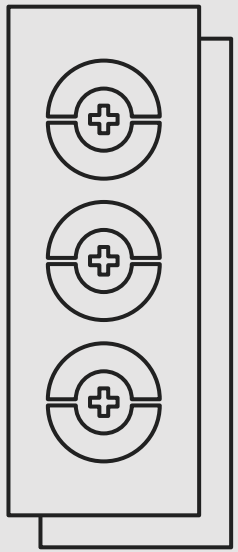
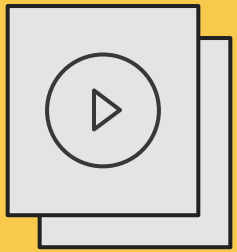
JUKEBOX

A GENERATIVE MODEL FOR MUSIC



Докладчик: Солодуха Мария
Рецензент: Добряев Иван
Хакер: Аъзам Бехруз

JUKEBOX



- OpenAI 2020
- Генеративная модель, способная генерировать музыку в различных жанрах и стилях
- Идея: обработать и сжать длинные необработанные аудиовходы с помощью многоуровневого автоэнкодера и уменьшить размерность, но сохранить важную музыкальную информацию.

ВИКТОРИНА 1

01 FRANK SINATRA

02 KANYE WEST

03 THE BEATLES

ВИКТОРИНА 1

01 FRANK SINATRA

02 KANYE WEST

03 THE BEATLES

ВИКТОРИНА 2

01 T.A.T.U

02 CELINE DION

03 LINKIN PARK

ВИКТОРИНА 2

01 T.A.T.U

02 CELINE DION

03 LINKIN PARK

BACKGROUND

МУЗЫКА

- Это непрерывная волна
  $x \in [-1, 1]^T$
- T = длительность ✕ частота дискретизации (16 - 48 кГц)

BACKGROUND

МУЗЫКА

- Это непрерывная волна $x \in [-1, 1]^T$
- T = длительность ✕ частота дискретизации (16 - 48 кГц)

CD

- 44,1 кГц
- 16 бит
- Аудиосегмент продолжительностью 4 минуты будет иметь входную длину 10 млн

BACKGROUND

МУЗЫКА

- Это непрерывная волна $x \in [-1, 1]^T$
- T = длительность ✕ частота дискретизации (16 - 48 кГц)

CD

- 44,1 кГц
- 16 бит
- Аудиосегмент продолжительностью 4 минуты будет иметь входную длину 10 млн

КАРТИНКИ

- RGB-изображение с высоким разрешением 1024 × 1024 пикселей имеет входную длину ≈3 миллиона, и каждая позиция содержит 24 бита информации

BACKGROUND

МУЗЫКА

- Это непрерывная волна $x \in [-1, 1]^T$
- T = длительность ✕ частота дискретизации (16 - 48 кГц)

CD

- 44,1 кГц
- 16 бит
- Аудиосегмент продолжительностью 4 минуты будет иметь входную длину 10 млн

КАРТИНКИ

- RGB-изображение с высоким разрешением 1024 × 1024 пикселей имеет входную длину ≈3 миллиона, и каждая позиция содержит 24 бита информации

ИТОГО

изучение генеративной модели музыки чрезвычайно трудоемкий процесс

VQ-VAE (vector quantized variational autoencoder)

- Одномерный VQ-VAE учится кодировать входную последовательность $x = \langle x_t \rangle^T_{t=1}$, используя последовательность дискретных токенов $z = \langle z_s \in [K] \rangle^S_{s=1}$, (где K обозначает размер словаря, а отношение T/S - длина перехода.
- Он состоит из:
  - энкодера E(x), который кодирует x в последовательность скрытых векторов $h = \langle h_s \rangle^S_{s=1}$
  - Bottleneck – квантует h_s в e_{z_s} путем сопоставления каждого h_s с его ближайшим соседом – вектором e_{z_s} из кодовой книги $C = \{e_k\}^K_{k=1}$
  - декодера D(e), который декодирует векторы вложения обратно во входное пространство.

