2020 OpenAI
Jukebox: A Generative Model for Music

Presenter: Yu Cheng-Hung
Thesis Advisor：Jian-Jiun Ding
Group meeting: 2022/01/18

Outline

1.Introduction

2.Background

3.Music VQ-VAE

4.Music Priors and Upsamplers

5.Result

6.Conclusion

1.

Introduction

Fake or Real Music

Art def: 「new, surprising, and has value as it is stimulating debate and interest」, By margaret boden

Introduction

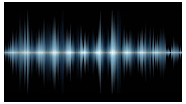
Which music is real ?

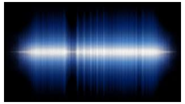
Completions

Re-renditions

Unseen lyrics

Introduction





Input: lyrics, genre, artist etc.

output: a new song
In raw audio domain

Models can produce highly diverse genres.

Introduction

Use

1. A hierarchical VQ-VAE architecture
   - for compress audio into a discrete space
2. Autoregressive scalable transformer
   - for top prior model training
3. Autoregressive upsampler
   - recreate the lost information

2.

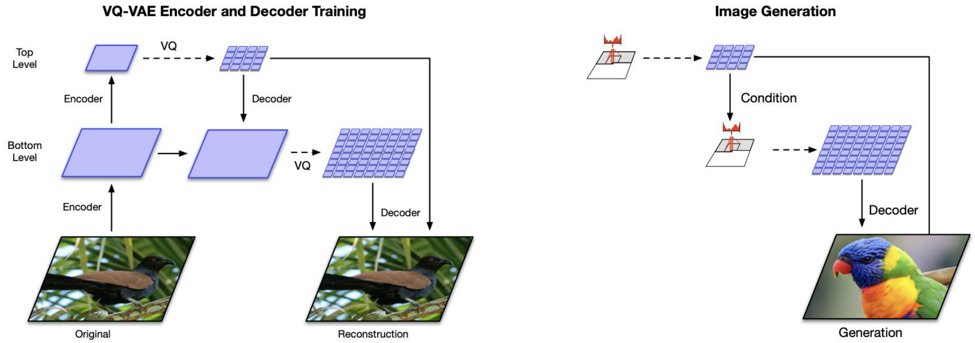
Background

Background

Raw audio → Continuous Waveform $x \in [-1, 1]^T$ → # T depends on audio duration & sampling rate

CD 44.1kHz with 4 mins → 10 million length input

VQ-VAE2



Background

VQ-VAE term

input sequence, $x =< x_t >_{t=1}^{T}$

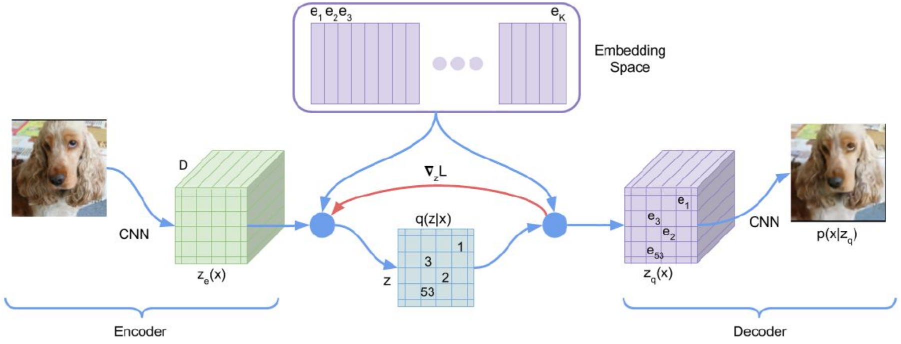
token $z =< z_s \in [K]_{s=1}^{S} >$, K: vocabulary size, T/S: hop length

VQ-VAE structure is composed of encoder, VQ, and decoder

$E(x)$: encoder，$x \xrightarrow{encode} h =< h_s >_{s=1}^{S}$, h is latent vector

VQ: Quantize the h and mapping $h_s \longrightarrow e_{z_s}$, e_{z_s} is the embedding vectors

