

Лабораторная работа: применение DDPM для генерации изображений из датасета

Федоров Артем Максимович

Январь 2025

Abstract

Настоящий документ представляет отчёт о выполненной работе в рамках вступительного испытания в лабораторию байесовских методов. В работе исследуются классические подходы применения диффузионных нейросетей (DDPM)¹ для генерации изображений на произвольных датасетах. В качестве результатов представлены фреймворк для обучения DDPM, методы логирования и оценки качества обучаемых моделей, а также сами обученные модели и сгенерированные ими изображения с последующим сравнительным анализом результатов.

1 Постановка задачи

Пусть задано множество изображений $\mathcal{I} = \{x_i\}_{i=1}^N$, где $x_i \in \mathbb{R}^{3 \times d \times d}$. Определим множества:

$$\mathcal{I}_{train}, \mathcal{I}_{test} : \mathcal{I}_{train} \cap \mathcal{I}_{test} = \emptyset, \mathcal{I}_{train} \cup \mathcal{I}_{test} = \mathcal{I}$$

Для заданных подмножеств рассмотрим параметрическую генеративную модель $G_{\theta_g} : \mathbb{R}^h \rightarrow \mathcal{I}$ и поставим задачу оптимизации функционала $J(\theta_g, \mathcal{I}_{train}) \rightarrow \min_{\theta_g}$. Валидация результатов обучения будет осуществляться с использованием метрик качества M_i и эмпирической оценки. В дальнейшем распределение реальных данных обозначим как $\pi(x)$.

2 Данные задачи

В качестве данных для проведения экспериментов выбран датасет изображений размерности $3 \times 64 \times 64$, содержащий примеры различных блюд, таких как паста, пироги, мороженое, пончики, рис и другие. Датасет разделён на две подвыборки: обучающую (train) и тестовую (test) для обучения и валидации моделей соответственно, с общей численностью 100,957 изображений (98,937 в обучающей и 2,020 в тестовой выборках).

Изображения изначально классифицированы по типам блюд, при этом распределение классов в обеих подвыборках совпадает ($JSD = 2.4 \cdot 10^{-4}$). Исходя из данного наблюдения, задача генерации изображений решается без учёта меток классов, поскольку алгоритм восстановления плотности должен также восстановить априорное распределение классов. Примеры данных рис. 1.



Figure 1: Примеры изображений датасета

¹DDPM – Diffusion Denoising Probability Model – архитектура генеративных сетей из класса моделей likelihood-based с приближенной плотностью.

3 Оценки качества валидации

Для оценки качества обучения модели будут использованы следующие метрики:

1. **Fréchet Inception Distance (FID)**² – метрика, основанная на расстоянии Фреше между распределениями признаков изображений, извлечённых с использованием предобученной модели InceptionV3.
2. MS-SSIM – расширение метрики SSIM (Structural Similarity Index Measure), основанное на структурном сходстве изображений, вычисляемое по парам изображений исходя из компонентов яркости, контраста и структуры.
3. Kernel-Inception Distance (KID) – метрика, основанная на расстоянии между распределениями признаков изображений, полученных с помощью предобученной модели InceptionV3, вычисляемая с использованием ядерной функции.

Подробнее про используемые метрики качества см. в разделе B.

4 Эксперименты

Конфигурация системы:

- ОС: Ubuntu 20.04.3 LTS
- CPU: Intel Xeon Gold 6336Y 8 vCPU
- GPU: NVIDIA GeForce RTX 2080 Ti (11 GB)
- RAM: 16 GB
- CUDA: 12.4

Общие параметры обучения:

- Число обучающих шагов: < 300,000 – до сходимости
- Шаг логирования потерь: 200
- Шаг логирования весов и метрик: 15,000

Скорость обучения Диффузионных моделей на порядок меньше скорости работы GAN. Однако, по наблюдениям, такие модели так же на порядок быстрее обучаются. Поэтому принято решение обучать модели до сходимости и вручную обрывать процесс, но не более 300000 шагов.

B исследовании используется единая архитектура модели. Подробнее см. в разделе A

4.1 Исследование функционала потерь

Одним из способов улучшения качества генерации DDPM является применение разных функционалов потерь. Оригинальная статья приводит квадратичную ошибку (покоординатное MSE) что следует из теоретического вывода, однако существует ряд примеров реализации DDPM с другими ошибками³, в частности MAE, что добиваются лучших результатов (даже будучи математически необоснованными). В данном разделе сравниваются два способа обучения DDPM при разных функциях потерь: MSE и MAE (`diffusion-linear-scheduler` и `diffusion-mae` соответственно) с линейной функцией шума.

4.1.1 Процесс обучения

²FID – основная метрика качества генерации в данном исследовании.

³Ряд статей по максимированным автоэнкодерам.

На графике 2 представлены зависимости функции потерь от шага обучения. Следует отметить, что, несмотря на то, что значения потерь двух моделей относятся к различным пространствам, их абсолютная динамика все же позволяет делать выводы о сходимости процессов обучения моделей. Видно, что поведение обучения обоих моделей по динамике лосса идентично: резкое падение функции эмпирического риска и выход на плато примерно в один и тот же промежуток обучения. Отсутствуют явные колебания или скачки (что отличает DDPM от GAN).

4.1.2 Сравнение метрик качества

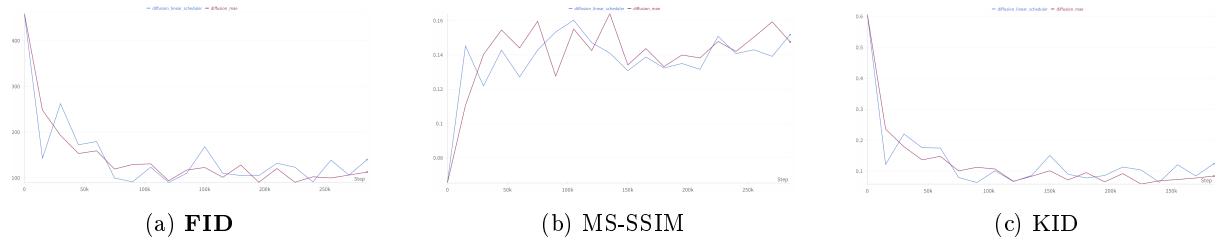


Figure 3: Сравнение метрик качества: синий – **diffusion-linear-scheduler**, красный – **diffusion-mae**

Графики 3 приводят динамику изменения метрик качества двух моделей во времени обучения. Мы видим большую коррелированность двух метрик моделей, что должно говорить о схожести их работы и их обучаемости, тем не менее для **diffusion-linear-scheduler** мы видим относительно **diffusion-mae** возросшую «дисперсию» оценок в точках.

