
Создание персонализированных генераций изображений

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

Степанов Илья Дмитриевич
ФПМИ
МФТИ
Долгопрудный
iliatut94@gmail.com

Филатов Андрей Викторович
Сколковский Институт Технологий
Москва
filatovandreiv@gmail.com

Abstract

Модели генерации изображений по тексту совершили значительный скачок в области искусственного интеллекта, обеспечив высококачественный и разнообразный синтез изображений из заданного текстового описания. Однако, когда возникает запрос на генерацию специфичного объекта, в нашем случае человека, модель не может сгенерировать его с необходимой точностью и передать его идентичность. Предлагается решение, которое будет способно генерировать изображения заданного человека в различных вариациях в высоком разрешении. Мы представляем подход, в основе которого лежит метод IP-Adapter. Данный подход способен обрабатывать несколько изображений одновременно, что приводит к повышению качества генерации.

Ключевые слова: Диффузионная модель, Stable Diffusion[8], IP-Adapter[14], DreamBooth[10].

1 Введение

В последние годы наблюдается быстрое развитие генеративных моделей, которые решают задачу генерации изображений по тексту. 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 исходным текстовым описаниям. Наша задача заключается в повышении качества визуальных представлений за счет большего количества изображений. В работе рассматриваются методы, которые позволяют решить вышеупомянутые проблемы, и затем сравниваются между собой. Все описанные далее подходы основаны на применении диффузионной модели[8].

Диффузионная модель состоит из двух процессов: прямого и обратного. Во время прямого процесса ко 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[10]. Он принимает на вход несколько изображений одного объекта вместе с соответствующим названием класса и возвращает специальный токен, идентифицирующий объект, который затем встраивается в текстовое описание, 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[14]. Он состоит из двух частей: энкодера для извлечения признаков изображения, текста и адаптированных модулей с механизмом перекрестного внимания[13]. 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 векторным представлениям изображений применяется агрегирующая функция. За счет подачи нескольких изображений добиваемся лучшей передачи идентичности. Рассмотренные методы сравниваются между собой по метрикам качества генерации и разнообразия, метрикам идентичности. Исследование проводится на выборке из датасета LFW Deep Funneled[5] — датасете изображений знаменитостей в высоком разрешении.

2 Постановка задачи

Определим датасет как $\mathfrak{D} = \{(x_i, \tau_i) : i = 1, \dots, n\}$, x_i — изображение, τ_i — соответствующий текстовый промпт. Рассматривается модель ϵ_θ из класса диффузионных моделей. На этапе обучения на каждом шаге из \mathfrak{D} удаляется изображение $x_j, j \sim \mathcal{U}\{1, \dots, n\}$, и модель учится восстанавливать его по оставшимся изображениям.

Определим функцию потерь:

$$\mathcal{L}(\epsilon, \epsilon_\theta) = \mathbb{E}_{\epsilon \sim N(0, I), \mathbf{c}_\tau, \mathbf{c}_i, t, \mathbf{c}_t^j} \|\epsilon - \epsilon_\theta(\mathbf{c}_\tau, \mathbf{c}_i, t, \mathbf{c}_t^j)\|^2, \quad (1)$$

где $\mathbf{c}_\tau = \Gamma_\tau(\tau_j)$ — текстовые признаки удаленного изображения, полученные путем применения текстового энкодера Γ_τ к текстовому промпту τ_j ; $\mathbf{c}_i = G(\Gamma_i(x_1), \dots, \Gamma_i(x_{j-1}), \Gamma_i(x_{j+1}), \dots, \Gamma_i(x_n))$ — признаки оставшихся изображений, являющиеся результатом применения агрегирующей функции G к эмбеддингам изображений, полученным с помощью image-энкодера Γ_i ; $\mathbf{c}^j = \Gamma_i(x_j)$ — признаки удаленного изображения; $t \in [0, T]$ — временной шаг диффузионного процесса, $\mathbf{c}_t^j = \alpha_t \mathbf{c}^j + \sigma_t \epsilon$ — запущенные данные удаленного изображения на шаге t ; α_t, σ_t — предопределенные функции от t , определяющие диффузионный процесс.