$D(e)$: decoder，embedding vectors $\xrightarrow{decode}$ input space



Background

$$\mathcal{L} = \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{codebook}} + \beta\mathcal{L}_{\text{commit}} \quad (1)$$

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

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

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

(2) means the distance between input and output

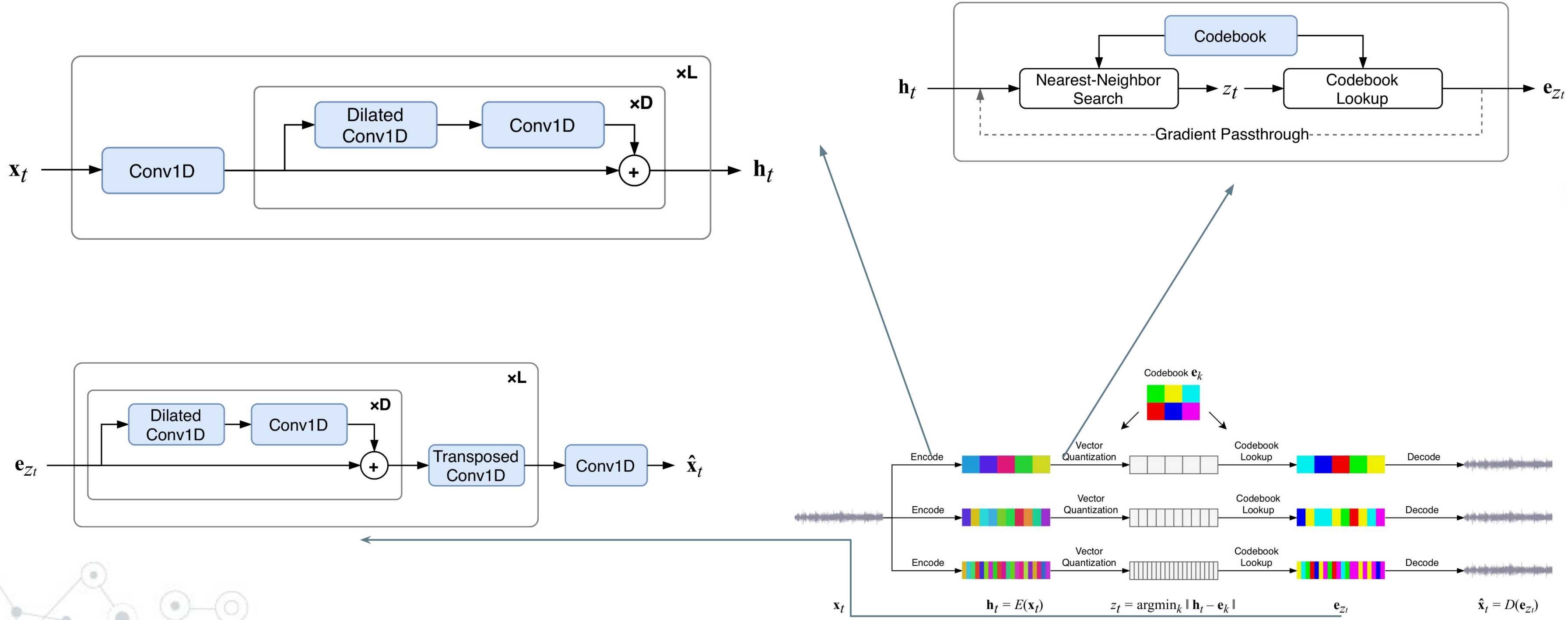
(3) means the distance between encoding and nearest neighbors from the codebook