4.1.3 Примеры генерации

Изображения C.1 и C.2 представляют примеры генерации изображений двумя моделями, **diffusion-linear-scheduler** и **diffusion-mae** соответственно. Основные замечания укладываются в тезисы:

- Обе модели часто генерируют темные изображения.
- Обе модели имеют малые артефакты на изображениях (области с будто наложенным гауссовским шумом), однако модель **diffusion-mae** здесь показала себя лучше.
- Модель **diffusion-linear-scheduler** имеет смещение в цветовой гамме генерируемых изображений.
- Имперически, модель **diffusion-mae** имеет большую вариативность генерации.

В итоге модель с функцией потерь тае генерирует более приятные для глаза изображения, что отобразилось и на метриках качества, имеет большую вариативность и

4.2 Изменение закона генерации шума в процессе зашумления

В статье On the Importance of Noise Scheduling for Diffusion Models утверждается, что именно выбор стратегии noise-scheduler в большей степени способен повлиять на качество генерации моделей. В ней же приводятся и противопоставляются линейному шедулеру две стратегии генерации шума через косинусоидальную и сигмоидальную функции.

Использование модели с функцией потерь MAE является ничем не обоснованным трюком, надеяется на который при работе с новым проектом нельзя. В данном эксперименте будем использовать функцию потерь MSE, проверим, насколько сильно получится улучшить качество генерации только лишь изменив закон генерации шума.

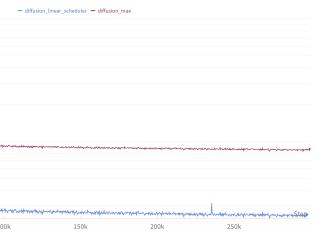


Figure 2: Изменения оценок лосса от шага обучения для: синий – **diffusion-linear-scheduler**, красный – **diffusion-mae**

4.2.1 Процесс обучения

Рассмотрим три модели: уже известную модель **diffusion-linear-scheduler**, **diffusion-cosinusoidal-scheduler** с косинусоидальной функцией изменения шума и **diffusion-sigmoidal-scheduler** с синусоидальной. График 4а показывает различные стратегии генерации гауссовского шума с дисперсией β^2 в прямом процессе.

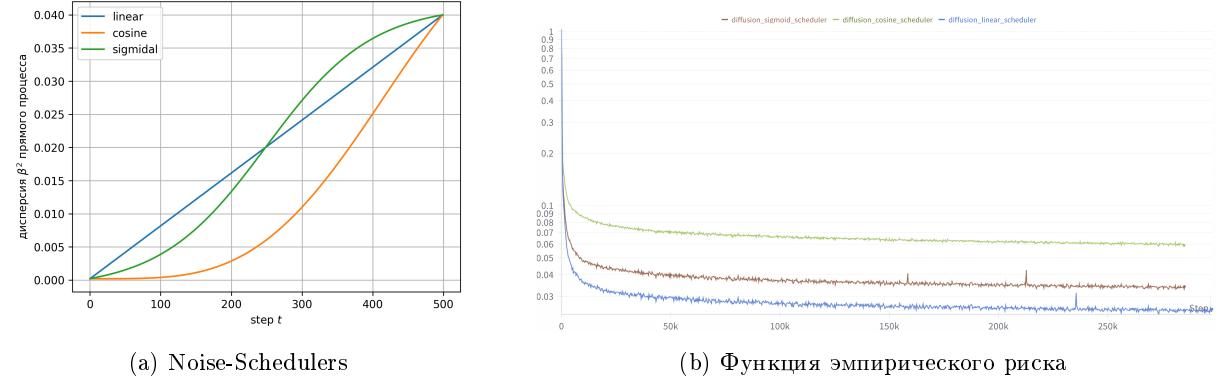


Figure 4: Сравнение динамики эмпирического риска DDPM при различных законах генерации шума: синий – **diffusion-linear-scheduler**, зеленый – **diffusion-cosinusoidal-scheduler**, коричневый – **diffusion-sigmoidal-scheduler**

По графику 4б видно, что для всех трех способов задачи шума процесс обучения носит одинаковый характер, такой же, что был введен в прошлом эксперименте. Видны стремительное падение эмпирического риска с выходом на плато до конца обучения, почти отсутствуют осцилляции значений в малых промежутках времени.

4.2.2 Сравнение метрик качества

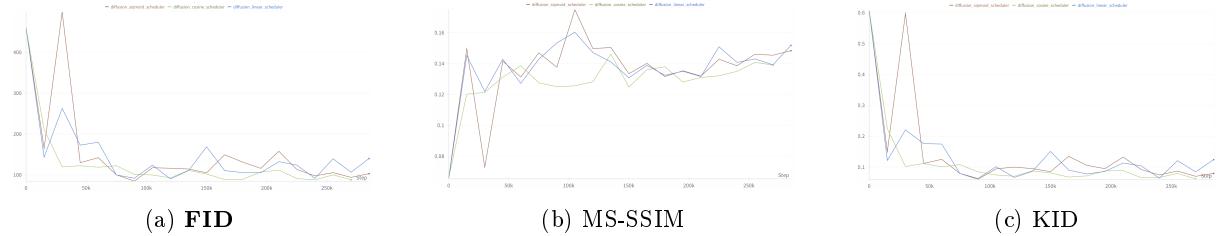


Figure 5: Сравнение метрик качества: синий – **diffusion-linear-scheduler**, зеленый – **diffusion-cosinusoidal-scheduler**, коричневый – **diffusion-sigmoidal-scheduler**

Графики оценок качества моделей (рис. 5) показывают, что применение обоих методик задания шума, отличных от линейной, уже способно улучшить качество генерации по метрикам **FID** и **KID**. Обе модели **diffusion-cosinusoidal-scheduler** и **diffusion-sigmoidal-scheduler** обошли **diffusion-linear-scheduler** по данным показателям к концу обучения. Тем не менее такая доминация не выражена на протяжении всего обучения, а проявляется ближе к концу, на графиках видны сегменты обучения, на которых **diffusion-linear-scheduler** превосходит остальные по всем критериям, при этом по метрике **MS-SSIM** обе модели не сильно проигрывают **diffusion-linear-scheduler** (данное наблюдение можно списать на внутреннюю дисперсию оценки из-за ее случайностной природы).

