Project Generate Music Using Discrete Diffusion
Deep Generative Models Course 361.2.2370

Students: Rom Hirsch, Omer Aviv, Omer Pilosof, Moshe Buchris
Lecturer: Dr. Eliya Nachmani

February 5, 2025

Abstract

This paper introduces a novel approach to music generation by leveraging token-based representations and a Discrete Diffusion Model (DDM). The project addresses the challenge of efficiently modeling and generating music audio while maintaining quality and coherence. Using WavTokenizer [4], a pre-trained music model, we convert audio into discrete tokens via an encoder and reconstruct audio from these tokens through a decoder. To model the joint probability distribution of tokens, we adopt the DDM framework, optimized with the innovative Score Entropy Discrete Diffusion (SEDD) loss function. Unlike prior methods such as Multi-source Diffusion [7], this approach uniquely integrates token representations with diffusion-based modeling.

The model is trained on the MTG-Jamendo dataset [1], a high-quality resource for music signal analysis. Performance evaluation involves the objective Fréchet Audio Distance (FAD) metric and subjective listening tests to assess the quality and coherence of the generated music. Despite challenges such as potential information loss in tokenization and the novelty of the DDM methodology, we aim to achieve performance comparable to or exceeding existing methods.

Compare to MSDM [8], our final trained model produced significantly higher-quality music samples, demonstrating the efficacy of token-based representations and discrete diffusion models in advancing music synthesis technologies. code at `https://github.com/romhirsch/SEDD-music`

1 Problem Description

This study focuses on the compression of music audio into discrete tokens, followed by the generation of music based on these tokens. By employing this approach, we aim to enhance the overall performance of the exist music generation model [7]. The use of tokens provides a structured and efficient representation of the audio data, facilitating better processing, modeling, and output quality in subsequent tasks.

2 Novelty of method with respect to the literature

The novelty of this work lies in the use of token-based representations for music, which provide a structured and compact way to model audio data. Unlike conventional approaches, this work employs a DDM in combination with the innovative SEDD loss function. This is a departure from prior methods such as Multi-source Diffusion [7], which do not leverage token representations or the advantages of SEDD. By integrating these techniques, our approach introduces a unique framework for music generation, enabling improved performance and efficiency in handling complex audio structures.

3 Datasets

The DDM training process used 3,600 audio files from the MTG-Jamendo dataset [1], a comprehensive and high-quality resource designed for tasks involving analysis and synthesis of music signals. This data set provides diverse and rich audio samples, allowing the DDM to effectively learn the joint probability distribution of token representations and ensure robust performance in the generation of music. The MTG-Jamendo dataset's extensive coverage of instrumental and compositional variations makes it an ideal choice for training models focused on token-based music generation.

4 Chosen Method for Solving

To address this problem, we will start with encode the dataset in to tokens. For the music generation process, we will learn the joint probability distribution of the tokens using a Discrete Diffusion Model (DDM) [6]. The DDM employs a novel loss function known as Score Entropy Discrete Diffusion (SEDD), which is specifically designed to optimize the diffusion process in discrete token spaces. The training of the DDM will be conducted using the MTG-Jamendo dataset [1], a high-quality dataset for music signal modeling. During inference, the trained DDM will be used to generate new samples by sampling novel token sequences from the learned distribution. These sampled tokens will then be passed through the pretrained speech model's decoder, which will transform the token sequences back into audio, resulting in the generation of music. This approach combines the robustness of pre-trained speech models with the generative power of DDM, enabling high-quality music synthesis. The entire method process is represented in Figure 1.