VQ-VAE

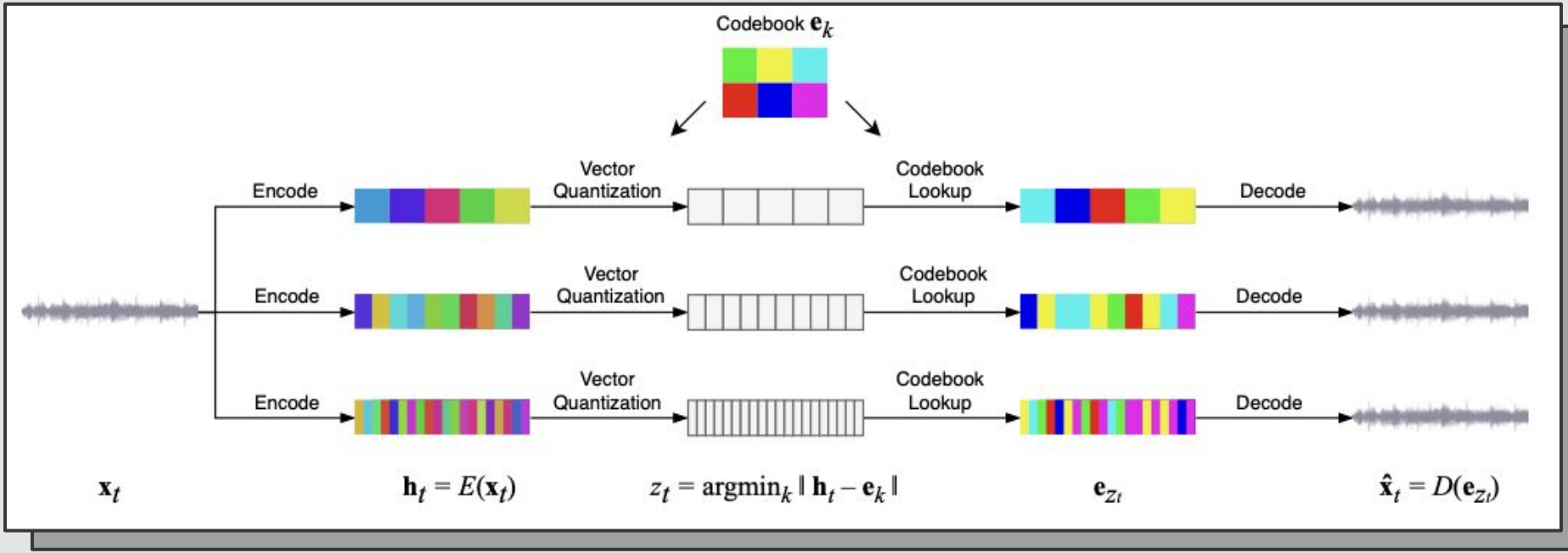
$$\mathcal{L} = \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{codebook}} + \beta\mathcal{L}_{\text{commit}} \tag{1}$$

$$\mathcal{L}_{\text{recons}} = \frac{1}{T}\sum_{t} \|\mathbf{x}_t - D(\mathbf{e}_{z_t})\|_2^2 \tag{2}$$

$$\mathcal{L}_{\text{codebook}} = \frac{1}{S}\sum_{s} \|\text{sg}\left[\mathbf{h}_s\right] - \mathbf{e}_{z_s}\|_2^2 \tag{3}$$

$$\mathcal{L}_{\text{commit}} = \frac{1}{S}\sum_{s} \|\mathbf{h}_s - \text{sg}\left[\mathbf{e}_{z_s}\right]\|_2^2 \tag{4}$$