(4) prevent the encodings from fluctuating too much. To stabilize the encoder

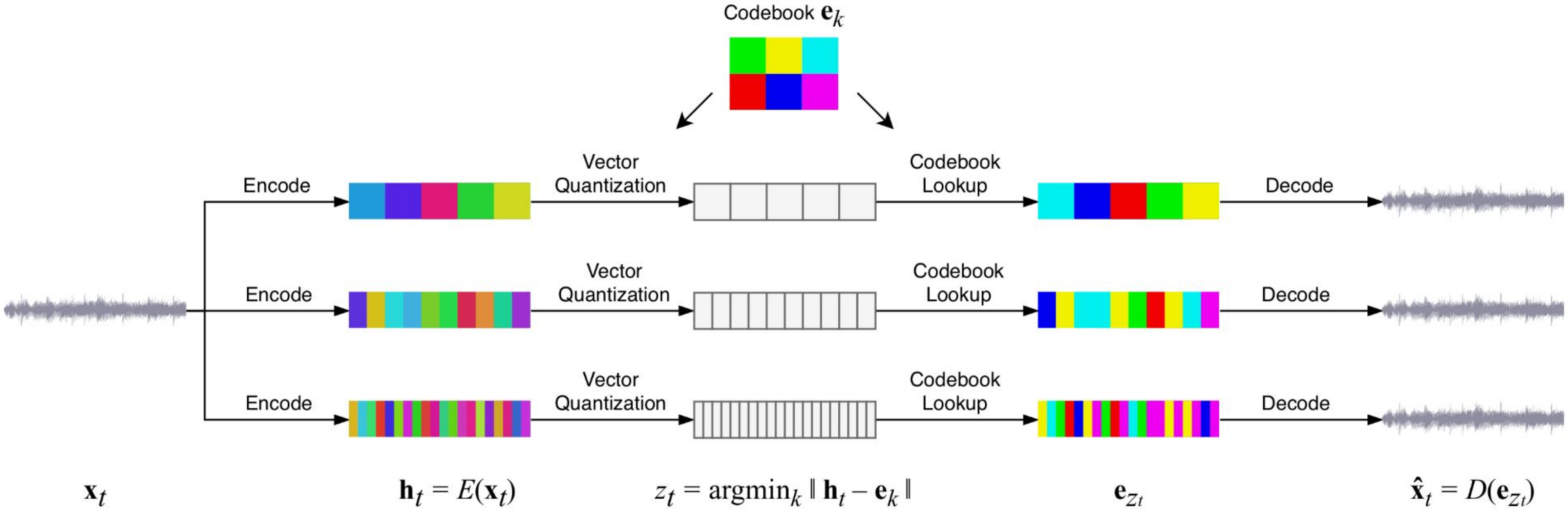
3.

Music VQ-VAE

Components of Music VQ-VAE



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
$\mathbf{x}_t$
$\mathbf{h}_t = E(\mathbf{x}_t)$
$z_t = \text{argmin}_k \| \mathbf{h}_t - \mathbf{e}_k \|$
$\mathbf{e}_{z_t}$
$\hat{\mathbf{x}}_t = D(\mathbf{e}_{z_t})$

Some modification from VQ-VAE to Music VQ-VAE

Random restarts for embeddings

- VQ-VAEs are known to suffer from codebook collapse
- Sol: Use random restarts => If mean usage of a codebook vector falls below threshold, then reset the vector to one of encoder outputs from current batch
- The solution mitigating codebook collapse.

Separated Autoencoders

- The bottlenecked top level is little used in Image VQ-VAE2
- Sol: three level autoencoder and train separated

Spectral loss $L_{spec} = |||STFT(x)| - |STFT(\hat{x})|||_2$

- Sample level reconstruction loss -> Model only learns to reconstruct low frequency
- Sol: Add spectral loss and encourage model to match spectral components
- The solution enable model to reconstruct mid-to-high frequency

4.