Способ задачи шума в модели **diffusion-sigmoidal-scheduler** приводит к самому неустойчивому обучению среди всех разбираемых моделей. На всех графиках мы видим, как данная модель обучается скачкообразно, в то время как для остальных свойственно не далеко уходить от «центра» тренда.

4.2.3 Примеры генерации

Какие проблемы использование новых шедулеров должно было решить:

1. Смещенная цветовая гамма
2. Затемненные участки либо полностью черные генерируемые изображения
3. Артефакты на изображениях

Рассмотрим примеры генерации изображений при использовании косинусоидального шедулера **diffusion-cosinusoidal-scheduler** (секция C.3). Видно что две из трех обозначенных проблемы решены: изображения выравнялись по цветовой гамме и стали больше отвечать нашим эмпирическим представлениям о том, как выглядят изображения еды, полностью решена проблема затемнения. Однако все так же присутствуют артефакты и подобие mode-collapse (четверть изображений содержит зелень, другая четверть что-то напоминающее сыр или пасту, часть изображений содержит белые продукты, часть красные или черные (как мясо)).

Модель **diffusion-sigmoidal-scheduler** показала себя хуже (секция C.4). Проблема артефактов и затемнения генерируемого изображения не решена, видно лишь отсутствие смещения палитры цветов. В остальном, ошибки генерации и плохое покрытие мод остается таким же, как и для модели **diffusion-cosinusoidal-scheduler**. Это является результатом формы сигмоидального шедулера – мы имеем плавный рост в начале и конце шума прямого процесса β_t и резкий скачок в середине, чего например не наблюдается у косинусоидального или линейного шедулера.

4.3 Построение флагманской модели

В данном эксперименте поставим задачу построить наилучшую модель DDPM, учитывая результаты предшествующих экспериментов. Рассмотрим основные выводы:

1. **MAE возможно лучше:** MAE показал, что способен модель лучше выдерживать палитру цветов изображений и возможно приводит к большей вариативности генерируемых изображений.
2. **Линейный шедулер плох:** использование любого из двух шедулеров приводит к улучшению качества генерации.

⇒ В качестве функционала возьмем MAE, отдельно обучим на косинусоидальный **diffusion-cosinusoidal-mae** и сигмоидальный **diffusion-sigmoidal-mae** шедулер, увеличив число шагов обучения в два раза.

4.3.1 Обучение

Отметим, что модель **diffusion-mae** обучалась на числе шагов вдвое меньшем, чем у двух новых. Из графиков, изображающего динамику изменения функции эмпирического риска трех моделей с функцией потерь MAE, видно, что поведение новых моделей на обучении схоже с таковыми на предыдущих шагах.

Основным различием выступает абсолютное значения функции потерь – для **diffusion-cosinusoidal-mae** таковое самое большое, для **diffusion-mae** с линейным шедулером – самое малое. Такое поведение может быть вызвано тем, что **diffusion-cosinusoidal-mae** имеет самый большой участок с малыми β_t , что означает большее число шагов денойзинга, на которых требуется предсказывать более точные значения сложных изображений.

4.3.2 Сравнение метрик качества

Сравним поведение оценок качества для двух моделей **diffusion-cosinusoidal-mae** и **diffusion-sigmoidal-mae**, приведенное на графиках 7. Видно, что относительно **diffusion-cosinusoidal-mae**, модель с сигмоидальной функцией шума менее устойчива, видны резкие скачки, в особенности в районе 435000 шага обучения. Такое поведение полностью соответствует наблюдениям

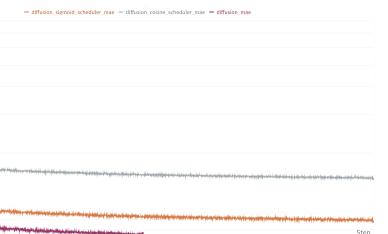


Figure 6: Сравнение обучения MAE моделей: **красный** – **diffusion-mae**, **серый** - **diffusion-cosinusoidal-mae** и **оранжевый** - **diffusion-sigmoidal-mae**

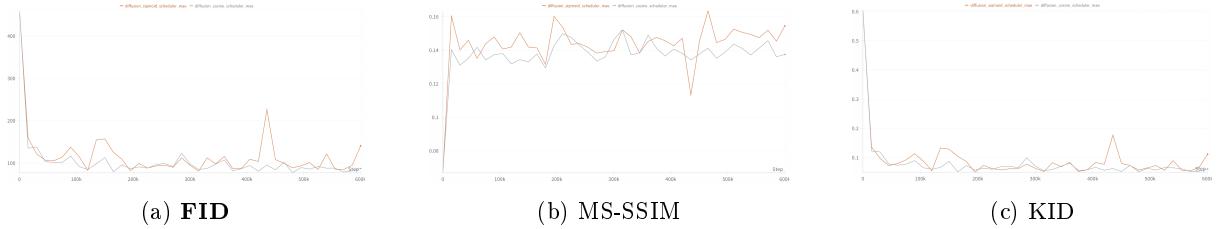


Figure 7: Сравнение метрик качества: серый - `diffusion-cosinusoidal-mae` и оранжевый - `diffusion-sigmoidal-mae`

эксперимента с MSE лоссом с кастомными шедулерами (очевидно, причина та же самая, в форме закономерности сигмоидальной функции).

Рассмотрим, что происходит с генерацией модели `diffusion-sigmoidal-mae` в районе обозначенного 435000 шага обучения. На изображении 8 видно, насколько она неустойчива в своей работе и обучении: между шагами 420000 и 450000, на которых модель способна генерировать хорошего качества изображения, на момент 435000 шага модель это полностью утрачивает (затемнение на всем изображении, неестественное расположение предметов). Такое поведение может сигнализировать о большой скоррелированности весов сети, ее переобучении.

