Ускорение семплирования из диффузионных моделей с использованием состязательных сетей

Охотников Н. В.
okhotnikov.nv@phystech.edu

Исаченко Р.В.
isa-ro@yandex.ru

В последние годы широкое распространение получили диффузионные генеративные модели, показывающие высокое качество получаемых семплов и хорошее покрытие исходного распределения. Главный их недостаток – скорость семплирования: для получения одного объекта требуется от сотен до тысяч итераций. Активно исследуются способы ускорения этого процесса. В работе анализируется один из таких способов – использование состязательных моделей для сокращения числа шагов, необходимых для получения семпла. Предлагается развить идею представленной ранее модели Denoising Diffusion GAN. Рассматриваются альтернативные варианты задания сложного распределения в обратном диффузионном процессе, анализируется скорость работы и качество получаемых семплов.

1 Введение

После того, как была представлена Denoising Diffusion Probabilistic Models (DDPM) [5] модели на ее основе становились все популярней, показывая лучшее или сравнимое со state-of-the-art состязательными сетями [6] качество генерации[2], при значительно более простом процессе обучения и более широком покрытии исходного распределения. Несмотря на это, использование диффузионных моделей на практике часто слишком дорого, ввиду необходимости запуска сети до 2000 раз для генерации каждого семпла. Предлагались различные способы ускорения этого процесса [10], многие из которых обобщены в FastDDPM [8], но большинство из них все еще требуют многих десятков итераций и не позволяют приблизиться к состязательным сетям по скорости генерации. В данной работе исследуется проблема дальнейшего ускорения семплирования.

Основное предположение для большинства диффузионных моделей – нормальность условного распределения следующего шага по предыдущему в обратном процессе. Это достигается лишь в приближении бесконечно малых во времени шагов, а значит на практике требует достаточного большого количества итераций. Для того, чтобы обойти это предположения необходимо уметь приближать сложное мультимодальное распределение, ведь каждому шумовому семплу соответствует целое множество семплов из исходного распределения. Одним из способов это реализовать является использование неявных генеративных моделей, например GAN, на каждом шаге обратного диффузионного процесса. В одной из работ [16] уже была представлена модель Denoising Diffusion GAN, реализовавшая эту идею. Подобный подход позволил снизить необходимое количество шагов до нескольких единиц и добиться ускорения семплирования на 2 порядка в сравнении с оригинальной DDPM.

В работе предлагается воспроизвести и развить результат описанный в оригинальной статье про DDGAN [16]. Основной рассматриваемой идеей является использование неявных генеративных моделей для восстановления исходного распределения. В качестве модели рассматривается GAN с различными схемами тренировки. Изучается влияние постановки оптимизационной задачи для состязательной сети на качество получаемых семплов в зависимости от числа шагов обратного диффузионного процесса.

2 Аппроксимация обратного диффузионного процесса мультимодальными распределениями

2.1 Дифузионные модели

В стандартных дифузионных моделях [13, 5] рассматриваются прямой и обратный диффузионные процессы. Модель получает на вход сэмпл из исходного распределения $\mathbf{x_0} \sim q(\mathbf{x})$ и в течение T шагов создает зашумленные его версии $\mathbf{x_1} \dots \mathbf{x_T}$ по следующему правилу:

$$q(\mathbf{x_t}|\mathbf{x_{t-1}}) = \mathcal{N}(\mathbf{x_t}; \sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t\mathbf{I}), \tag{1}$$

где $\{\beta_t \in (0,1)\}_{t=1}^T$. Откуда принимая $\alpha_t = 1 - \beta_t$, $\overline{\alpha_t} = \prod_{i=1}^t \alpha_i$:

$$q(\mathbf{x_t}|\mathbf{x_0}) = \mathcal{N}(\mathbf{x_t}; \sqrt{\overline{\alpha_t}}\mathbf{x_0}, (1-\overline{\alpha_t})\mathbf{I}). \tag{2}$$

При достаточно больших T со сколько угодно большой точностью $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$. Таким образом, прямой диффузионный процесс за большое число шагов, добавляя на каждом некоторый нормальный шум, сводит сэмпл из исходного распределения к сэмплу из стандартного нормального.