Music Priors and Upsamlers

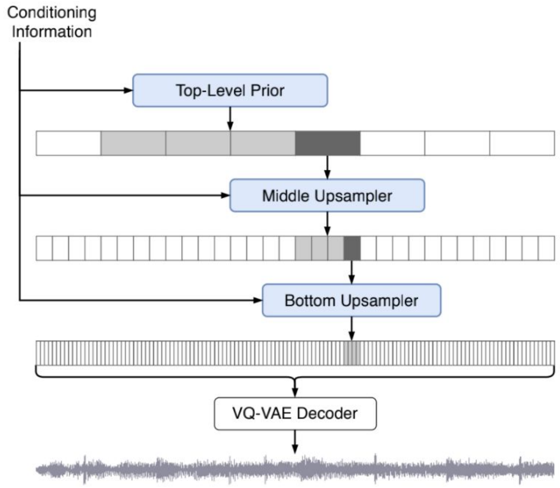
Music Priors and Upsamplers

- Prior and upsamplers model formula

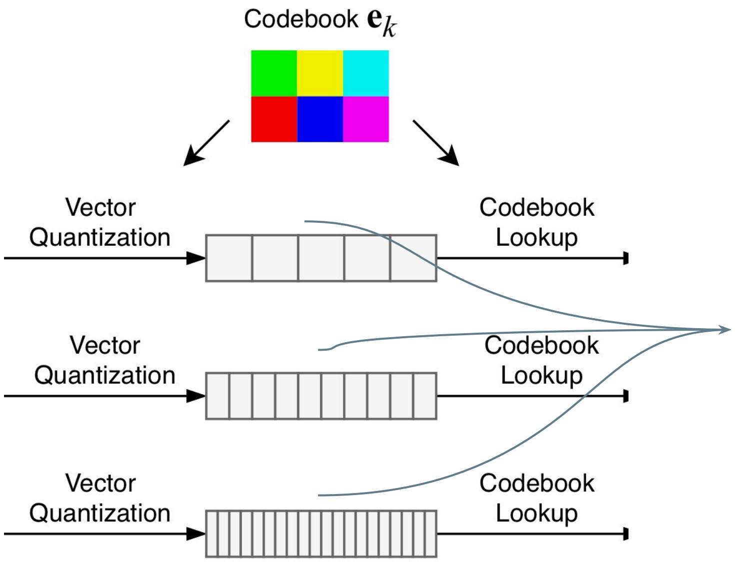
$$p(z) = p(z^{top}, z^{middle}, z^{bottom}) = p(z^{top})p(z^{middle}|z^{top})p(z^{bottom}|z^{middle}, z^{top})$$

  - top-level prior $p(z^{top})$
  - upsamplers $p(z^{middle}|z^{top})$ and $p(z^{bottom}|z^{middle}, z^{top})$