Решается следующая оптимизационная задача:

$$\epsilon_\theta^* = \arg \min_{\epsilon_\theta} \mathcal{L}(\epsilon, \epsilon_\theta), \quad (2)$$

Текстовые признаки, извлеченные из текстового энкодера, передаются в предобученную диффузионную модель через слои перекрестного внимания. Для передачи в модель признаков изображения каждому слову перекрестного внимания для текстовых признаков сопоставляется слой перекрестного внимания для признаков изображения. Выход полученного слоя изолированного перекрестного внимания определяется как:

$$\mathbf{Z}^{new} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \lambda \text{Attention}(\mathbf{Q}, \mathbf{K}', \mathbf{V}'), \quad (3)$$

где λ — весовой коэффициент, \mathbf{Z} — признаки запроса, $\mathbf{Q} = \mathbf{Z} \mathbf{W}_q$, $\mathbf{K} = \mathbf{c}_t \mathbf{W}_k$, $\mathbf{K}' = \mathbf{c}_i \mathbf{W}'_k$, $\mathbf{V} = \mathbf{c}_t \mathbf{W}_v$, $\mathbf{c}_i \mathbf{W}'_v$ — матрицы запросов, ключей и значений механизмов внимания для текста и изображений соответственно, а $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}'_k, \mathbf{W}_v, \mathbf{W}'_v, \mathbf{W}'_k, \mathbf{W}'_v$ — соответствующие матрицы весов.

Для определения качества модели введем метрики качества генерации Frechet Inception Distance (FID)[2] и Inception Score (IS)[11]:

$$FID = \|\mu_p - \mu_q\|^2 + Tr(\Sigma_p + \Sigma_q - 2(\Sigma_p \Sigma_q)^{1/2}) \quad (4)$$

где μ_p и μ_q — средние значения признаков в реальных и сгенерированных изображениях соответственно, Σ_p и Σ_q — ковариационные матрицы для распределений признаков на реальных и сгенерированных изображениях соответственно.

$$IS(x) = \exp(\mathbb{E}_x [D_{KL}(p(y|x) || p(y))]) \quad (5)$$

Где D_{KL} — дивергенция Кульбака-Лейблера для двух распределений; $p(y|x)$ — вероятность класса y для изображения x ; $p(y)$ — равномерное распределение на множестве классов.

3 Метод

В данной секции мы сначала введем необходимые понятия, затем опишем принцип работы существующих методов решения поставленной задачи: IP-Adapter и DreamBooth. Наконец, представим описание разработанного нами метода.

3.1 Вводные сведения

3.1.1 Диффузионные модели

Диффузионная модель состоит из двух процессов: прямого и обратного.

Прямой процесс представляет собой последовательность зашумленных версий входного изображения x_0, \dots, x_T , где T — количество шагов, а x_t получается по следующей формуле:

$$x_t = \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} \varepsilon, \quad (6)$$

где $\varepsilon \sim \mathcal{N}(0, I)$,

$$x_t | x_{t-1} \sim \mathcal{N}(\sqrt{1 - \beta_t} x_{t-1}, \beta_t I). \quad (7)$$

При $T \rightarrow \infty$, $x_T \rightarrow \mathcal{N}(0, I)$. На последнем шаге итераций получается гауссовский шум.

Положим $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$. Тогда

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \varepsilon, \quad (8)$$

где $\varepsilon \sim \mathcal{N}(0, I)$,

$$x_t | x_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I). \quad (9)$$

Во время обратного процесса исходное изображение восстанавливается из шума. Знаем $x_T \sim \mathcal{N}(0, I)$. Семплирование происходит итеративно шаг за шагом:

$$\hat{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\hat{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon}_t \right) + \sqrt{\beta_t} \varepsilon, \quad (10)$$

где \hat{x}_t — восстановленное изображение на итерации t , при этом, если $t = T$, то $\hat{x}_t = x_t$; $\hat{\epsilon}_t$ — реконструкция шума, полученная моделью для \hat{x}_t ; $\varepsilon \sim \mathcal{N}(0, I)$ — шум, который позволяет генерировать различные изображения.