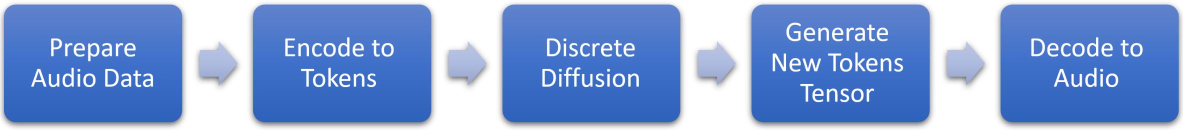

Figure 1: Method Process

4.1 Music To Tokens

we propose to utilize WavTokenizer [4], a pre-trained music model that includes both an encoder and a decoder. The encoder will convert audio signals into discrete tokens (each second to 75 tokens for 24kHz audio), while the decoder will reconstruct audio signals from these tokens. Every token is 12 bits long. The encoder compresses audio with convolutional layers and LSTMs using Vector Quantization, a single quantizer with expanded codebook for compression. The decoder use Inverse Fourier Transform and attention mechanisms ensure high-quality reconstruction. Additionally, the model use Advanced Discriminator, a Multi-scale discriminators, to improve perceptual quality.

4.2 Discrete Diffusion Model (DDM)

Standard diffusion models are grounded in the well-established theory of score matching. However, attempts to extend this approach to discrete structures have faced challenges and have not demonstrated comparable empirical improvements. The Discrete Diffusion Model (DDM) [**1**] addresses this limitation by introducing score entropy—a novel loss function that naturally adapts score matching to discrete spaces. This innovative approach seamlessly integrates into the development of discrete diffusion models, leading to significant performance enhancements.

4.2.1 DDM Process

The DMM forward and revised based on continuous-time Markov process given by a linear ordinary differential equation:

$$\frac{d\mathbf{p}_t}{dt} = Q_t \mathbf{p}_t, \quad \mathbf{p}_0 \approx \mathbf{p}_{\text{data}}, \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^N$ is probability mass vectors over a finite support $\mathcal{X} = \{1, \ldots, N\}$ and $\mathbf{p}_t \in \mathbb{R}^N$ family of distributions according to the a continuous time Markov process

$$Q_t = \sigma(t)Q, \tag{2}$$

where Q_t are the diffusion matrices in $\mathbb{R}^{N \times N}$. These matrices satisfy the Non-diagonal entries of Q_t are non-negative and columns which sum to zero. where $\sigma(t)$ is a scalar function of level noise.

One can simulate this process using small Euler steps of size Δt, combined with random sampling of the resulting transitions:

$$p(x_{t+\Delta t} = y \mid x_t = x) = \delta_{xy} + Q_t(y, x)\Delta t + O(\Delta t^2) \tag{3}$$

where δ_{xy} is the Kronecker delta function.

This process has a well-known time-reversal counterpart, characterized by another diffusion matrix Q_{T-t}:

$$\frac{dp_{T-t}}{dt} = Q_{T-t} p_{T-t} \tag{4}$$

where Q_{T-t} defined as:

$$Q_{T-t}(y, x) = \frac{p_t(y)}{p_t(x)} Q_t(x, y), \quad Q_t(x, x) = -\sum_{y \neq x} Q_t(y, x) \tag{5}$$

we can see that for the reverse process we need to learning the ratios $\frac{p_t(y)}{p_t(x)}$, The ratios are collectively referred to as the concrete score [9], a generalization of the conventional score function $\nabla_x \log p_t$.

4.2.2 Score Entropy Discrete Diffusion

The goal of a discrete diffusion model is to construct the aforementioned reverse process by learning the ratios $\frac{p_t(y)}{p_t(x)}$. The article represent $\mathcal{L}_{DSE}$ Denoising Score Entropy and show that the optimal θ that minimizes the $\mathcal{L}_{DSE}$ satisfies $s_{\theta*} = \frac{p(y)}{p(x)}$ for all pairs x, y.

$$\mathcal{L}_{DSE} = \mathbb{E}_{x_0 \sim p_0, x \sim p(\cdot|x_0)} \left[\sum_{y \neq x} w_{xy} \left(s_\theta(x)y - \frac{p(y|x_0)}{p(x|x_0)} \log s_\theta(x)y \right) \right] \tag{6}$$