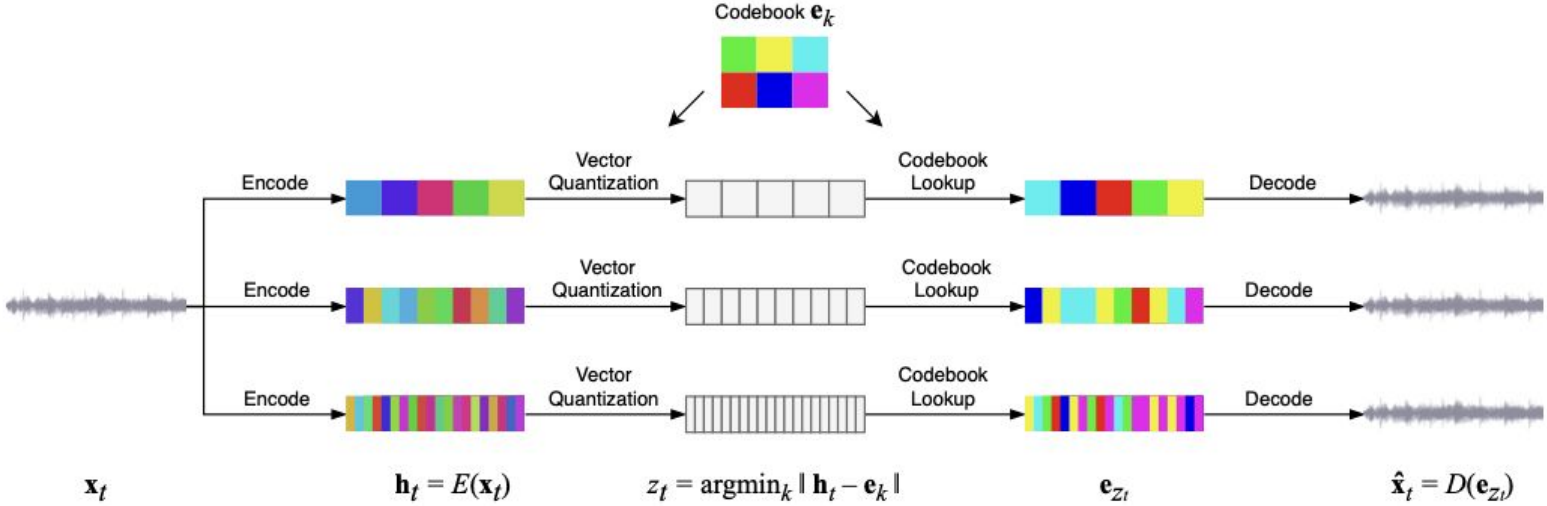
MUSIC VQ-VAE


Codebook $\mathbf{e}_k$
Encode
Vector Quantization
Codebook Lookup
Decode
Encode
Vector Quantization
Codebook Lookup
Decode
Encode
Vector Quantization
Codebook Lookup
Decode
$\mathbf{x}_t$
$\mathbf{h}_t = E(\mathbf{x}_t)$
$z_t = \text{argmin}_k \| \mathbf{h}_t - \mathbf{e}_k \|$
$\mathbf{e}_{z_t}$
$\hat{\mathbf{x}}_t = D(\mathbf{e}_{z_t})$

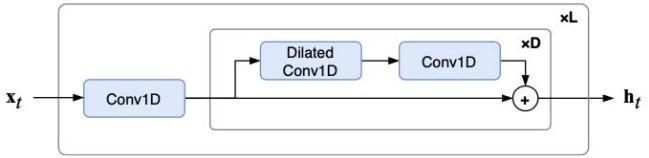
PRIOR AND UPSAMPLERS

$$p(\mathbf{z}) = p(\mathbf{z}^{\text{top}}, \mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{bottom}}) \tag{5}$$

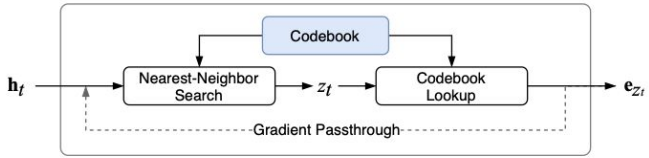
$$= p(\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{middle}}|\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{bottom}}|\mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{top}}) \tag{6}$$



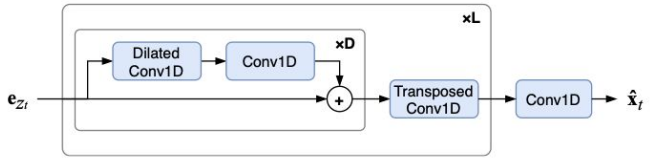
COMPONENTS OF THE VQ-VAE MODEL



(a) The encoder compresses the raw audio input into a sequence of embeddings. The length of this latent representation relative to the raw audio duration determines the amount of compression, and is an important factor for the trade-off between fidelity and coherence.