3.1.2 Stable Diffusion

Процесс работы модели Stable Diffusion состоит из трех основных этапов. Сначала энкодер CLIP[7] преобразует входное изображение в эмбеддинг в пространстве меньшей размерности. Затем диффузионная модель выполняет преобразование полученного эмбеддинга (в скрытом пространстве). Наконец, VAE[6] декодер переводит преобразованный эмбеддинг в изображение исходного размера. В качестве диффузионной модели используется UNet[9] архитектура с механизмом внимания.

3.1.3 Classifier-free Guidance

Можно регулировать силу влияния условия c без специального классификатора. Метод classifier-free guidance[3] позволяет увеличить степень, с которой модель ориентируется на промпт. Во время семплирования предсказание получается путем линейной комбинации предсказаний обусловленной и необусловленной моделей:

$$\hat{\epsilon}_\theta(x_t, c, t) = w \epsilon_\theta(x_t, c, t) + (1 - w) \epsilon_\theta(x_t, t), \quad (11)$$

где w — весовой коэффициент, $t \in [0, T]$ — временной шаг диффузионного процесса, x_t — зашумленные данные удаленного изображения на шаге t .

3.2 DreamBooth

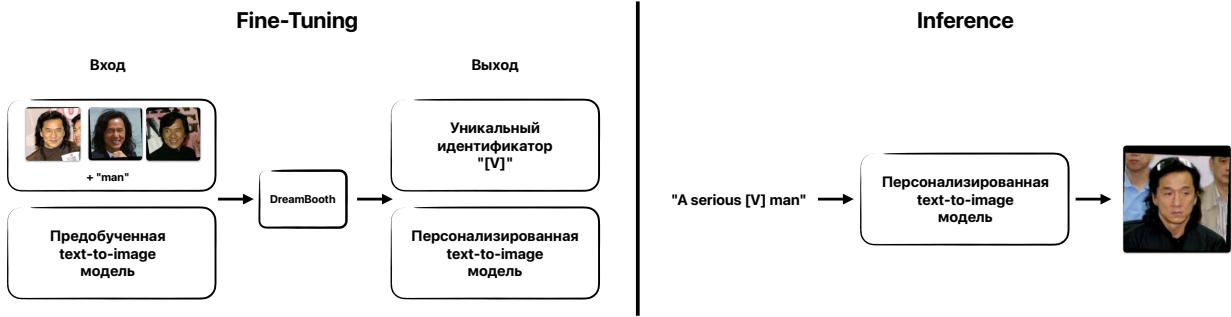


Рис. 1: DreamBooth

В основе данной модели лежит предобученная диффузионная text-to-image модель \hat{x}_θ , функция потерь которой определяется как:

$$\mathbb{E}_{\mathbf{x}, \epsilon \sim N(0, I), \mathbf{c}, t} w_t \|\hat{x}_\theta(\alpha_t \mathbf{x} + \sigma_t \epsilon, \mathbf{c}) - \mathbf{x}\|^2, \quad (12)$$

где \mathbf{x} — латентное представление исходного изображения, $\mathbf{c} = \Gamma(P)$ — вектор условия, полученный с помощью текстового энкодера Γ и текстового промпта P , $t \in [0, T]$ обозначает временной шаг диффузионного процесса; α_t , σ_t , w_t — предопределенные функции от t , определяющие процесс диффузии.

Исходная диффузионная модель дообучается на нескольких входных изображениях одного объекта в паре с текстовым промптом, содержащим название класса, к которому принадлежит данный объект. Генерируются данные $\mathbf{x}_{\text{pr}} = \hat{x}(z, \mathbf{c}_{\text{pr}})$ с использованием сэмплера на основе предобученной диффузионной модели со случайным начальным шумом $z \sim \mathcal{N}(0, I)$ и вектором условия $\mathbf{c}_{\text{pr}} := \Gamma(f("a [class noun"]))$, где f — токенизатор. Функция потерь принимает следующий вид:

$$\mathbb{E}_{\mathbf{x}, \epsilon, \epsilon', \mathbf{c}, t} [w_t \|\hat{x}_\theta(\alpha_t \mathbf{x} + \sigma_t \epsilon, \mathbf{c}) - \mathbf{x}\|^2 + \lambda w_{t'} \|\hat{x}_\theta(\alpha_{t'} \mathbf{x}_{\text{pr}} + \sigma_{t'} \epsilon', \mathbf{c}_{\text{pr}}) - \mathbf{x}_{\text{pr}}\|^2], \quad (13)$$

где λ — весовой коэффициент.