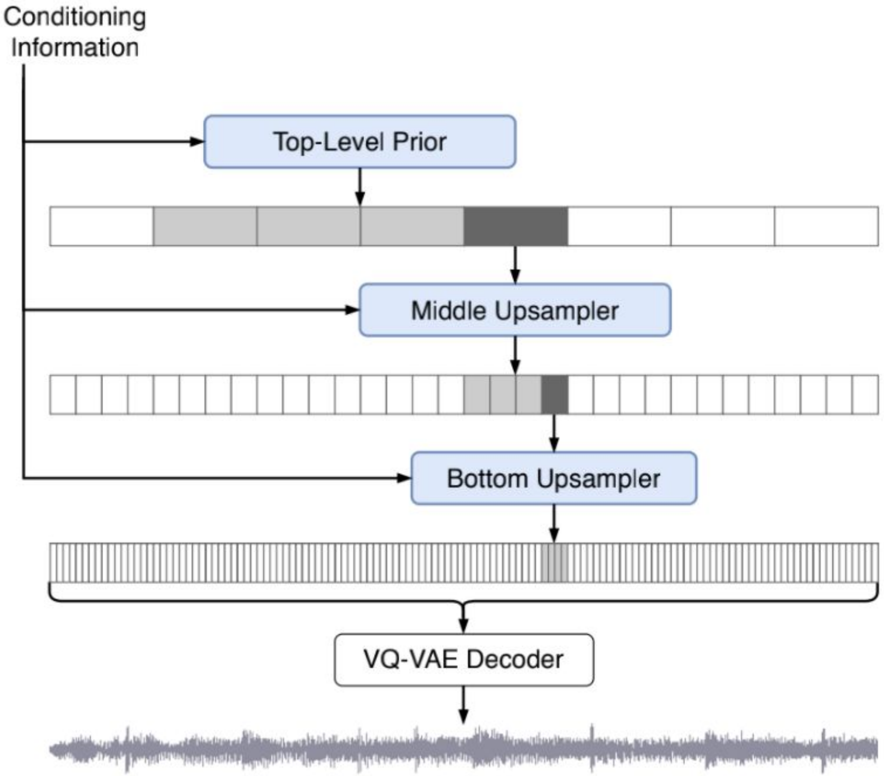
- Use scalable transformer



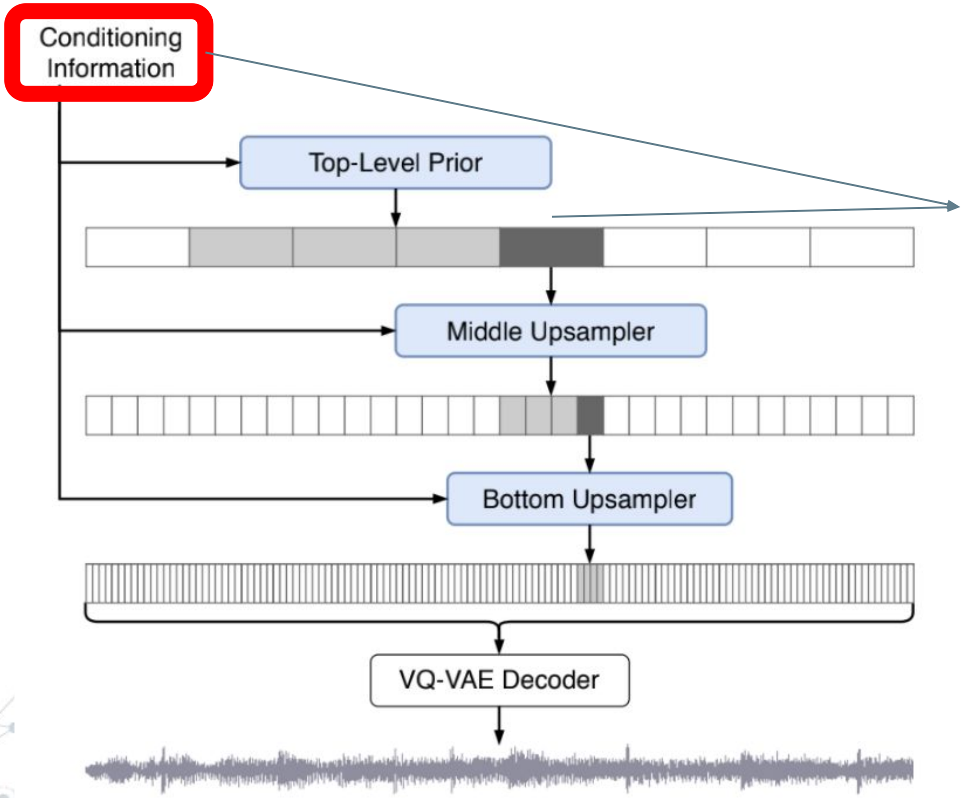
Music Priors and Upsamplers



$$z_t = \text{argmin}_k \parallel \mathbf{h}_t - \mathbf{e}_k \parallel$$



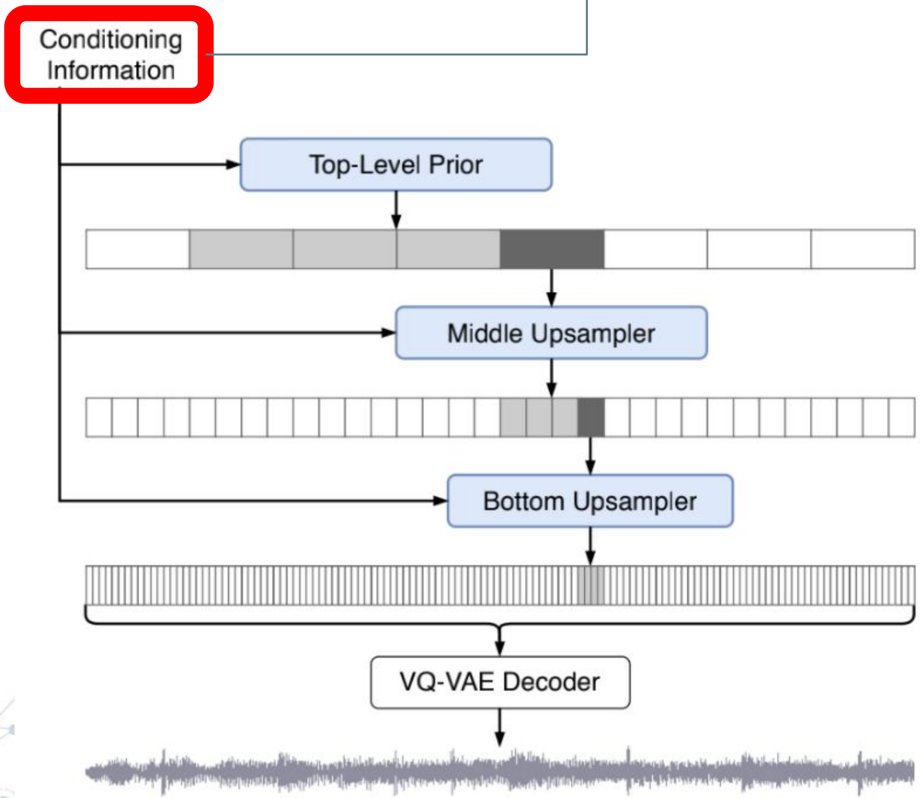
Music Priors and Upsamplers - Artist, Genre Conditioning



Model can be more controllable with info as below

- **Artist, Genre, and timing**
  - Reduce entropy
  - Know the song pattern
- Lyrics Conditioning (next page)
  - Align lyrics and singing
  - Same lyrics but different tone, style
- Encoder-decoder model