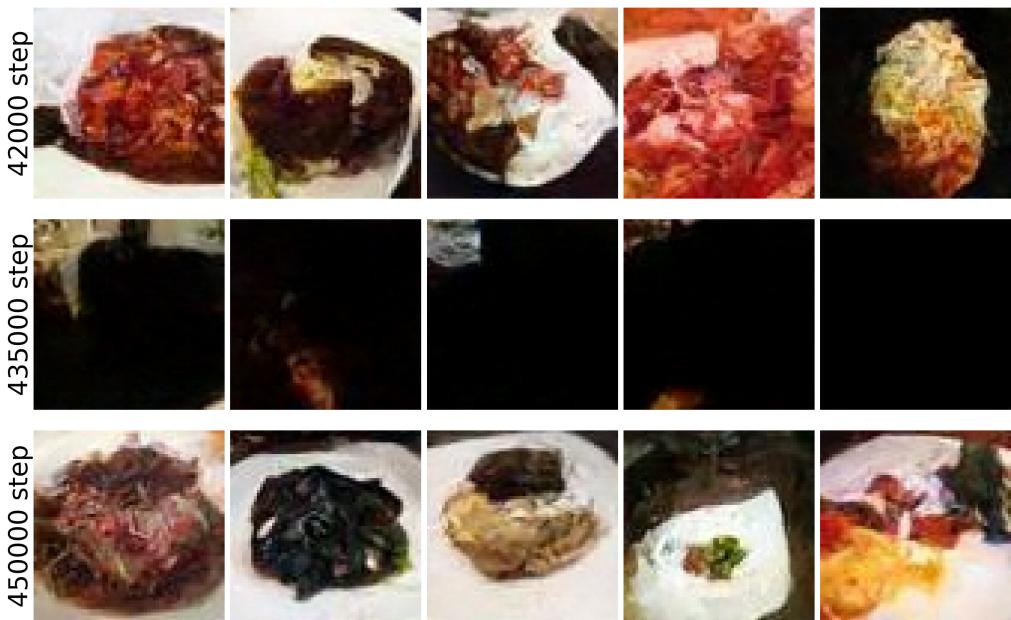


Figure 8: Примеры генерации `diffusion-sigmoidal-mae` в окрестности 435000 шага обучения

4.3.3 Примеры генерации

Примеры генерации для двух моделей приведены с сегментах C.5 и C.6. Основные замечания:

- Решена проблема смешения цветовой гаммы.
- Отсутствуют явно выраженные затемнения на изображениях.
- Уменьшено число артефактов (не удалось решить проблему полностью).
- Эмпирическая оценка показывает доминацию косинусоидального шедулера над сигмоидальным (однако крайне малую)

Можно сказать, что модель диффузии с лоссом МАЕ при косинусоидальном и сигмоидальном шедулере не различимы на глаз, хотя различие и наблюдается на метриках качества. Тем не менее, при прочих равных, по метрикам качества и поведению на обучении выбор «SOTA» модели следует отнести на `diffusion-cosinusoidal-mae`.

5 Выводы

В данном исследовании были затронуты основные способы отладки обучения DDPM, что привели к получению нескольких «SOTA» моделей. В таблице 1 приведено сравнение моделей по метрикам FID, MS-SSIM и KID после обучения. Основной метрикой исследований является **FID**, по которой выигрывает модель **diffusion-cosinusoidal-scheduler**, однако обратимся к усложненной метрике KID и уже видим доминацию модели с лоссом MAE – **diffusion-cosinusoidal-mae**. По визуальным оценкам генерируемых изображений, модель **diffusion-cosinusoidal-mae** обладает меньшим числом недостатков, нежели **diffusion-cosinusoidal-scheduler**, из-за чего принято решение в качестве результата поиска наилучшей модели DDPM привести обе модели.

Модель	Размер модели	FID	MS-SSIM	KID
diffusion-linear-scheduler	258M	140.00	0.1517	0.1239
diffusion-cosinusoidal-scheduler	258M	86.93	0.1482	0.0794
diffusion-sigmoidal-scheduler	258M	102.86	0.1323	0.1482
diffusion-mae	258M	112.82	0.1475	0.0839
<i>diffusion-cosinusoidal-mae</i>	258M	87.18	0.1375	0.0612
<i>diffusion-sigmoidal-mae</i>	258M	141.10	0.1546	0.1119

Table 1: Сравнение моделей по метрикам FID, MS-SSIM и KID

A Архитектура генеративной модели. DDPM

Диффузионные модели (Diffusion Probabilistic Models, DDPM) представляют собой современный подход к генерации данных, впервые предложенный в статье Denoising Diffusion Probabilistic Models. Эти модели основаны на пошаговом добавлении и последующем удалении шума из данных в предположении, что наблюдаемые значения есть результат процесса последовательного зашумления гауссовским шумом, что позволяет эффективно аппроксимировать сложные распределения.

Основная идея модели: процесс генерации DDPM состоит из двух этапов:

1. **Прямой процесс VP (forward-pass variance-preserving):** Преобразование данных $x_0 \in \mathcal{I}$ в сильно зашумлённое состояние x_T путём последовательного добавления гауссовского шума с малым шагом амплитуды α_t :

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I) \quad t \in 1, \dots, T \text{ - шаги шума}$$

2. **Обратный процесс (backward-pass):** Восстановление данных из зашумлённого состояния x_T путём поэтапного удаления шума. Такое возможно из условия $|\alpha_t - \alpha_{t-1}| \ll 1$, что гарантирует нормальное распределение оператора перехода в backward-pass:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad \theta \text{ - параметры модели}$$

Что в свою очередь может быть переписан как:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} f_\theta(x_t, t) \right)$$

A.1 Архитектура модели

DDPM включает в себя нейросеть f_θ , обучаемую восстановлению шума ϵ_t из зашумлённых данных x_t :

$$\mathcal{L}(\theta) = \mathbb{E}_{\{x_0 \sim \mathcal{I}, \epsilon, t\}} [\|\epsilon - f_\theta(x_t, t)\|^2]$$

В качестве архитектуры нейросети f_θ будем использовать классический выбор – U-net, способную эффективно сочетать глобальный и локальный контексты благодаря специальному устройству Residual Connection, поддерживая выходную размерность равной входной (рис. 9). Так как процесс денойзинга своими параметрами принимает как x_t – состояние объекта на конкретном шаге, так и параметр времени t , будем использовать процедуру смешения временных эмбеддингов.