Генерация изображений происходит путем встраивания уникального идентификатора в текстовый промпт в виде: "a [identifier] [class noun]".

3.3 IP-Adapter

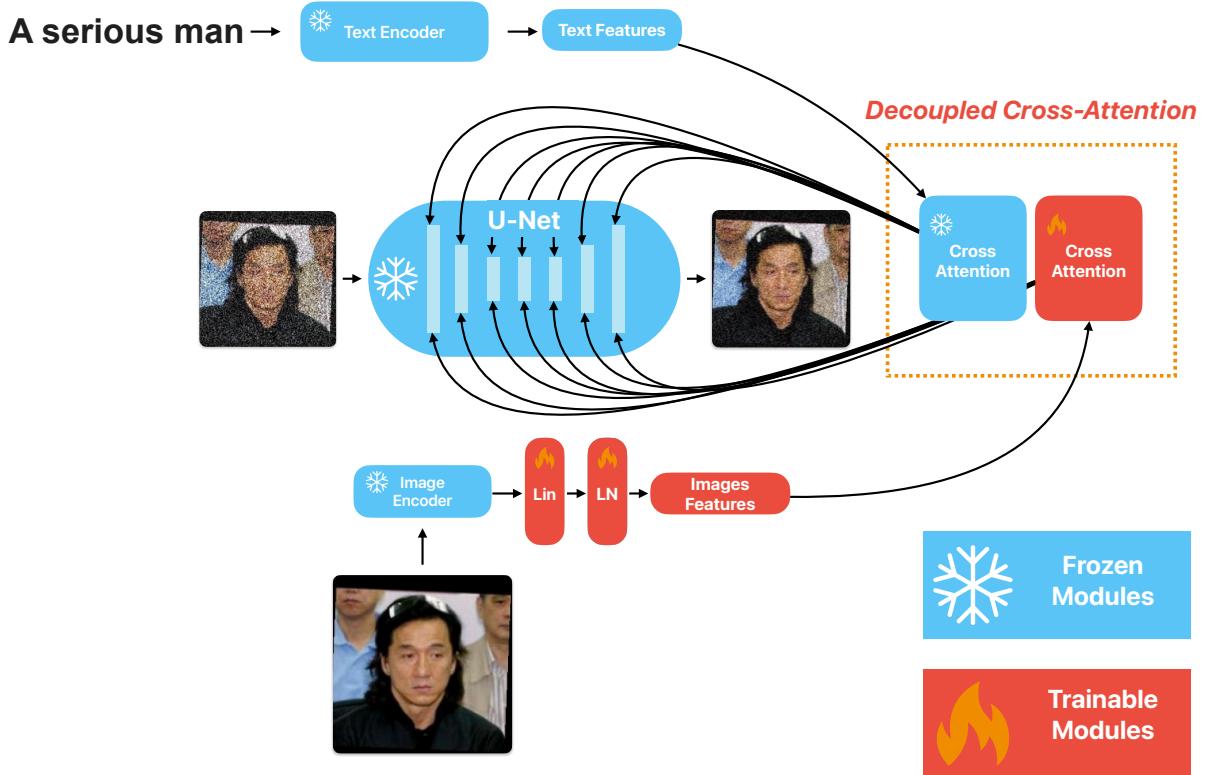


Рис. 2: IP-Adapter

IP-Adapter состоит из двух частей: image-энкодера для извлечения признаков изображения из промпта и адаптированных модулей с механизмом изолированного перекрестного внимания для встраивания признаков изображения в предобученную text-to-image модель.

Для получения признаков изображения используется обучаемая сеть, состоящая из линейного слоя и слоя нормализации, на вход которой подается эмбеддинг изображения, полученный с помощью предобученного image-энкодера CLIP. На этапе обучения вышеупомянутый энкодер заморожен.