Пусть $\mathbf{X} = (\mathbf{x_0^1} \dots \mathbf{x_0^n}) \sim p_0(\mathbf{x})$ – обучающая выборка, т.е. некоторые семплы из распределения, объекты которого модель должна научиться генерировать. $p(\mathbf{x_1} \dots \mathbf{x_T}) = p_\theta(\mathbf{x_T})p_\theta(\mathbf{x_{T-1}}|\mathbf{x_T}) \dots p_\theta(\mathbf{x_0}|\mathbf{x_1})$ – совместное распределение генерируемых на каждом шаге обратного процесса объектов, θ – обучаемые параметры модели. Тогда для построения обратного процесса воспользуемся методом максимального правдоподобия:

$$\theta^* = \arg\max_\theta p(\mathbf{X}|\theta) = \arg\max_\theta \sum_{i=1}^n \log p(\mathbf{x_0^i}|\theta), \tag{3}$$

$$\sum_{i=1}^n \log p(\mathbf{x_0^i}|\theta) \geqslant \sum_{i=1}^n \mathcal{L}(\theta, q; \mathbf{x_0^i}) = \sum_{i=1}^n \int q(\mathbf{x_1} \dots \mathbf{x_T}) \log \frac{p(\mathbf{x_0^i}, \mathbf{x_1} \dots \mathbf{x_T})}{q(\mathbf{x_1} \dots \mathbf{x_t}|\mathbf{x_0^i})} d\mathbf{x_1} \dots \mathbf{x_T}. \tag{4}$$

Неравенство обосновано в [7]. Подставляя в правую часть известные из прямого процесса распределения и пользуясь формулой Байеса получаем, что для максимизации правой части требуется (см. [5]) минимизировать следующее выражение:

$$\sum_{t=1}^n \mathbb{E}_{\mathbf{x_1} \dots \mathbf{x_T}} KL\left(q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0}) \;||\; p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})\right). \tag{5}$$

Для стандартной диффузионной модели после этого шага $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$ принимается нормальным, что очевидно из теоремы Байеса в приближении малости шага, тогда KL-дивергенция существенно упрощается и после правильной параметризации $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_T})$ получаем итоговое выражение, которое оптимизируется в модели:

$$L(\theta) = \sum_{i=1}^n \sum_{t=2}^n \frac{\beta^2}{2\overline{\beta_t}(1-\beta)(1-\overline{\alpha_t})} ||\varepsilon - \widehat{\varepsilon_\theta}(\mathbf{x_t^i}, t)||^2 \longrightarrow \min_\theta, \tag{6}$$

где

$$\overline{\beta_t} = \frac{1-\overline{\alpha_{t-1}}}{1-\overline{\alpha_t}}, \qquad \widehat{\varepsilon_\theta}(\mathbf{x_t^i}, t) = \frac{\mathbf{x_t^i} - \sqrt{\overline{\alpha}_t}\mathbf{x_0^i}}{\sqrt{1-\overline{\alpha_t}}}.$$

2.2 Мультимодальные распределения

Итак, мы показали, что требование к большому количеству итераций в диффузионных моделях вытекает из предположения нормальности $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})$, необходимого для подсчета KL-дивергенции. Однако, это распределение также может быть проинтерпретировано как $p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) = q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0} = f_\theta(\mathbf{x_t}, t))$, т.е. как предсказание $\mathbf{x_0}$ некоторой моделью, зависящей от времени и последующее зашумление его до семпла из $q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0})$ с помощью известных из прямого прохода переходов. Таким образом, если уметь аппроксимировать некоторое сложное мультимодальное распределение (одному зашумленному семплу соответствует целое множество из исходного распределения), то можно достичь поставленной цели – значительно сократить количество итераций обратного прохода.

2.3 DDGAN

Одним из возможных подходов для этого является использование состязательных сетей. В таком случае обратный процесс устроен следующим образом: для каждого времени t из объектов $\{\mathbf{x_t^i}\}$с помощью генератора получаем сэмплы $\{\mathbf{x_0^i}\}$ из исходного распределения, последовательно добавляя шум получаем $\mathbf{x_{t-1}^i}$, и применяем дискриминатор к парам из $\mathbf{x_{t-1}^i}$ и истинных зашумленных объектов из прямого прохода $\{\widehat{\mathbf{x_t^i}}\}$, который обучается определять, являются ли $\mathbf{x_{t-1}^i}$ правдоподобными «расшумленными» версиями $\{\widehat{\mathbf{x_t^i}}\}$.