Для этого зададим e_t – набор синусоидальных позиционных эмбеддингов, что далее будут использоваться в качестве аддитивной добавки к промежуточному выходу внутри каждого Down и Up блоков («Time-Blending»).

$$x_t^{k+1} = \text{Convolution}(\text{MLP}_k(e_t) + \text{Projection}_k(x_t^k))$$

B Оценки качества

Подробное описание с причинами использования оценок качества.

B.1 Fréchet Inception Distance

FID измеряет различие между распределениями признаков реальных (P_r) и сгенерированных (P_g) изображений. Признаки извлекаются из предпоследнего слоя предобученной на *ImageNet* модели *Inception-v3*, после чего вычисляется расстояние Фреше при предположении о нормальном распределении признаков (расстояние между двумя многомерными гауссовыми распределениями).

Реализация метрики взята из библиотеки torch-fidelity.

DDPM Architecture Unet

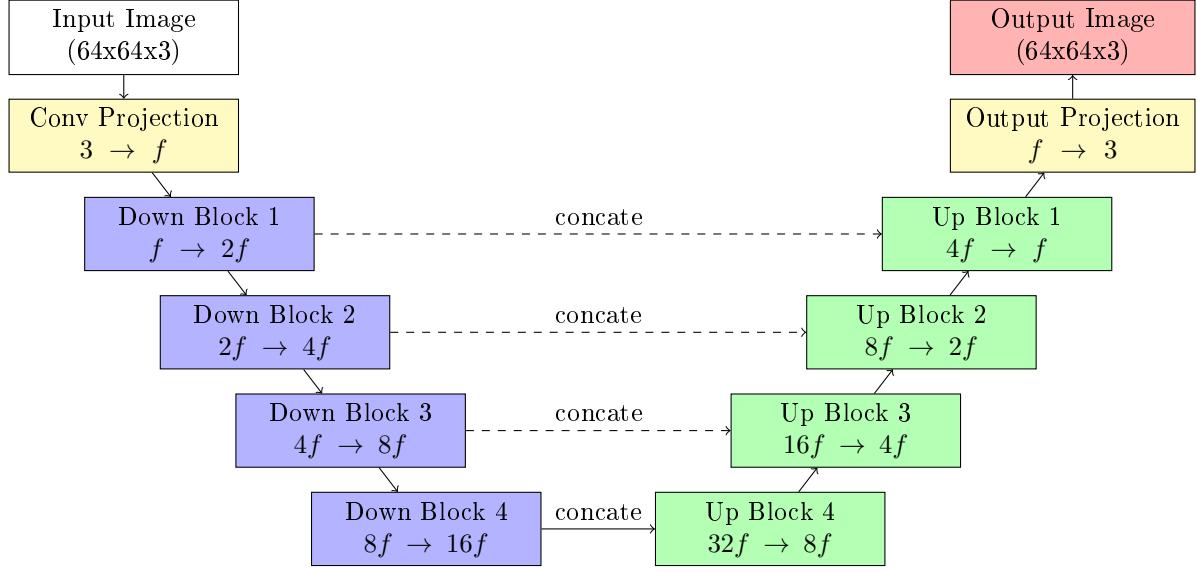


Figure 9: Архитектура DDPM с остаточными соединениями и блоками проекции.

B.2 MS-SSIM

MS-SSIM (Multi-Scale Structural Similarity Index) является расширением популярной метрики SSIM (Structural Similarity Index Measure), учитывающим различия между изображениями на нескольких пространственных масштабах. Это повышает чувствительность метрики к мелким и крупным артефактам. MS-SSIM сравнивает близость изображений не по технической части, а по «семантической», из-за чего даже плохое значение метрики (малые значения) могут указывать не сколько на плохую генерацию, сколько на то, что генерируемые изображения слишком отличны от изначальных. Данная метрика также основывается на предобученных моделях, однако использует интерпретируемые признаки.

По указанным причинам было принято решение дополнить оценку качества FID метрикой MS-SSIM, учитывая ее потенциально большую чувствительность к артефактам изображений.

Реализация метрики взята из библиотеки Lightning.ai - TorchMetrics.

B.3 Kernel-Inception Distance

KID (Kernel Inception Distance) — основана на идеи FID, тем не менее не использует предположение об распределении признаков изображений. Вместо этого используется приближение KDE с Гауссовским ядром признаков изображений с предобученной модели *Inception-v3*. После чего KID вычисляет максимальное среднеквадратичное отклонение

Данная метрика принята дублировать основную метрику FID, в виду ее независимости от возможно неверного предположения о гауссовском распределении признаков.