Текстовые признаки, извлеченные из текстового энкодера CLIP, передаются в предобученную модель UNet через слои перекрестного внимания. Пусть даны признаки запроса \mathbf{Z} , признаки изображения \mathbf{c}_i и текстовые признаки \mathbf{c}_t , тогда выход слоя перекрестного внимания \mathbf{Z}' определяется как:

$$\mathbf{Z}' = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \quad (14)$$

где $\mathbf{Q} = \mathbf{Z}\mathbf{W}_q$, $\mathbf{K} = \mathbf{c}_t\mathbf{W}_k$, $\mathbf{V} = \mathbf{c}_t\mathbf{W}_v$ — матрицы запросов, ключей и значений механизма внимания для текстовых признаков соответственно, а \mathbf{W}_q , \mathbf{W}_k , \mathbf{W}_v — соответствующие матрицы весов. Для передачи в модель UNet признаков изображения каждому слою перекрестного внимания для текстовых признаков сопоставляется слой перекрестного внимания для признаков изображения. Пусть даны признаки изображения c_i , тогда выход нового слоя перекрестного внимания \mathbf{Z}'' определяется как:

$$\mathbf{Z}'' = \text{Attention}(\mathbf{Q}, \mathbf{K}', \mathbf{V}') = \text{softmax}\left(\frac{\mathbf{Q}(\mathbf{K}')^T}{\sqrt{d}}\right)\mathbf{V}', \quad (15)$$

где $\mathbf{Q} = \mathbf{Z}\mathbf{W}_q$, $\mathbf{K}' = \mathbf{c}_i\mathbf{W}'_k$, $\mathbf{V}' = \mathbf{c}_i\mathbf{W}'_v$ — матрицы запросов, ключей и значений механизма внимания для признаков изображения соответственно, а \mathbf{W}'_k , \mathbf{W}'_v — соответствующие матрицы весов. Выход изолированного перекрестного внимания получается как сумма вышеупомянутых выходов:

$$\mathbf{Z}^{\text{new}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} + \text{softmax}\left(\frac{\mathbf{Q}(\mathbf{K}')^T}{\sqrt{d}}\right)\mathbf{V}' \quad (16)$$

Модель UNet замораживается, поэтому только \mathbf{W}'_k и \mathbf{W}'_v являются обучаемыми параметрами.

В процессе обучения минимизируется следующая функция потерь:

$$\mathcal{L}(\epsilon, \epsilon_\theta) = \mathbb{E}_{\mathbf{x}_t, \epsilon, \mathbf{c}_t, \mathbf{c}_i, t} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_t, \mathbf{c}_i, t)\|^2, \quad (17)$$

где \mathbf{x}_t — зашумленное изображение на шаге t .

Для того чтобы задействовать classifier-free guidance на этапе вывода, во время обучения случайным образом отбрасываются условия изображения:

$$\hat{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}_t, \mathbf{c}_i, t) = w\epsilon_\theta(\mathbf{x}_t, \mathbf{c}_t, \mathbf{c}_i, t) + (1-w)\epsilon_\theta(\mathbf{x}_t, t) \quad (18)$$

Если условие изображения отброшено, эмбеддинг соответствующего изображения зануляется.

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

$$\mathbf{Z}^{\text{new}} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) + \lambda \cdot \text{Attention}(\mathbf{Q}, \mathbf{K}', \mathbf{V}'), \quad (19)$$

где λ — весовой коэффициент.