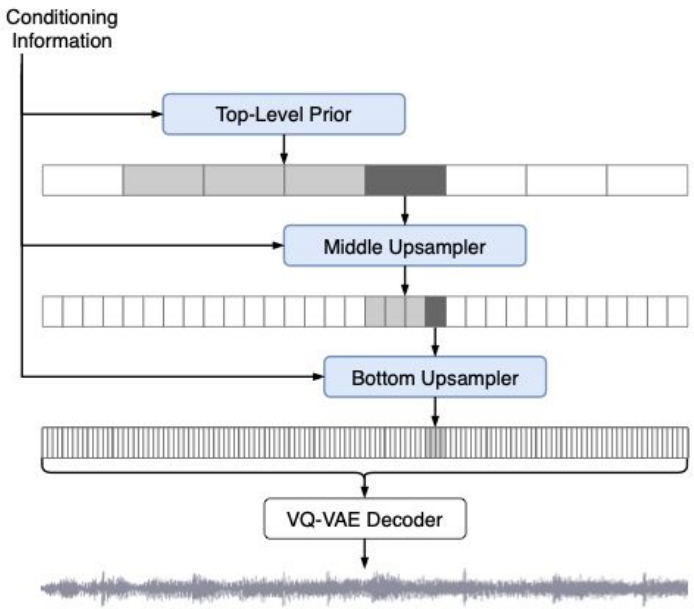


(b) The bottleneck takes the sequence of embeddings from the encoder and maps it into a sequence of code vectors from the codebook. This sequence of code indices is used as a discrete representation to be modeled by the priors. Larger codebooks improve fidelity but may be more difficult to compress.



(c) The decoder reconstructs the raw audio from latent representations. It is a mirror of the encoder where dilations constracts by a factor of 3 down to 1 at the last block. The final Conv1D projects to the desired number of audio channels and also acts as a smoothing operation after a sequence of transposed convolutions.

SAMPLING METHODS



(a) **Ancestral sampling**: Priors for the VQ-VAE codes are trained using a cascade of Transformer models, shown in blue. Each model takes conditioning information such as genre, artist, timing, and lyrics, and the upsampler models are also conditioned on the codes from the upper levels. To generate music, the VQ-VAE codes are sampled from top to bottom using the conditioning information for control, after which the VQ-VAE decoder can convert the bottom-level codes to audio.

ARTIST, GENRE, AND TIMING CONDITIONING

Генеративную модель можно сделать более управляемой, предоставляя дополнительные условные сигналы во время обучения

LYRICS CONDITIONING

- Авторы предоставляют больше контекста во время обучения, настраивая модель на тексты песен, соответствующие каждому аудиосегменту, позволяя модели воспроизводить пение одновременно с музыкой.
- AutoLyricsAlign для выравнивания текстов

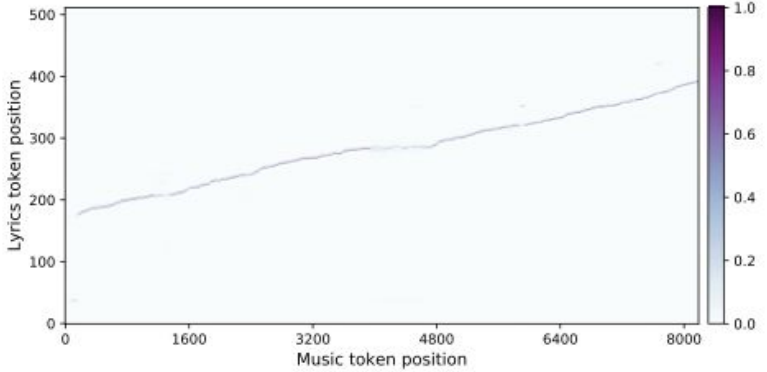
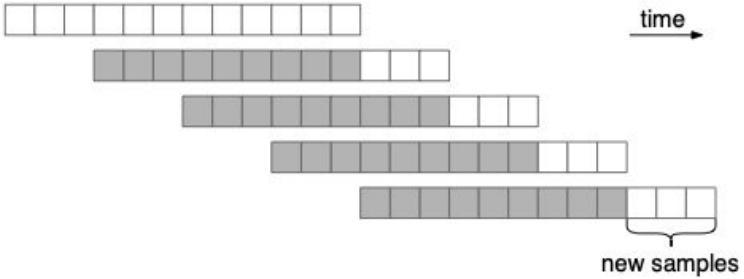
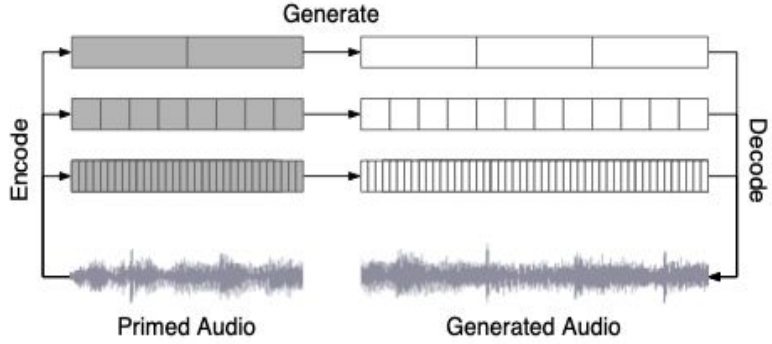