Реализация метрики взята с библиотеки torch-fidelity

C Примеры генерации

C.1 diffusion-linear-scheduler

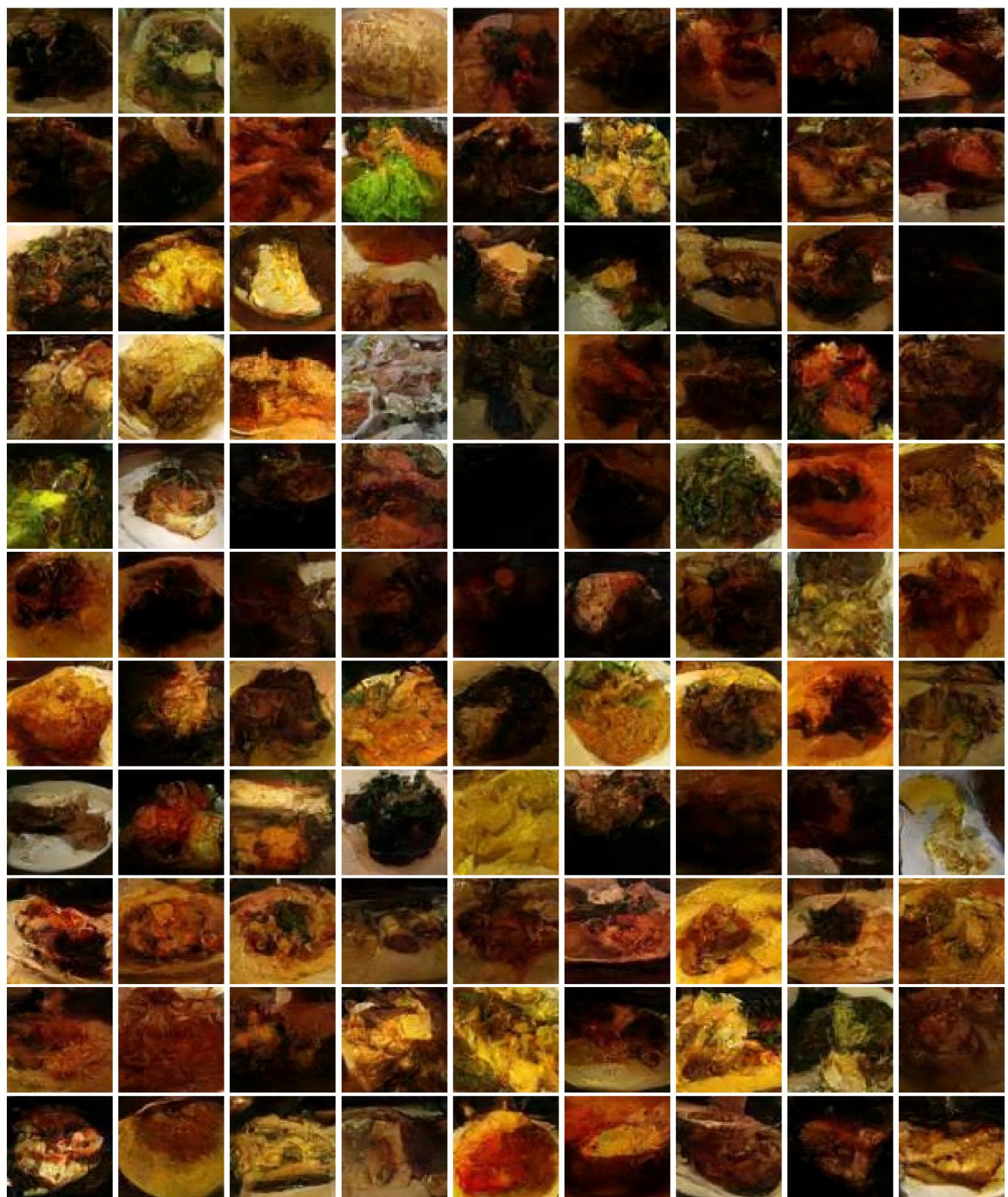


Figure 10: Пример 99 изображения сгенерированного **diffusion-linear-scheduler**

C.2 diffusion-mae

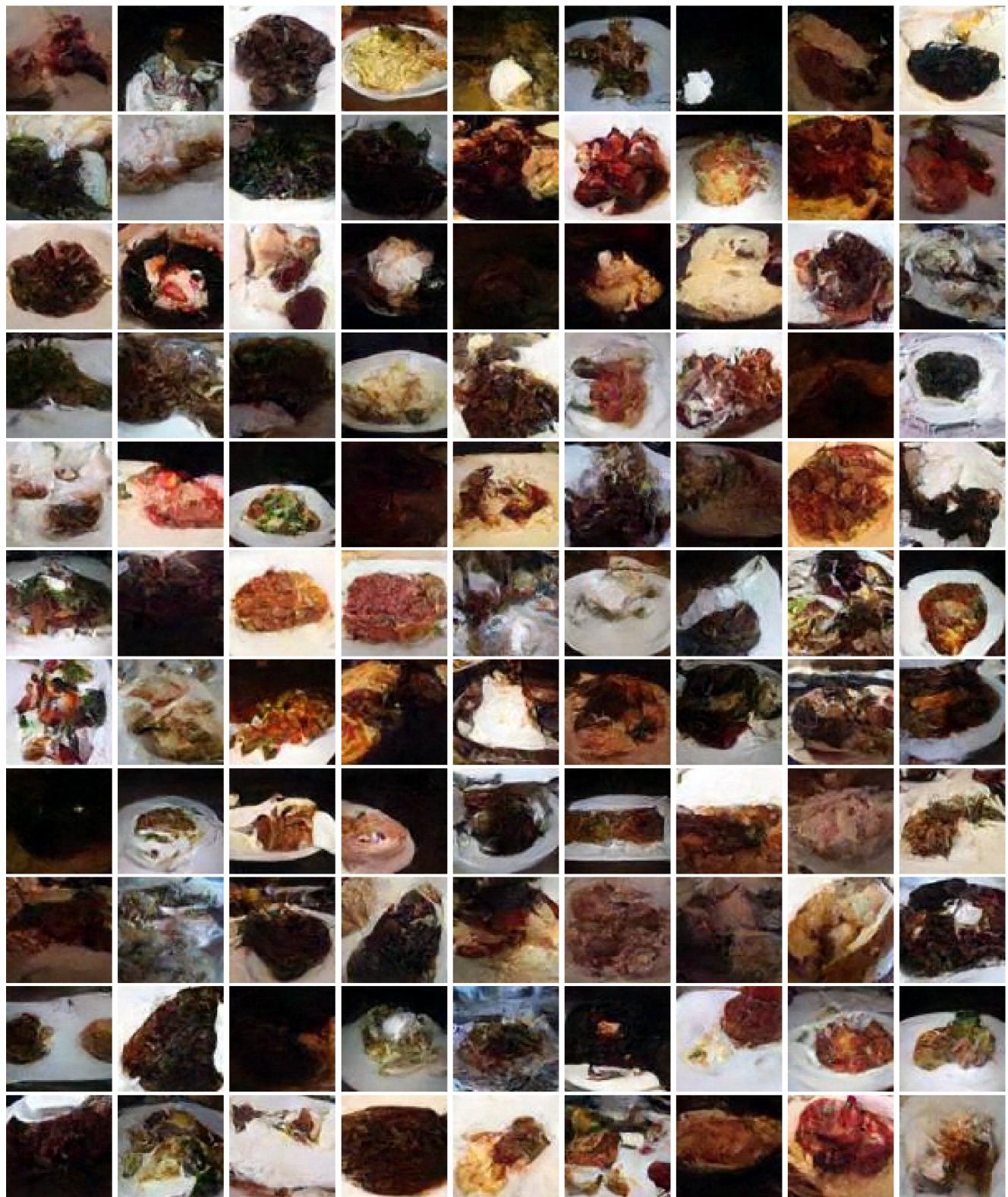


Figure 11: Пример 99 изображения сгенерированного **diffusion-mae**

C.3 diffusion-cosinusoidal-scheduler



Figure 12: Пример 99 изображения сгенерированного **diffusion-cosinusoidal-scheduler**