3.4 IP-Adapter + агрегирующая функция

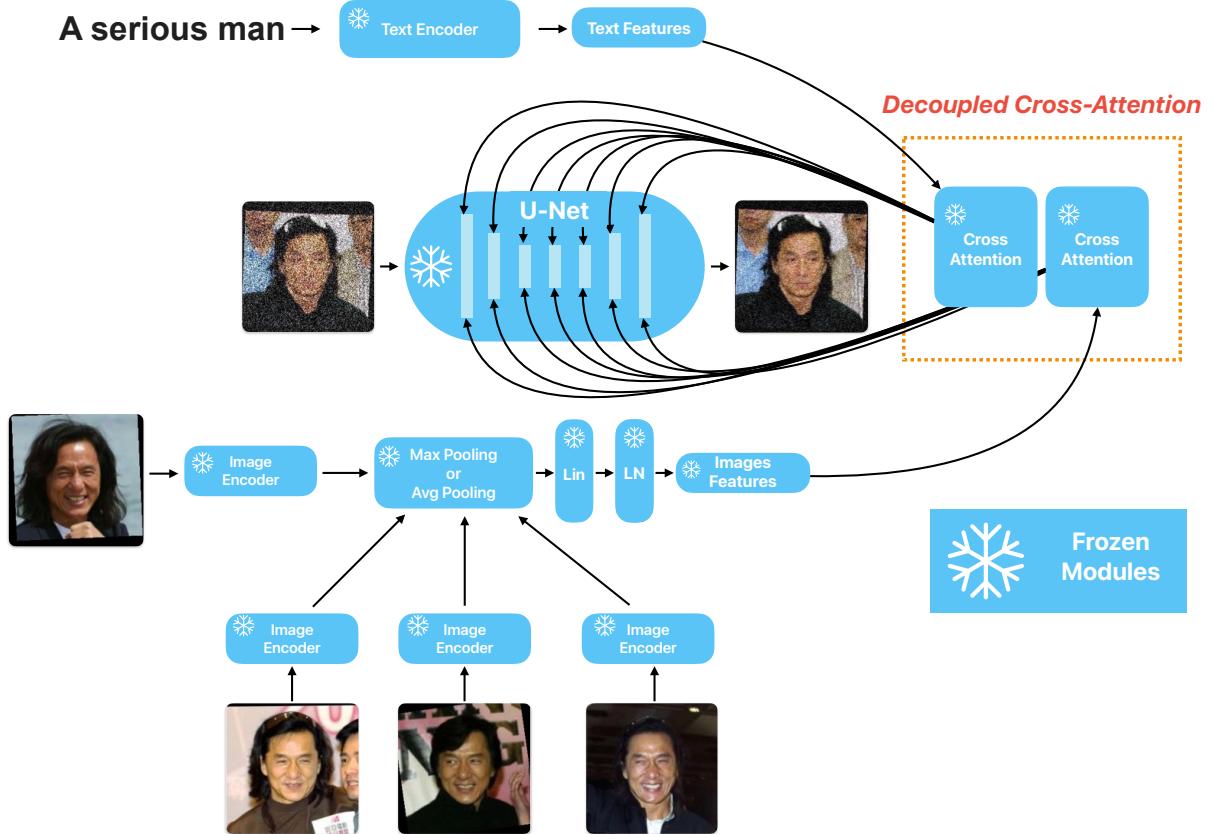


Рис. 3: IP-Adapter + Pooling

Данный метод представляет собой модификацию метода IP-Adapter и принимает на вход несколько изображений вместо одного. К эмбеддингам входных изображений применяется агрегирующая функция (в нашем случае Max Pooling или Average Pooling)[1]. Результат ее применения передается в полностью предобученный IP-Adapter.

