Music Generation - Deep Learning 2k25

Hassan Iftikhar
Ali Arshad
Bogdan Monogov
Alen Aliev

Motivation & Background

Why Generate Music with Deep Learning?

- Music is structured, hierarchical, and temporal.
- Manual composition is creative but time-consuming.
- Deep generative models can learn and replicate compositional structure
- Applications:
    - Assisting composers
    - Generating background music
    - Music therapy and personalization

Overview of Models Explored

Model Type	Approach	Strengths
Transformers	Sequence modeling	Long-term memory, flexible tokenization
MuseGAN	GAN for multi-track music	Polyphonic and harmonic generation
Diffusion	Score-based denoising models	High-quality diverse outputs
VAE	Latent space interpolation	Style mixing, variation

Datasets Used

Bach Cello Suites:

Only top note of each chord is used

Converted to note-duration text tokens

Bach Chorales:

Bach Chorales 4-part harmonized chorales (SATB)

Polyphonic and multi-track

229 songs, 2 bars per song, 16 timesteps per bar, 4 voices

Transformers for Monophonic Music

Decoder Transformer trained to predict the next note & duration.

Monophonic setup using top voice of the cello.

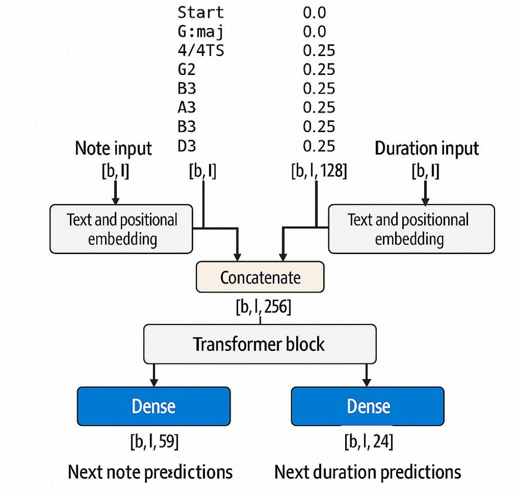
Input : Tokenized sequence of (note, duration) pairs

Output: Predicted next (note, duration)

Transformers for Monophonic Music

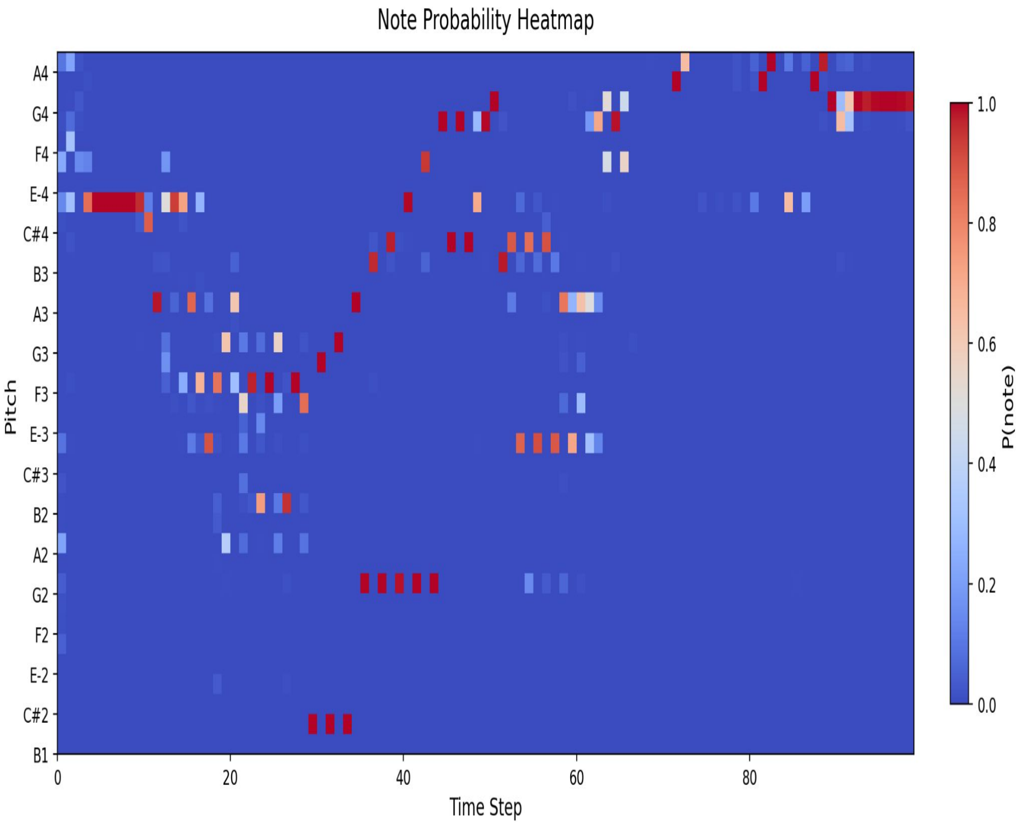
```
Notes string:
 ['E-3', 'B-2', 'C3', 'D3', 'E-2', 'G3', 'F3', 'E-3', 'F3', 'B-3', 'G3', 'G#3', 'G3', 'F3', 'G3', 'E-3', 'F3', 'G3', 'F3',
'E-3', 'F3', 'B-3', 'G3', 'G#3', 'G3', 'F3', 'G3', 'E-3', 'G3', 'G#3', 'B-3', 'G#3', 'B-3', 'E-4', 'C4', 'C#4', 'C4', 'B-
3', 'C4', 'G#3', 'B-3', 'C4', 'B-3', 'G#3', 'B-3', 'E-4', 'C4', 'C#4', 'C4', 'B-3', 'C4'] …

Durations string:
 ['0.5', '0.5', '0.5', '0.5', '2.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5',
'0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.
5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5', '0.5'] …
```