For time dependent score $s(\cdot, t)$ the article introduce $\mathcal{L}_{DWDSE}$ diffusion weighted denoising score entropy (DWDSE) loss:

$$\int_0^T \mathbb{E}_{x_t \sim p_t(\cdot|x_0)} \left[\sum_{y \neq x} Q_t(x_t, y) \left(s_\theta(x,t)_y - \frac{p_{t|0}(y|x_0)}{p_{t|0}(x|x_0)} \log s_\theta(x,t)_y \right) \right] dt \tag{7}$$

where $s_\theta(x,t) \approx \frac{p_t(y)}{p_t(x)}_{y \neq x}, \quad s_\theta : \mathcal{X} \times \mathbb{R} \to \mathbb{R}^{|\mathcal{X}|}$

4.2.3 Practical Implementation

Given sequences $\mathcal{X} = \{1, \ldots, N\}$ and $x = x^1 \ldots x^d$ (e.g., sequences of tokens), computing $\frac{p(y)}{p(x)}$ directly is computationally infeasible due to the sheer number of ratios, which scales as $O(N^{2d})$. To address this computational limitation, only ratios between sequences with Hamming distance 1 are considered, reducing the complexity to $O(Nd)$ ratios.

$$s_\theta(x^1 \ldots x^i \ldots x^d, t)_{i, \hat{x}^i} \approx \frac{p(x^1 \ldots \hat{x}^i \ldots x^d)}{p(x^1 \ldots x^i \ldots x^d)} \tag{8}$$

To compute $\mathcal{L}_{DWDSE}$, it is necessary to calculate the forward transition $p^{\text{seq}}_{t|0}(\cdot|\cdot)$:

$$p^{\text{seq}}_{t|0}(\hat{x}|x) = \prod_{i=1}^{d} p^{\text{tok}}_{t|0}(\hat{x}_i|x_i) \tag{9}$$

where

$$p^{\text{tok}}_{t|0}(\cdot|x) = \text{x-th column of } \exp(\bar{\sigma}(t) Q^{\text{tok}}) \tag{10}$$

In the article, two standard matrices with special structures are proposed for efficiently computing the loss: the diffusion matrices Q^{uniform} and Q^{absorb}. These matrices enable a more efficient and scalable implementation for practical applications.

5 Training and Sampling

5.1 Training

In this study, we convert audio data from the Music Audio Dataset into token sequences using the algorithm described in Algorithm (1). Each audio file is represented as a sequence of tokens $x = \{x^1, \ldots, x^d\}$, where each token corresponds to an element in the vocabulary $\mathcal{X} = \{1, \ldots, N\}$. For training the Denoising Diffusion Model (DDM), we use the training procedure outlined in Algorithm 1.

The model is trained with the following parameters:

Parameter	**Value**
Batch size	128
Vocabulary size	4095
Sampling method	Euler method
Q	absorb
Time steps	128
Optimizer	Adam
learning rate	3×10^{-4}
Iterations	$150e3$
Model size	Small

Table 1: Training Parameters

Due to computational limitations, we have focused on the Euler sampling method, the absorption of Q, and the small model size. This approach allows us to effectively evaluate the model under these specific settings.

