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

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(2), IP-Adapter(1), DreamBooth(3).

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(2).

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(3).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(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 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 $\mathfrak{D} = \{(x_i, \tau_i) : i = 1, \ldots, n\}$, 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 $x_j, j \sim \mathcal{U}\{1, \ldots, n\}$, и модель учится восстанавливать его по оставшимся изображениям.

Определим функцию потерь:

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

где $\mathbf{c}_\tau = \Gamma_\tau(\tau_j)$ — текстовые признаки удаленного изображения, полученные путем применения текстового энкодера Γ_τ к текстовому промпту τ_j; $\mathbf{c}_i = G(\Gamma_i(x_1), \ldots, \Gamma_i(x_{j-1}), \Gamma_i(x_{j+1}), \ldots, \Gamma_i(x_n))$ — 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; $\mathbf{c}_t^j = \alpha_t \mathbf{c}^j + \sigma_t \epsilon$ — зашумленные данные удаленного изображения на шаге t; α_t, σ_t — предопределенные функции от t, определяющие диффузионный процесс.

Решается следующая оптимизационная задача:

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

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

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

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) и Inception Score (IS):

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

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.

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

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 $x_0, \ldots, x_T$, где T — количество шагов, а x_t получается по следующей формуле:

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

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

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

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

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

$$x_t = \sqrt{\overline{\alpha_t}}x_0 + \sqrt{1-\overline{\alpha_t}}\varepsilon, \tag{8}$$

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

$$x_t|x_0 \sim \mathcal{N}(\sqrt{\overline{\alpha_t}}x_0, (1-\overline{\alpha_t})I). \tag{9}$$

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

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

где $\hat{x}_t$ — восстановленное изображение на итерации t, при этом, если $t = T$, то $\hat{x}_t = x_t$; $\hat{\varepsilon}_t$ — 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 преобразует входное изображение в эмбеддинг в пространстве меньшей размерности. Затем диффузионная модель выполняет преобразование полученного эмбеддинга (в скрытом пространстве). Наконец, VAE декодер переводит преобразованный эмбеддинг в изображение исходного размера. В качестве диффузионной модели используется UNet архитектура с механизмом внимания.

3.1.3 Classifier-free Guidance

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

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

3.2 DreamBooth

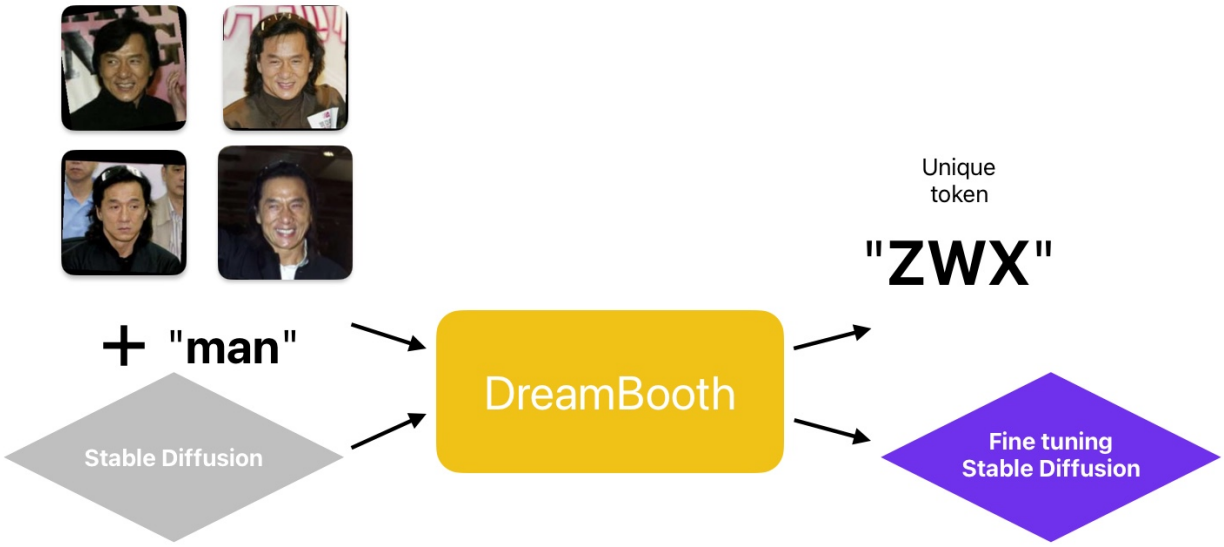

Рис. 1: DreamBooth

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

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

где x — исходное изображение; $c = \Gamma(P)$ — 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 уникальный идентификатор и название класса, к которому принадлежит данный объект (например, "A [V] man"). Генерируются данные $x_{pr} = \hat{x}(z, c_{pr})$ с использованием сэмплера на основе замороженной предобученной диффузионной модели со случайным начальным шумом $z \sim \mathcal{N}(0, I)$ и вектором условия $c_{pr} := \Gamma(f(\text{"a [class noun]"}))$. Функция потерь принимает следующий вид:

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

где λ — весовой коэффициент, а $c = \Gamma(f(\text{"a [identifier] [class noun]"}))$

3.3 IP-Adapter

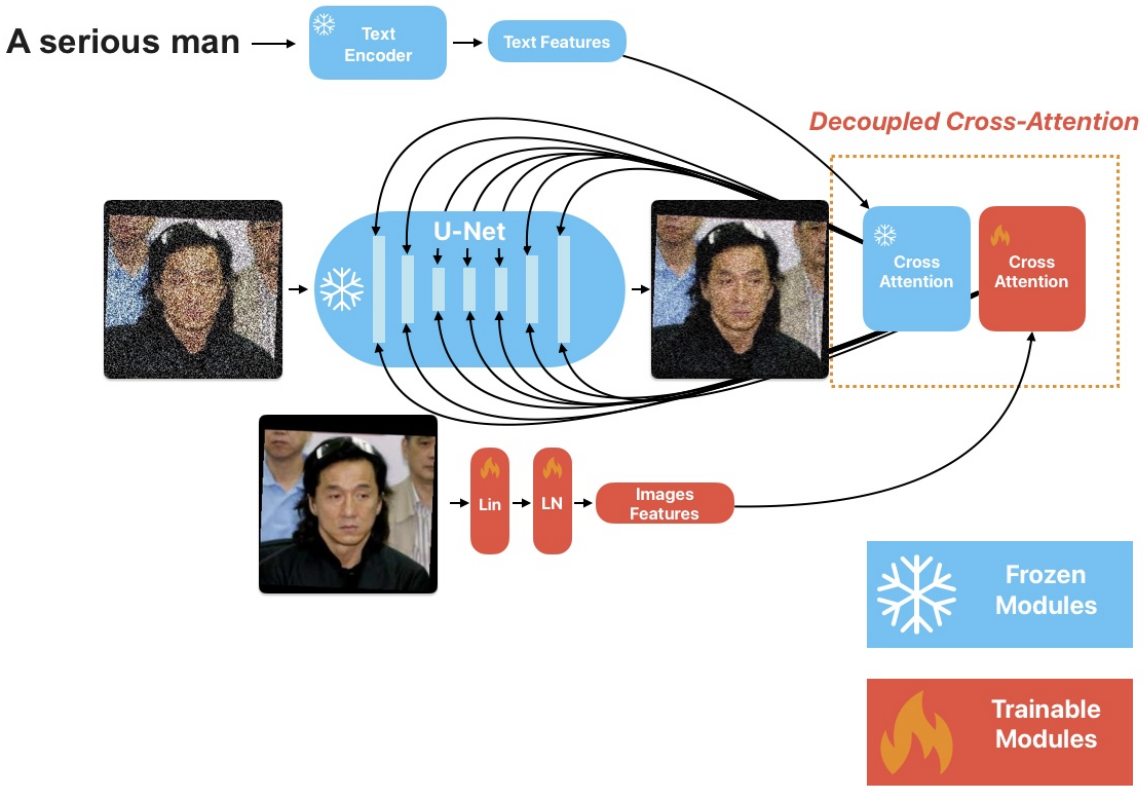

Рис. 2: IP-Adapter

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 Z и текстовые признаки c_t, тогда выход слоя перекрестного внимания Z' определяется как:

$$Z' = Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V,$$

где $Q = ZW_q, K = c_tW_k, V = c_tW_v$ — матрицы запросов, ключей и значений механизма внимания для текстовых признаков соответственно, а W_q, W_k, 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 Z'' определяется как:

$$Z'' = Attention(Q, K', V') = Softmax(\frac{Q(K')^T}{\sqrt{d}})V',$$

где $Q = ZW_q, K' = c_iW'_k, V' = c_iW'_v$ — матрицы запросов, ключей и значений механизма внимания для признаков изображения соответственно, а W'_k, W'_v — соответствующие матрицы весов. Выход

изолированного перекрестного внимания получается как сумма вышеупомянутых выходов:

$$Z^{new} = Softmax(\frac{QK^T}{\sqrt{d}})V + Softmax(\frac{Q(K')^T}{\sqrt{d}})V'$$

Модель UNet замораживается, поэтому только W'_k и W'_v являются обучаемыми параметрами.