3.5 IP-Adapter + Self-Attention

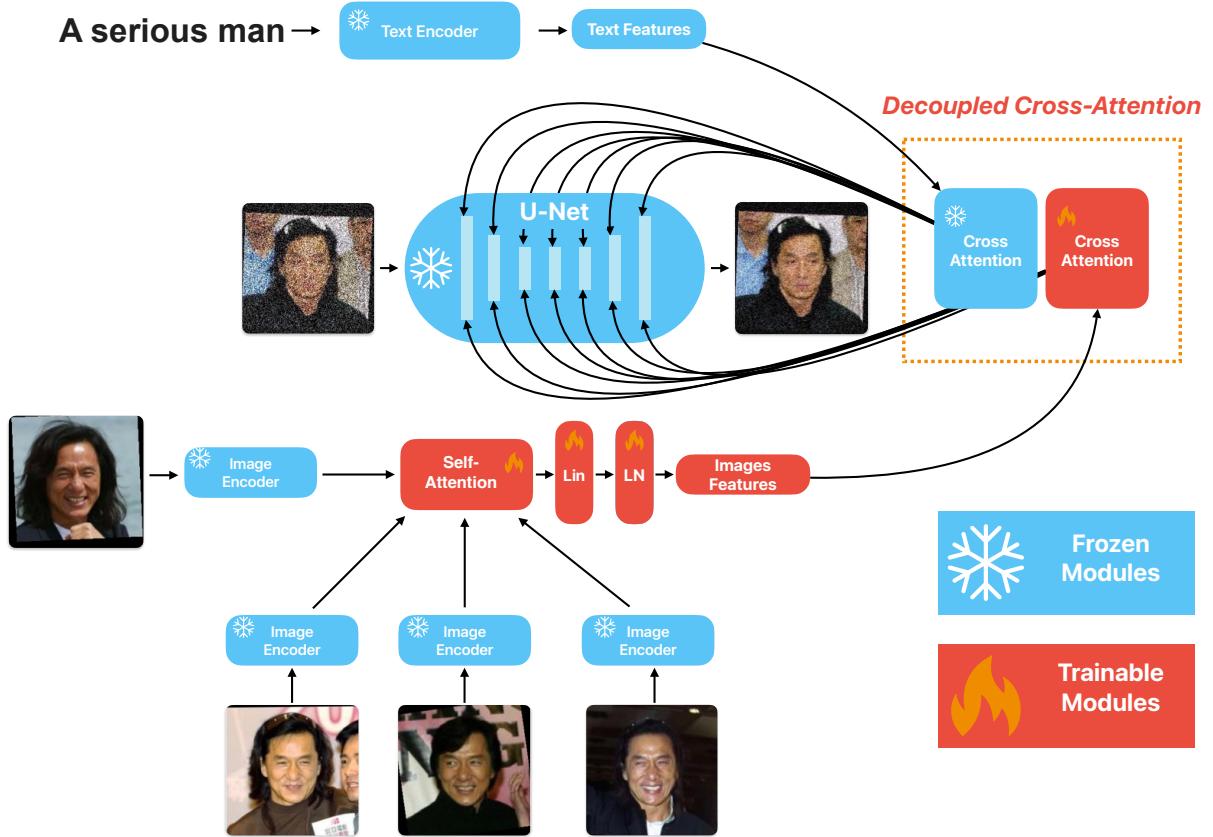


Рис. 4: IP-Adapter + Self-Attention

Предложенная модификация метода IP-Adapter включает в себя обработку нескольких изображений, к эмбеддингам которых применяется алгоритм Self-Attention[13]. На вход модели подается N изображений одного объекта (в нашем случае $N = 10$), каждому из которых соответствует свой текстовый промпт. В ходе обучения случайным образом выбирается изображение, которое удаляется из рассмотрения, и модель пытается предсказать отброшенное изображение по его текстовому промпту и эмбеддингам оставшихся изображений, полученных с помощью image-энкодера CLIP.

После завершения работы модуля Self-Attention следуют модули IP-Adapter без изменений. В данном случае обучаемыми являются слои Self-Attention, Linear, Layer Norm и Cross-Attention. Поскольку модификация Self-Attention обучается на 9 изображениях, то, если на вход поступает большее или меньшее число изображений, в первом случае лишние изображения просто удаляются, а во втором запускается процедура бутстрапа до достижения нужного количества картинок.

Algorithm 1 Self-Attention

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

4 Вычислительный эксперимент

Для экспериментов с вышеупомянутыми моделями используется датасет LFW Deep Funneled, который представляет собой набор изображений лиц людей вместе с их именами. В этом наборе данных 100 людям соответствует не меньше десяти разных фотографий.



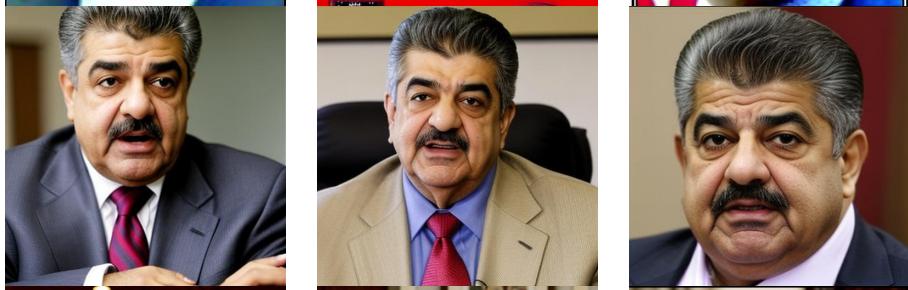
Рис. 5: Примеры изображений из датасета LFW Deep Funneled.