C.4 diffusion-sigmoidal-scheduler

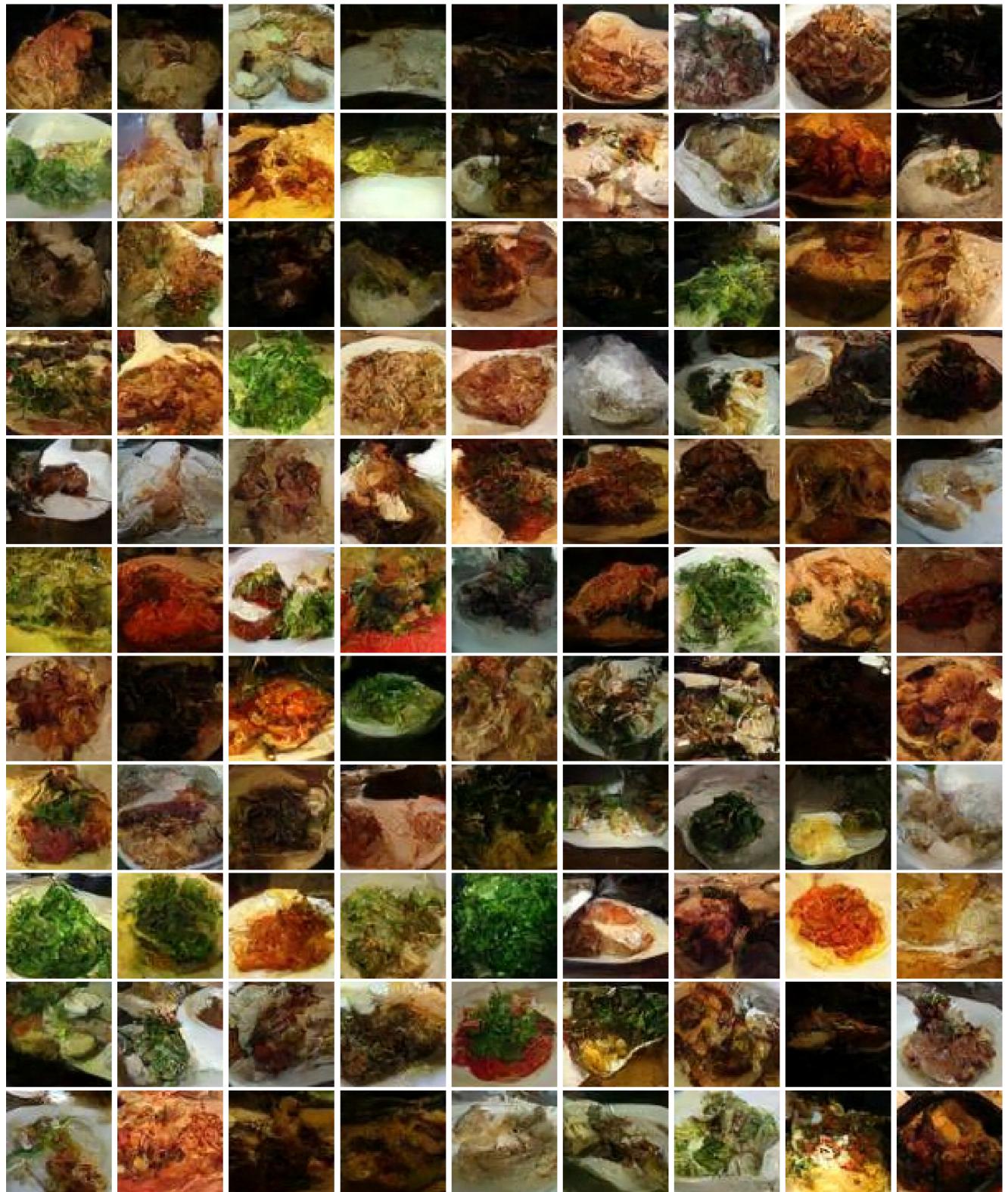


Figure 13: Пример 99 изображения сгенерированного **diffusion-sigmoidal-scheduler**