В процессе обучения минимизируется следующая функция потерь:

$$\mathcal{L}(\epsilon, \epsilon_\theta) = \mathbb{E}_{x_0,\epsilon,c_t,c_i,t}\|\epsilon - \epsilon_\theta(x_t, c_t, c_i, t)\|^2$$

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

$$\hat{\epsilon}_\theta(x_t, c_t, c_i, t) = w\epsilon_\theta(x_t, c_t, c_i, t) + (1 - w)\epsilon_\theta(x_t, t)$$

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:

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

где λ — весовой коэффициент.

3.4 IP-Adapter + агрегирующая функция

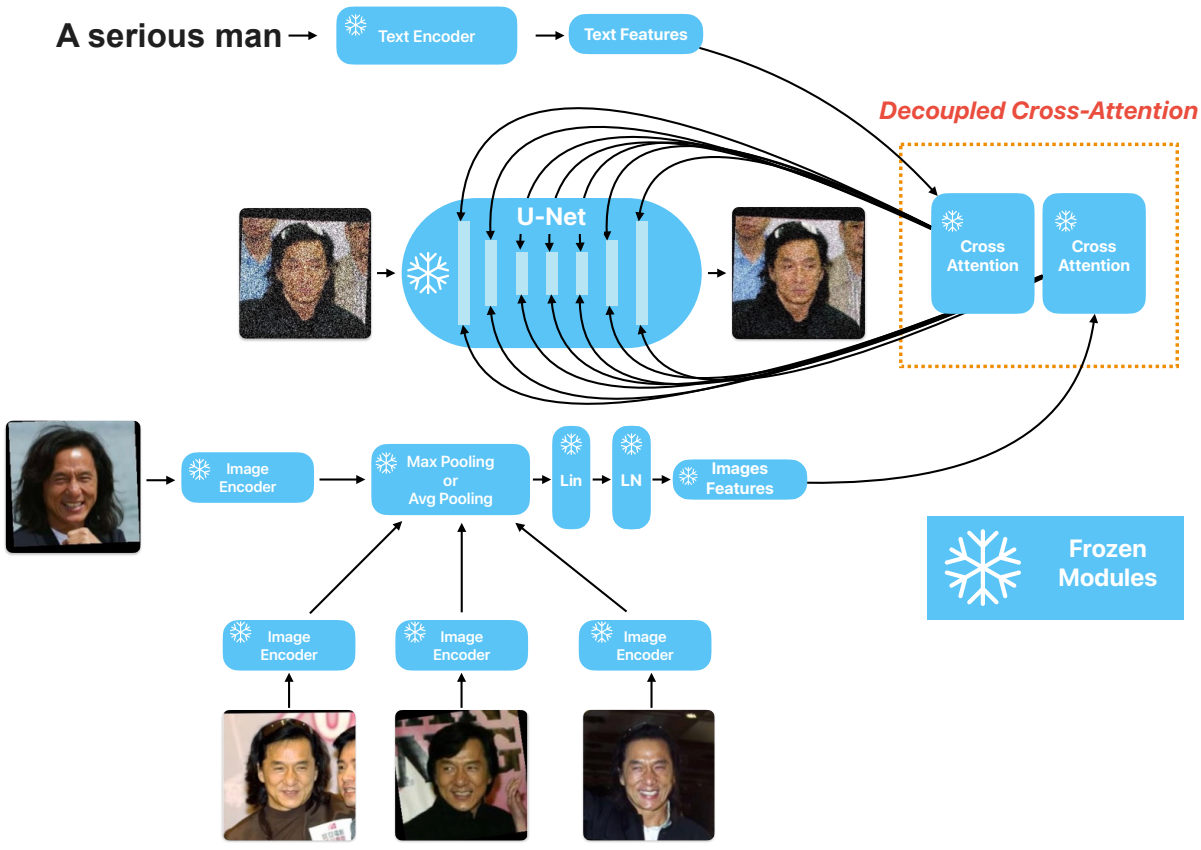

Рис. 3: IP-Adapter + Pooling

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). Результат ее применения передается в полностью предобученный IP-Adapter.