Исходный датасет разделяется на тренировочную и тестовую выборки в соотношении 2 : 1. Модель IP-Adapter + Self-Attention обучается на тренировочной выборке, содержащей набор персон, каждой из которых соответствует имя персоны и несколько её изображений. Процесс обучения описан выше. В остальных случаях для генерации используются предобученные модели, процедура дообучения не проводится.

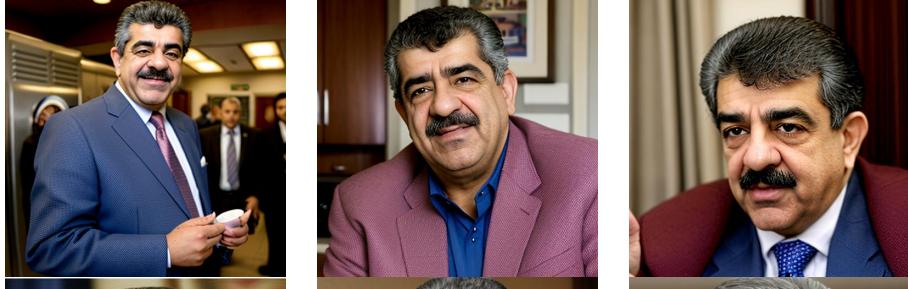
DreamBooth



IP-Adapter



IP-Adapter + Max Pooling



IP-Adapter + Avg Pooling



IP-Adapter +
Self-Attention



Рис. 6: Результаты генерации для рассмотренных моделей.

Точность качества генерации оценивается по метрикам Frechet Inception Distance (FID) и Inception Score (IS). Результаты приведены в таблице:

Метод	IS \uparrow	FID \downarrow
IP-Adapter	15.37	8.92
DreamBooth	17.64	9.61
IP-Adapter + Max Pooling	14.12	10.10
IP-Adapter + Avg Pooling	13.56	11.82
IP-Adapter + Self-Attention	18.72	7.56

Таблица 1: Сравнение предлагаемых нами методов с существующими по метрикам IS и FID на датасете LFW Deep Funneled. Лучшие результаты выделены жирным шрифтом.

5 Заключение

Модели генерации текста в изображение подтолкнули вперед возможности искусственного интеллекта, позволяя создавать качественные и разнообразные изображения на основе текстовых описаний. Тем не менее, возникают трудности при создании изображений конкретных объектов, таких как люди, из-за ограничений точности и передачи идентичности. Для преодоления этих проблем мы предложили новые решения, в том числе улучшенная модификация IP-Адаптера, которая умеет обрабатывать несколько изображений одновременно и улучшает качество генерации. В качестве агрегирующей функции были рассмотрены Pooling и механизм Self-Attention. Данные методы показали высокие значения на метриках качества FID и IS. В дальнейшем существует возможность модифицировать уже наши алгоритмы посредством использования LoRA[4], FaceNet[12] и других.

Список литературы

- [1] Hossein Gholamalinezhad and Hossein Khosravi. Pooling methods in deep neural networks, a review. CoRR, abs/2009.07485, 2020.
- [2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. CoRR, abs/1706.08500, 2017.
- [3] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.
- [4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. CoRR, abs/2106.09685, 2021.
- [5] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
- [6] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. CoRR, abs/1906.02691, 2019.
- [7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. CoRR, abs/2103.00020, 2021.
- [8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. CoRR, abs/2112.10752, 2021.
- [9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. CoRR, abs/1505.04597, 2015.
- [10] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.
- [11] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. CoRR, abs/1606.03498, 2016.
- [12] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. CoRR, abs/1503.03832, 2015.

- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. CoRR, abs/1706.03762, 2017.
- [14] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. 2023.