://arxiv.org/pdf/2208.12242.pdf.

[4] "Attention"https://arxiv.org/pdf/1706.03762.pdf.