Figure 3. Lyrics-singing alignment learned by one of the encoder-decoder attention layers. The x-axis is the position of music queries, and the y-axis is the position of lyric keys. The positions attended to by the decoder correspond to the characters being sung.

SAMPLING METHODS

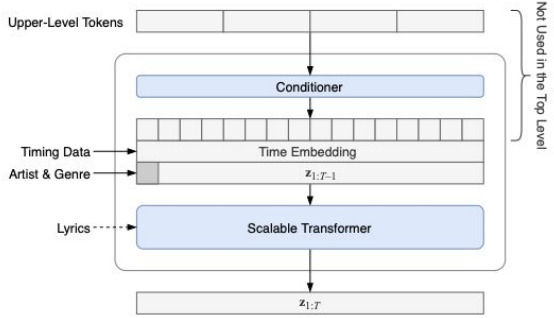


(b) **Windowed sampling**: To generate music longer than the model's context length (12 in this figure), we repeatedly sample continuations at each level, using overlapping windows of previous codes as the context. The overlap amount is a hyperparameter, and the figure shows an example of 75% overlap with hop length 3.

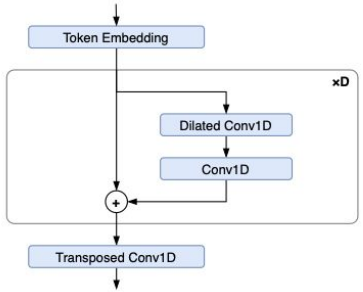


(c) **Primed sampling**: The model can generate continuations of an existing audio signal by converting it into the VQ-VAE codes and sampling the subsequent codes in each level.

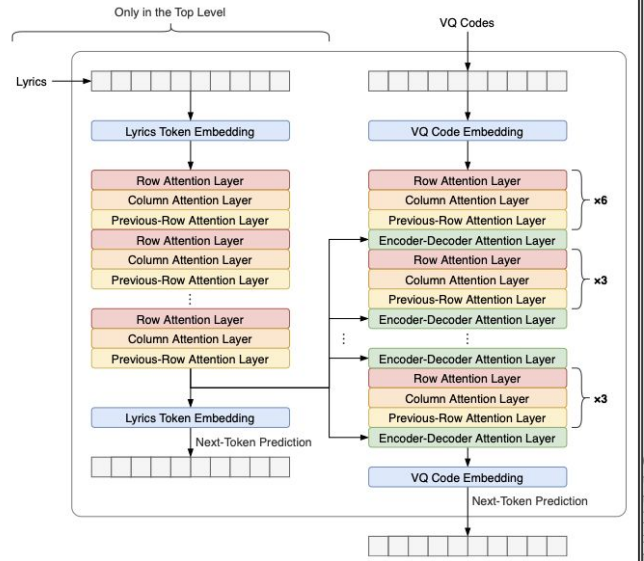
DETAILED ARCHITECTURE OF THE MUSIC PRIOR AND UPSAMPLER MODELS



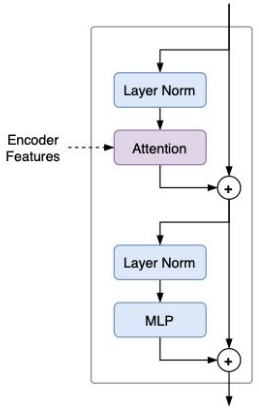
(a) The structure of our prior models, performing next-token prediction at each level. The Transformer takes the embeddings of the tokens $\mathbf{z}_{1:T-1}$ prepended by the sum of the artist and genre embeddings, in addition to the time embedding that encodes relative and absolute timing of the segments in the duration of the song. The upsampler priors additionally take the tokens from the upper level, which are fed to the conditioner network and added to the input sequence. The top-level prior takes lyrics as conditioning information as well (see Figure 8c).