Подставляя в 5 вместо KL-дивергенции метрику Васерштейна, дивергенцию Йенсена-Шеннона, f-дивергенцию и другие функции, использующиеся в различных подходах к обучению GAN моделей [3, 1, 11, 12] как D_{adv} получаем следующую задачу минимизации:

$$\min_{\theta} \sum_{t \geqslant 1}^{n} \mathbb{E}_{q(\mathbf{x_t})}[D_{adv}\left(q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0}) \;||\; p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})\right)] \tag{7}$$

Зададим дискриминатор от параметров φ как $D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)$ и рассмотрим для начала non-saturating GAN [3], в таком случае дискриминатор будем тренировать на минимизацию следующего выражения:

$$\min_{\varphi} \sum_{t \geqslant 1}^{n} \mathbb{E}_{q(\mathbf{x_t})}[\mathbb{E}_{q(\mathbf{x_{t-1}}|\mathbf{x_t})}[-\log\left(D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)\right)] + \mathbb{E}_{p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}[-\log\left(1 - D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)\right)]]. \tag{8}$$

С известным дискриминатором тренируем генератор на максимизацию:

$$\max_{\theta} \sum_{t \geqslant 1}^{n} \mathbb{E}_{q(\mathbf{x_t})} \mathbb{E}_{p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}[\log\left(D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)\right)]. \tag{9}$$

Причем параметризуем целевое распределение следующим образом:

$$p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t}) := \int p_\theta(\mathbf{x_0}|\mathbf{x_t}) q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0}) d\mathbf{x_0} = \int p(\mathbf{z}) q(\mathbf{x_{t-1}}|\mathbf{x_t}, \mathbf{x_0} = G_\theta(\mathbf{x_t}, \mathbf{z}, t)) d\mathbf{z}. \tag{10}$$

где $p_\theta(\mathbf{x_0}|\mathbf{x_t})$ – неявное распределение порождаемое генератором $G_\theta(\mathbf{x_t}, \mathbf{z}, t) : \mathbb{R}^N \times \mathbb{R}^L \times \mathbb{R} \to \mathbb{R}^N$, прогнозирующим $\mathbf{x_0}$ по $\mathbf{x_t}$ и свободной нормально распределенной переменной $\mathbf{z} \sim p(\mathbf{z}) := \mathcal{N}(\mathbf{z}; 0, \mathbf{I})$. Введение такой случайной переменной и позволяет восстанавливаемому распределению стать сложным мультимодальным.

2.4 Альтернативные схемы тренировки

Оригинальная GAN модель, используемая в DDGAN тренируется на минимизацию JS-дивергенции между исходным и сгенерированным распределениями. JS-дивергенция входит в семейство f-дивергенций определяемых следующим образом:

$$D_f(Q||P) = \int_{\mathcal{X}} p(\mathbf{x}) f\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}, \tag{11}$$

где f – выпуклая непрерывная слева функция, такая что $f(1) = 0$, называемая порождающей. Преобразование Фенхеля функции определяется как

$$f^* = \sup_{u \in \mathrm{dom}_f} (ut - f(u)). \tag{12}$$

Из выпуклости функции и свойств такого преобразования следует:

$$f^{**} = f = \sup_{t \in \mathrm{dom}_{f^*}} (ut - f^*(t)). \tag{13}$$

Тогда произвольную f-дивергенцию можно оценить снизу (подробнее см. [9]):

$$D_f(Q||P) = \int_{\mathcal{X}} p(\mathbf{x}) \sup_{t \in \mathrm{dom}_{f^*}} \left(t\frac{q(\mathbf{x})}{p(\mathbf{x})} - f^*(t)\right) d\mathbf{x} \geqslant \tag{14}$$

$$\geqslant \sup_{T\in\mathcal{T}} \left(\int_{\mathcal{X}} q(\mathbf{x})T(\mathbf{x})d\mathbf{x} - \int_{\mathcal{X}} p(\mathbf{x})f^*(T(\mathbf{x}))d\mathbf{x} \right) = \tag{15}$$

$$= \sup_{T\in\mathcal{T}} \left(\mathbb{E}_{\mathbf{x}\sim Q}[T(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P}[f^*(T(\mathbf{x}))] \right), \tag{16}$$

где $\mathcal{T}$ – произвольный класс функций $T : \mathcal{X} \to \mathbb{R}$. Было показано [9], что при заданных условиях на f оценка достигается при

$$T^*(\mathbf{x}) = f'\left(\frac{q(\mathbf{x})}{p(\mathbf{x})}\right) \tag{17}$$

На основании этой оценки можно построить обучения генеративной сети параметризовав две модели – генеративную $P = P_\theta$ и приближающую функцию из оценки 16 $T = T_\omega$, где ω, θ – векторы обучаемых параметров. В таком случае целевая функция приобретает вид

$$F(\theta, \omega) = \mathbb{E}_{\mathbf{x}\sim Q}[T_\omega(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim P_\theta}[f^*(T_\omega(\mathbf{x}))] \tag{18}$$

Теперь получим частные случаи для различных f-дивергенций, чтобы далее экспериментально оценить влияние постановки минимизационной задачи для GAN на качество и скорость генерации в диффузионной модели. Для этого рассмотрим $T_\omega = g_f(V_\omega)$, где $V_\omega : \mathcal{X} \to \mathbb{R}$, а $g_f : \mathbb{R} \to \mathrm{dom}_{f^*}$ – функция активации, подходящая для выбранной дивергенции.