Algorithm 1 Score Entropy Training Loop

```
Require: Network $s_\theta$, noise schedule $\sigma$ (total noise $\sigma$), data distribution $p_{\text{data}}$, token transition matrix $Q$, time $[0, T]$
  Sample $x_0 \sim p_0$, $t \sim U([0, T])$
  Construct $x_t$ from $x_0$. In particular, $x_t^i \sim p_{t|0}(\cdot|x_0^i) = \exp(\sigma(t)Q)_{x_0^i}$
  if $Q$ is Absorb then
    $x_t^i = e^{-\sigma(t)} x_0^i + (1 - e^{-\sigma(t)}) e_{\text{MASK}}$
  else if $Q$ is Uniform then
    $x_t^i = e^{\sigma(t)-1} n e^{\sigma(t)} 1 + e^{-\sigma(t)} x_0^i$
  end if
  Compute the loss function:
    $\mathcal{L}_{\text{DWDSE}} = \sigma(t) \sum_{i=1}^{d} \sum_{y=1}^{n} \left(1 - \delta_{x_t^i(y)}\right) \left( s_\theta(x_t, t)_{i,y} - \frac{p_{t|0}(y|x_0^i)}{p_{t|0}(x_t^i|x_0^i)} \log s_\theta(x_t, t)_{i,y} \right)$
  Backpropagate $\nabla_\theta \mathcal{L}_{\text{DWDSE}}$
  Run optimizer
```

5.2 Sampling

Sample algorithm present in algorithm (2), The process iteratively updates the sequence by decrementing a time index t until $t = 0$. For each step, transition densities $p_i(y|x_i^t)$ are computed. The updated sequence $x_{t-\Delta t}$ is then constructed from these values, and the time step is decremented by Δt.

6 Metrics

The performance of the proposed model was evaluated using an objective metric known as the Fréchet Audio Distance (FAD), which measures the similarity between the distri-

Algorithm 2 Score Entropy Sampling (Unconditional)

```
Require: Network $s_\theta$, noise schedule $\sigma$ (total noise $\sigma$), token transition matrix $Q$, time $[0, T]$, step size $\Delta t$.
  Sample $x_T \sim p_{\text{base}}$ by sampling each $x_T^i$ from the stationary distribution of $Q$.
  $t \leftarrow T$
  while $t > 0$ do
    if Using Euler then
      Construct transition densities $p_i(y|x_t^i) = \delta_{x_t^i}(y) + \Delta t \cdot Q_{\text{tok}}^t(x_t^i, y) s_\theta(x_t, t)_{i,y}$.
    else if Using Tweedie Denoising then
      Construct transition densities
        $p_i(y|x_t^i) = exp((\bar{\sigma}(t - \Delta t) - \bar{\sigma}(t))Q) s_\theta(x_t, t)_{i,y} \exp((\bar{\sigma}(t) - \bar{\sigma}(t - \Delta t))Q)(x_i^t, y).$
    end if
    Normalize $p_i(\cdot|x_i^t)$ (clamp values to be at least 0 and renormalize the sum to 1 if needed).
    Sample $x_{t-\Delta t}^i \sim p_i(y|x_t^i)$ for all $i$, constructing $x_{t-\Delta t}$ from $x_{t-\Delta t}^i$.
    $t \leftarrow t - \Delta t$
  end while
  return $x_0$
```

butions of generated and real audio in a perceptual feature space:

$$\text{FAD} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}) \tag{11}$$

where:

- μ_r, Σ_r are the mean and covariance matrix of the reference (real) data features.
- μ_g, Σ_g are the mean and covariance matrix of the generated (model) data features.
- Tr denotes the trace of a matrix.

To estimate the distribution of a particular audio set, we used 2 well-known methods: VGGish [3] and PANN [5]. The data we created are compared to two sets from known datasets, MTG-Jamendo [1] we trained on and Nlakh [2]. Our goal was to achieve performance that either matches or exceeds the results of the Multi-source Diffusion method [7]. Furthermore, we wanted to get closer in performance to the MG^2 [10] used in the Chat-GPT 4-o model. Both models were likely trained for longer and on larger data sets. The results demonstrate the effectiveness of our approach in generating high-quality, coherent music audio using token-based representations and the DDM framework.

7 Results

We created a set of 300 audio files that we Generated. We extracted the VGGish distributions of our set and sets from the MTG-Jamendo [1] and Nlakh [2] datasets. We compared them using (11). The results of our model compared to MSDM [8] and MG^2 [10], who presented the results in their articles using only VGGish, is shown in Table 2.