3.5 IP-Adapter + Self-Attention

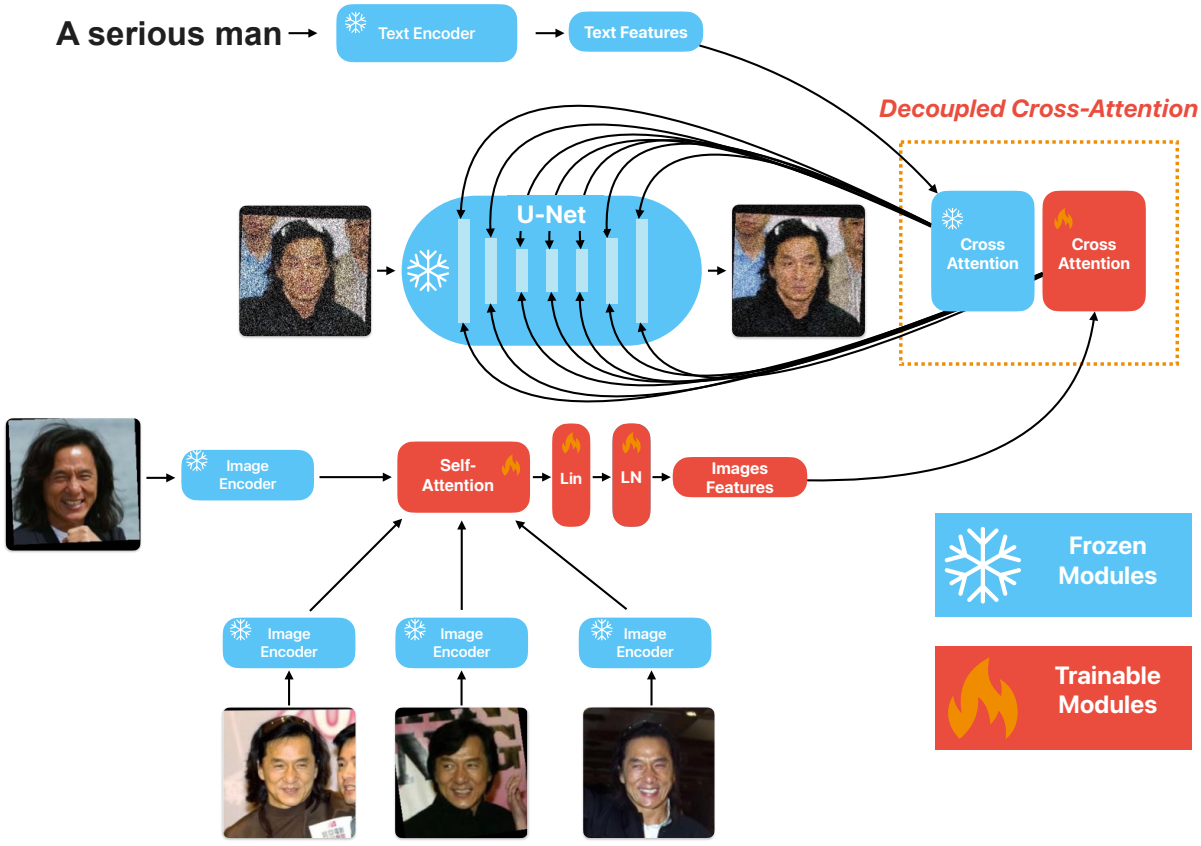

Рис. 4: IP-Adapter + Self-Attention

Предложенная модификация метода IP-Adapter включает в себя обработку нескольких изображений, к эмбеддингам которых применяется алгоритм Self-Attention(4). На вход модели передается 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.

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 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 бутстрепа до достижения нужного количества картинок.

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 1680 людям соответствует не меньше двух 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.

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

Метод	IS ↑	FID ↓
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.

Рис. 6: Результаты генерации для рассмотренных моделей.

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 модифировать уже наши алгоритмы посредством использования LoRA(7), FaceNet(6) и других.

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`.

[5] "Dataset"`https://vis-www.cs.umass.edu/lfw/`.

[6] "FaceNet"`https://arxiv.org/abs/1503.03832`.

[7] "LoRA"`https://arxiv.org/pdf/2106.09685`.