Рассмотрим квадрат расстояния Хелингера

$$\mathrm{H}^2(Q|P) - \frac{1}{2}\int_{\mathcal{X}} \left(\sqrt{q(\mathbf{x})} - \sqrt{p(\mathbf{x})}\right)^2 d(x). \tag{19}$$

Порождающая функция и ее сопряженная:

$$f_{H^2}(u) = (\sqrt{u} - 1)^2, \qquad f^*_{H^2}(t) = \frac{t}{1-t}. \tag{20}$$

T параметризуем через выход дискриминатора $D_{H^2}(x) = \dfrac{1}{1-\exp(-V)}$, взяв функцию активации:

$$g_{f_{H^2}} = 1 - \exp(V) = \frac{1}{1 - D(x)}. \tag{21}$$

Итого получаем следующую оптимизационную задачу в терминах, введенных для диффузионной модели:

$$\min_\theta \max_\varphi \sum_{t\geqslant 1}^{n} \mathbb{E}_{q(\mathbf{x_t})}[\mathbb{E}_{q(\mathbf{x_{t-1}}|\mathbf{x_t})}[\frac{1}{1-(D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t))}] + \mathbb{E}_{p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}[\frac{1}{D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)}]]. \tag{22}$$

TO DO: ФУКНЦИИ ДЛЯ ДИСКРИМИНАТОРА И ГЕНЕРАТОРА

В качестве другого частного случая, рассмотрим обратную KL-дивергенцию:

$$\text{Reverse-KL}(Q||P) = \mathrm{KL}(P||Q) = \int_{\mathcal{X}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \tag{23}$$

Порождающая функция и ее сопряженная:

$$f_{R-KL}(u) = -\log u, \qquad f^*_{R-KL}(t) = -1 - \log(-t). \tag{24}$$

T параметризуем таким же образом со следующей функцией активации:

$$g_{f_{R-KL}} = -\exp(V) = \frac{D(x)}{1 - D(x)}. \tag{25}$$

Минимизационная задача для диффузионной модели с исопльзованием минимизации обратной KL-дивергенции:

$$\min_\theta \max_\varphi \sum_{t\geqslant 1}^{n} \mathbb{E}_{q(\mathbf{x_t})}[\mathbb{E}_{q(\mathbf{x_{t-1}}|\mathbf{x_t})}[\frac{D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)}{1 - D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)}] + \mathbb{E}_{p_\theta(\mathbf{x_{t-1}}|\mathbf{x_t})}[\log\left(-\frac{D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)}{1 - D_\varphi(\mathbf{x_{t-1}}, \mathbf{x_t}, t)}\right)]]. \tag{26}$$

TO DO: ФУКНЦИИ ДЛЯ ДИСКРИМИНАТОРА И ГЕНЕРАТОРА

Wasserstien GAN ?

3 Вычислительный эксперимент

Целью эксперимента является анализ влияния количества шагов обратного диффузного процесса на качество и скорость семплирования для различных способов ускорения. В качестве метрики качества используем FID-score [4]. Для двух многомерных нормальных распределений Fréchet inception distance определяется как:

$$FID(\mathcal{N}(\mu, \mathbf{\Sigma}), \mathcal{N}(\mu', \mathbf{\Sigma}'))^2 = \|\mu - \mu'\|_2^2 + \mathbf{tr}\left(\mathbf{\Sigma} + \mathbf{\Sigma}' - 2\left(\mathbf{\Sigma}^{\frac{1}{2}} \cdot \mathbf{\Sigma}' \cdot \mathbf{\Sigma}^{\frac{1}{2}}\right)^{\frac{1}{2}}\right). \tag{27}$$

Для других распределений такая формула является ближайшим практически вычислимым приближением. В качестве сравниваемых распределений для вычисления метрики рассматривают распределения признаков информативных векторных представлений семплов, полученных с помощью предобученной сверточной сети InceptionV3 [14].

Тестируем все рассматриваемые модели на FashionMNIST [15].

3.1 Диффузионная модель

Для начала обучаем простую диффузионную модель аналогично [5] и измерим FID-score.

В качестве модели для восстановления шума в обратном проходе используем собственную реализацию U-Net архитектуры. Для различного количества шагов T от 10 до 1000 обратного диффузионного процесса считаем метрику и строим график FID(T) (рис. 1).