C.5 diffusion-sigmoidal-mae

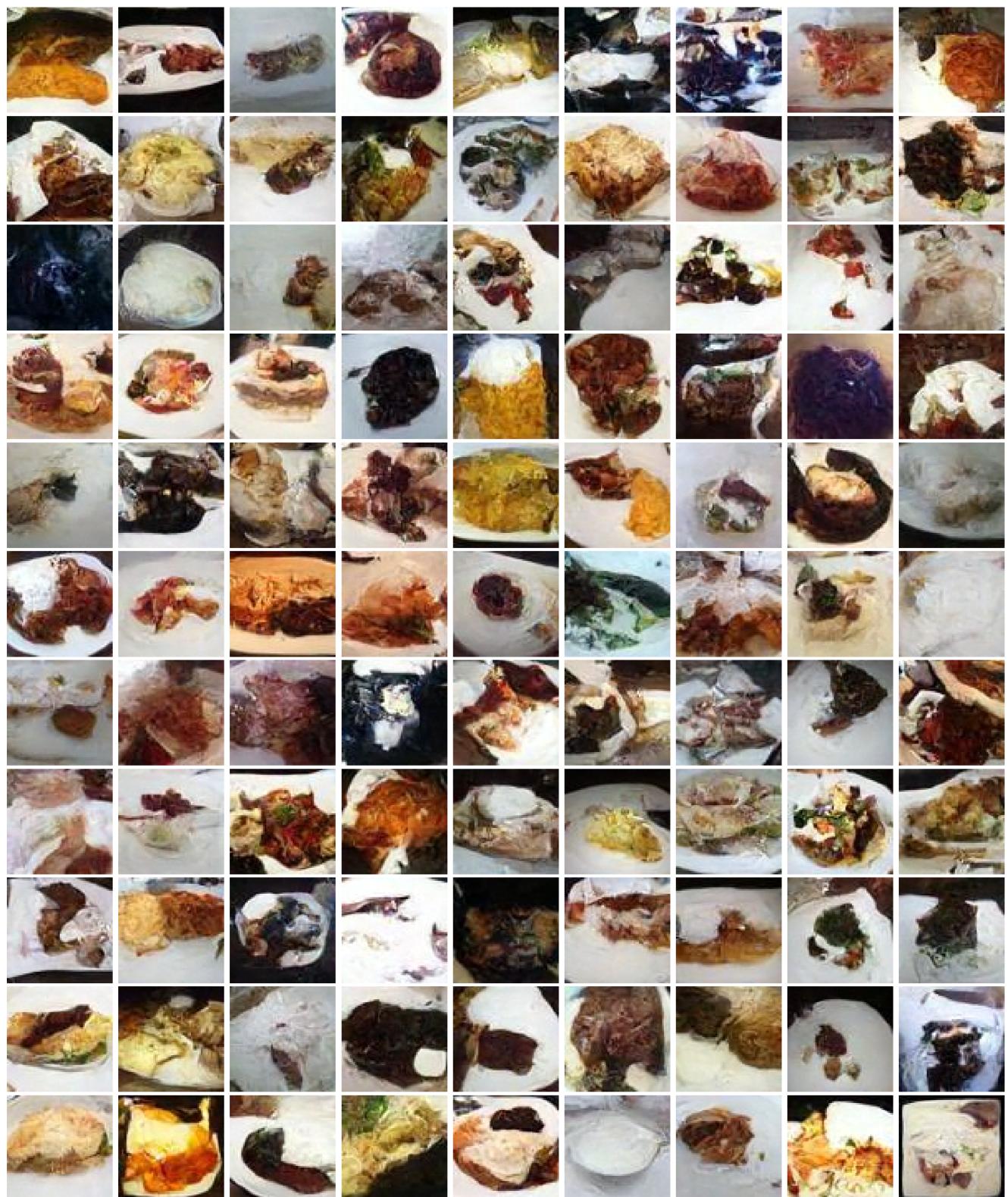


Figure 14: Пример 99 изображения сгенерированного **diffusion-sigmoidal-mae**

C.6 diffusion-cosinusoidal-mae

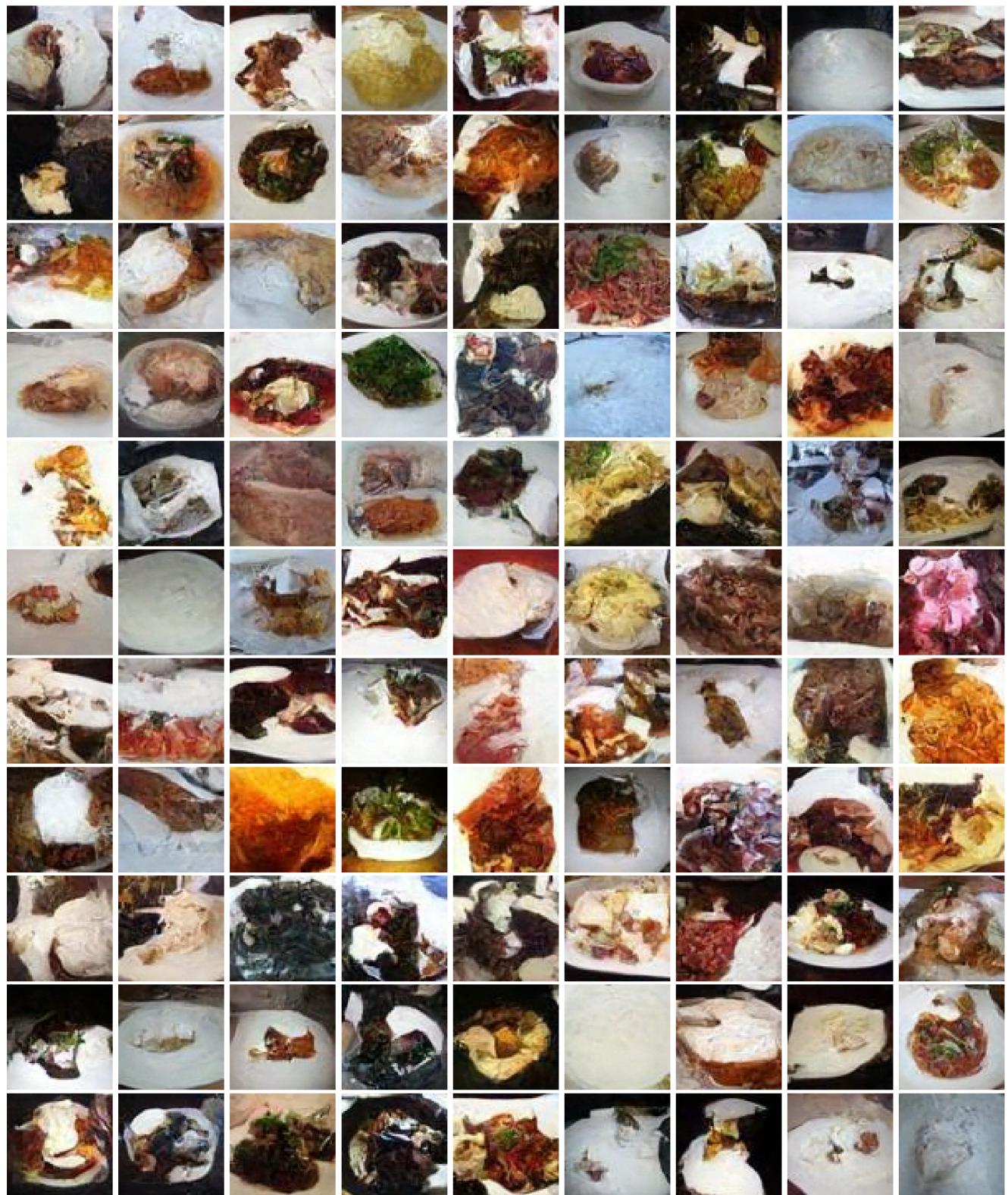


Figure 15: Пример 99 изображения сгенерированного **diffusion-cosinusoidal-mae**