Music Priors and Upsamplers - lyrics conditioning



Lyrics-to-singing(LTS) task

- Model learn to align lyrics and singing

LTS difficulties are

- No separation between lead vocals, accompanying vocals, and background music.
- Mismatching portions of lyrics with corresponding music.

Providing lyrics for chunks of a audio

- Training dataset is song-level but with shorte chunks of audio
- Linearly align work well but fail in fast song

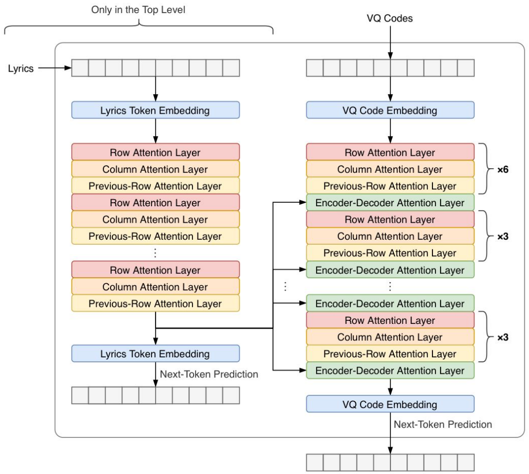
Solution for fast song

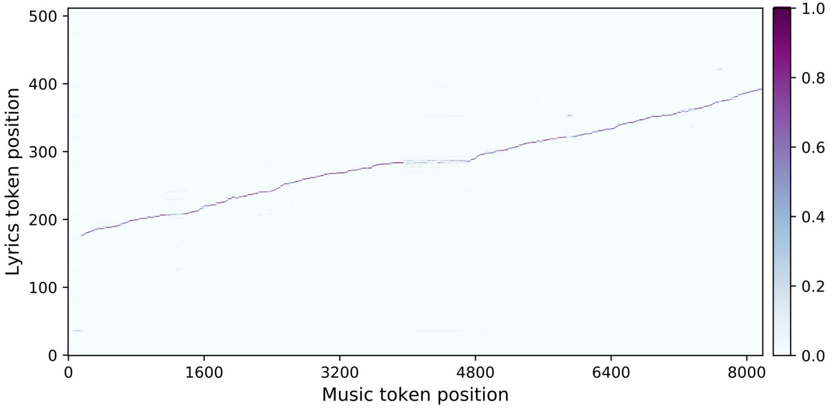
- Spleeter , NUS AutoLyrics align, and bigger window size