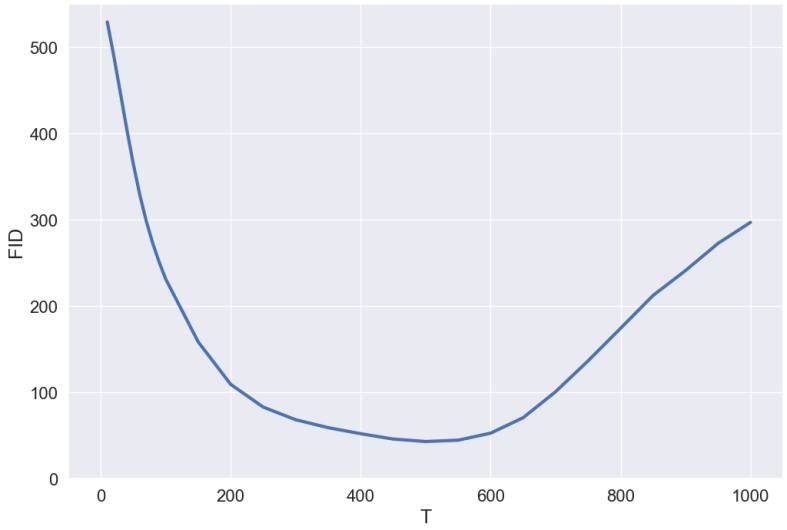

Рис. 1: Зависимость FID от количества шагов для DDPM

Как и ожидалось, качество семплов) значительно повышается с ростом T, что видно на примерах, см. рис. 2.

3.2 DDGAN

Теперь заменим модель, используемую для диффузионного процесса на неявную модель. Для начала используем non-saturating GAN [3] как в оригинальной статье. Используем в качестве дискриминатора половину U-Net модели из базового эксперимента с дополнительным линейным выходным слоем, в качестве генератора аналогичный U-Net с дополнительными слоями эмбеддинга для латентной переменной.

Целью использования неявной модели является существенное уменьшение числа шагов обратного процесса, поэтому будем рассматривать различные T от 1 до 10. Для них посчитаем FID, построим график (рис. 3) и рассмотрим сгенерированные моделью примеры. (рис. 4)

(a) T = 50 (b) T = 300 (c) T = 1000

Рис. 2: Семплы из DDPM для различных значений T

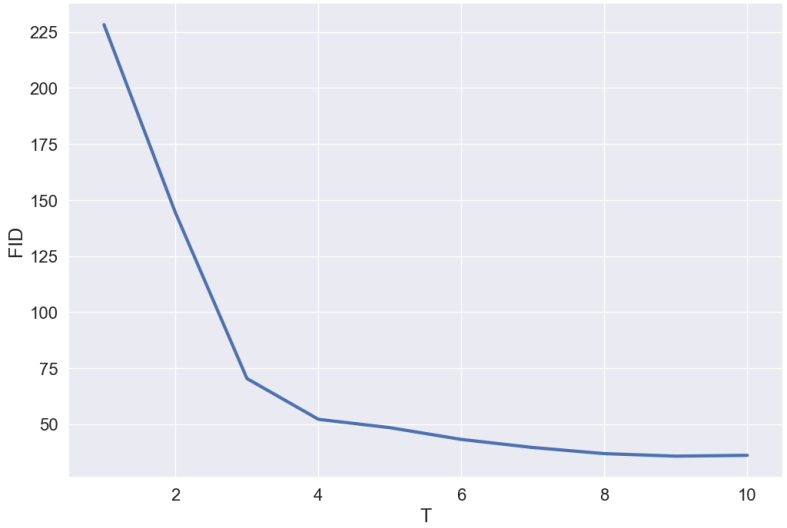

Рис. 3: Зависимость FID от количества шагов для DDGAN

(a) T = 3 (b) T = 5 (c) T = 10

Рис. 4: Семплы из DDGAN для различных значений T

3.3 Альтернативные GAN модели

Далее изменим постановку задачи минимизации, для соответствия другой схеме тренировки GAN модели – (TO DO)

Список литературы

[1] Martin Arjovsky, Soumith Chintala, and L?on Bottou. Wasserstein gan, 2017.

[2] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 2017.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[8] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models, 2021.

[9] XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory, 56(11):5847–5861, nov 2010.

[10] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

[11] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization, 2016.

[12] Matt Shannon, Ben Poole, Soroosh Mariooryad, Tom Bagby, Eric Battenberg, David Kao, Daisy Stanton, and RJ Skerry-Ryan. Non-saturating gan training as divergence minimization, 2020.

[13] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[15] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[16] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans, 2021.