(b) The conditioner network takes the tokens from the upper level, and their embedding vectors go through non-causal WaveNet-like layers with increasingly dilated convolutions. The transposed 1-D convolution upsamples the sequence to the higher temporal resolution of the current level.



(c) The Scalable Transformer architecture, shown with the lyrics Transformer used in the top-level prior. The Transformer layers use the three factorized attention types alternatingly, i.e. repeating row, column, and previous-row attentions. In the top-level prior, the VQ Transformer additionally includes interleaved encoder-decoder attention layers that apply lyrics conditioning by attending on the activation of the last encoder layer.



(d) Each Transformer layer is a residual attention block, which performs two residual operations, attention and MLP, each prepended with layer normalization. Depending on the layer's type, it uses either one of the three factorized attentions or encoder-decoder attention taking the lyrics features from the encoder.

ДАННЫЕ

- Авторы собрали набор данных из 1,2 миллиона песен в сочетании с текстами песен и метаданными из LyricWiki.
- Метаданные:
  - Исполнитель
  - Альбом
  - Жанр
  - Год выпуска
  - Общие настроения или ключевые слова списка воспроизведения, связанные с каждой песней.

РЕЗУЛЬТАТЫ

ЦЕЛОСТНОСТЬ

Сэмплы остаются очень когерентными в музыкальном плане на протяжении всего контекста

МУЗЫКАЛЬНОСТЬ

- Сэмплы часто имитируют знакомые музыкальные гармонии
- Часто самые высокие или продолжительные ноты мелодии соответствуют словам, которые певец-человек предпочел бы подчеркнуть

НОВЫЕ СТИЛИ

создание песни в необычном жанре, обычно не связанном с исполнителем

НОВЫЕ ТЕКСТЫ

Авторы просили Jukebox спеть стихи и новые куплеты, созданные GPT-2, в результате продемонстрировали, что модель действительно может петь новые тексты

ДАЛЬНЕЙШАЯ РАБОТА

УЛУЧШЕНИЕ КАЧЕСТВА

- Хорошее музыкальное произведение должно быть **высококачественным** во всех временных масштабах: оно должно иметь развивающуюся музыкальную и эмоциональную структуру во всем произведении
- Избавление от нежелательного шума
- Ускорение работы

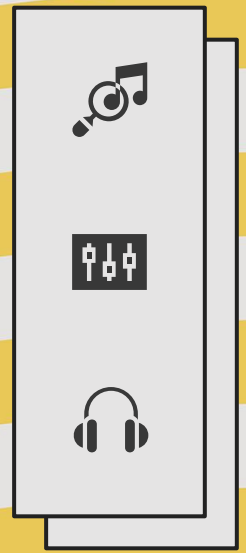
НОВЫЕ МУЗЫКАЛЬНЫЕ СТРУКТУРЫ

- Внедрение припевов и куплетов, вопросно-ответной формы
- Улучшение качества дуэтов
- Разнообразить языки и стили
- Внедрение новых параметров (настроение, динамика)

СПАСИБО !

Авторы представили Jukebox - модель, которая генерирует необработанную музыку, имитирующую множество различных стилей и исполнителей, а также при желании можно указать текст для сэмпла.

Модель способна генерировать фрагменты продолжительностью в несколько минут с узнаваемым пением естественными голосами.



ЛИТЕРАТУРА

JUKEBOX

https://arxiv.org/pdf/2005.00341.pdf

VQ-VAE

Oord et al., 2017; Dieleman et al., 2018; Razavi et al., 2019

EXAMPLES

https://jukebox.openai.com/?song=787633465

NUS AUTOLYRICSALIGN

Gupta et al., 2020

THIS IS A TIMELINE

9AM
DAY 1
Mars is actually a cold place

10AM
DAY 2
Saturn is a gas giant and has several rings

2PM
DAY 3
Venus has a beautiful name

4PM
DAY 4
Mercury is a very small planet

6PM
DAY 5
Jupiter is the biggest planet of them all