FAD	MSDM	MG^2	Our SEDD
VGGish *(relative to Jamendo)*	**6.55**	–	**3.13**
VGGish *(relative to Nlakh)*	–	**0.99**	**1.16**
PANN *(multiply by* 10^3*)*	–	–	**0.101**

Table 2: Results

The study evaluated the performance of the proposed model (SEDD) compared to previous methods such as the Multi-source Diffusion Model (MSDM) [8] and MG^2 [10], using the FAD metric. The results indicate that our new model achieved **lower FAD values** compared to MSDM, demonstrating a significant improvement in music generation quality:

- Relative to **MTG-Jamendo**, the model achieved **3.13**, compared to **6.55** for MSDM – a substantial enhancement in audio quality.
- Relative to **Nlakh**, the model scored **1.16**, close to **0.99** for MG^2 – indicating performance comparable to larger and more complex models.
- In the **PANN** metric, which assesses acoustic similarity, the model scored **0.101** $\times$ **10**$^-$3, suggesting strong preservation of original sound characteristics.

These findings demonstrate that leveraging **token-based representations** and the **Discrete Diffusion Model (DDM)** leads to a **notable improvement in music generation quality**. The new method **narrows the gap with MG**^{2}, a more advanced and large-scale model, proving the effectiveness of this approach in generating coherent and high-quality music.

Summary

This paper presents a novel approach to music generation by leveraging token-based representations and a Discrete Diffusion Model (DDM). The proposed method converts music audio into discrete tokens using WavTokenizer and reconstructs it using a decoder. To model the probability distribution of these tokens, the study introduces the Score Entropy Discrete Diffusion (SEDD) loss function, optimizing the diffusion process in discrete token spaces. The model is trained on the MTG-Jamendo dataset and evaluated using FAD from quality audio sets. Results demonstrate that the proposed approach outperforms existing methods, such as Multi-source Diffusion (MSDM), and generates high-quality music samples.

Future possible developments include adding conditional inputs, allowing users to specify the desired genre or musical style, enhancing control over the generated music. Additionally, classical signal processing techniques, such as noise reduction and energy-based voice activity detection (VAD), will be integrated to optimize audio clarity and structure. Another possible feature could be that the approach will be extended beyond music generation to handle more complex tasks, such as speech synthesis and image-based audio generation, further broadening the scope and applicability of the system.

References

[1] Won Bogdanov et al. “The MTG-Jamendo Dataset for Automatic Music Tagging”. In: *ICML* (2019).

[2] Minju Choi et al. “Show Me the Instruments: Musical Instrument Retrieval from Mixture Audio”. In: *arXiv preprint arXiv:2211.07951* (2022).

[3] Shawn Hershey et al. “Cnn Architectures For Large-Scale Audio Classification”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 131–135.

[4] Shengpeng Ji et al. “WavTokenizer: An Efficient Acoustic Discrete Codec Tokenizer for Audio Language Modeling”. In: *arXiv preprint arXiv:2408.16532* (2024).

[5] Qiuqiang Kong et al. “PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2019), pp. 2880–2894. DOI: 10.1109/TASLP.2020.3030497.

[6] Aaron Lou, Chenlin Meng, and Stefano Ermon. “Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution”. In: *Forty-first International Conference on Machine Learning*. 2024.

[7] Giorgio Mariani et al. “Multi-source diffusion models for simultaneous music generation and separation”. In: *arXiv preprint arXiv:2302.02257* (2023).

[8] Giorgio Mariani et al. “Multi-source diffusion models for simultaneous music generation and separation”. In: *arXiv preprint arXiv:2302.02257* (2023).

[9] Chenlin Meng, Yang Song, and Stefano Ermon. “Concrete score matching: Generalized score matching for discrete data”. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 34532–34545.

[10] Manzhen Shaopeng et al. “Melody is all you need for music generation”. In: *arXiv preprint arXiv:2409.20196* (2024).