Music Priors and Upsamplers - Encoder-decoder model

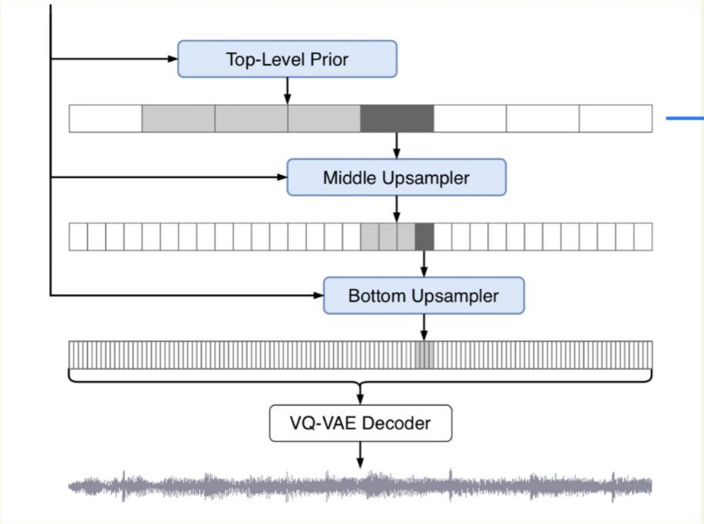
Trained encoder and decoder

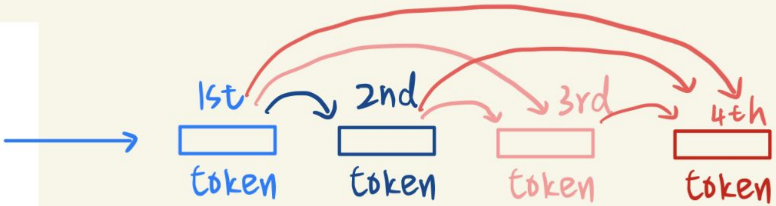
- Encoder producing features from lyrics
- Decoder produces the top level music token





Music Priors and Upsamplers - Sampling





5.

Results

Results - Datasets and training details

Datasets

- dataset with the lyrics and metadata from LyricWiki (LyricWiki).
- Metadata including singer, album, release year, genre
- 1.2 million songs (600k in English)
- Train on 32 bit, 44.1kHz, momo channel audio

Training details

- Top level prior is hardest to train (about 512 V100s for 4 weeks)

Results - Samples

From first model until final model, final model is aka Jukebox.

- 5B, 1B, 44kHz for top level prior, upsamplers, VQ-VAE respectively

New manually measurement for music

- Coherence
- Musicality
- Diversity
    - Re-renditions
    - Completions

Results - Novelty, some application of Jukebox

Novelty styles

- Hard to change the singer singing style

Novel voices

- Most cases, can fuse somebody style to a new voice

Novel lyrics

- Jukebox can sing a non lyrics-like. E.g., poems, novel verses.

Novel riffs

- Finish a riff or add a classical music element to punk song.

6.

Conclusion

Conclusion

Model can

- imitating many different styles and artists
- Style transfer a music on specific artists and genres
- Fuse the lyrics for the sample
- Generate multiple minutes long than previous work with 20-30 seconds

Reference

Dhariwal, Prafulla, et al. "Jukebox: A generative model for music." arXiv preprint arXiv:2005.00341 (2020)

Hung-yi Lee M/L slides

Jukebox: A Generative Model for Music (Paper Explained)

https://blog.csdn.net/zjuPeco/article/details/116159855