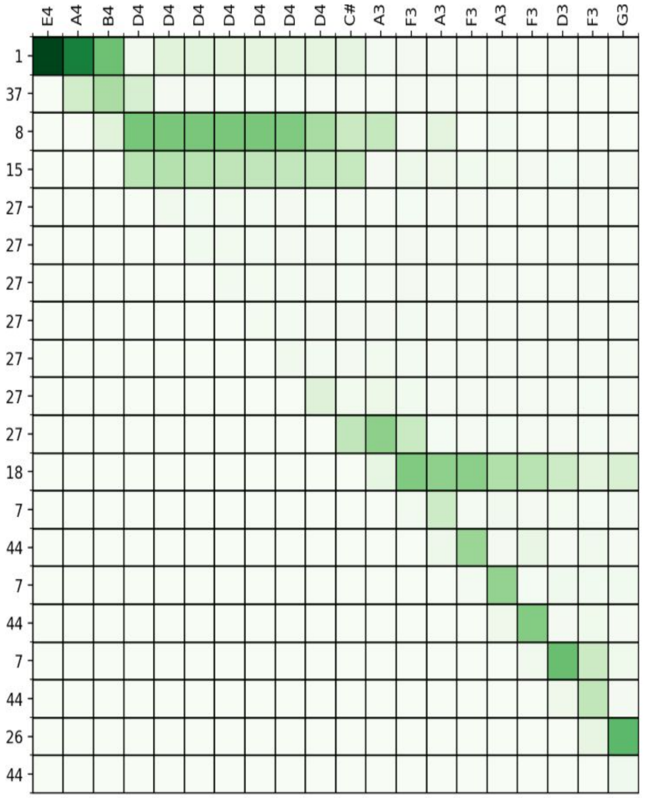
Step	Input (note, dur)		Target (note, dur)	
0	10 (B-3)	2 (0.5)	7 (A3)	2 (0.5)
1	7 (A3)	2 (0.5)	51 (G3)	2 (0.5)
2	51 (G3)	2 (0.5)	7 (A3)	2 (0.5)
3	7 (A3)	2 (0.5)	44 (F3)	2 (0.5)
4	44 (F3)	2 (0.5)	36 (E3)	2 (0.5)



Transformers Results

Music files here: Music_Generation-/transformer/output at main · Micro046/Music_Generation-





MuseGAN for Polyphonic Chorales

Trained on chorales dataset & split into two-bar phrases.

Each sample: Shape (2, 16, 84, 4) → bars × timesteps × pitches × tracks.

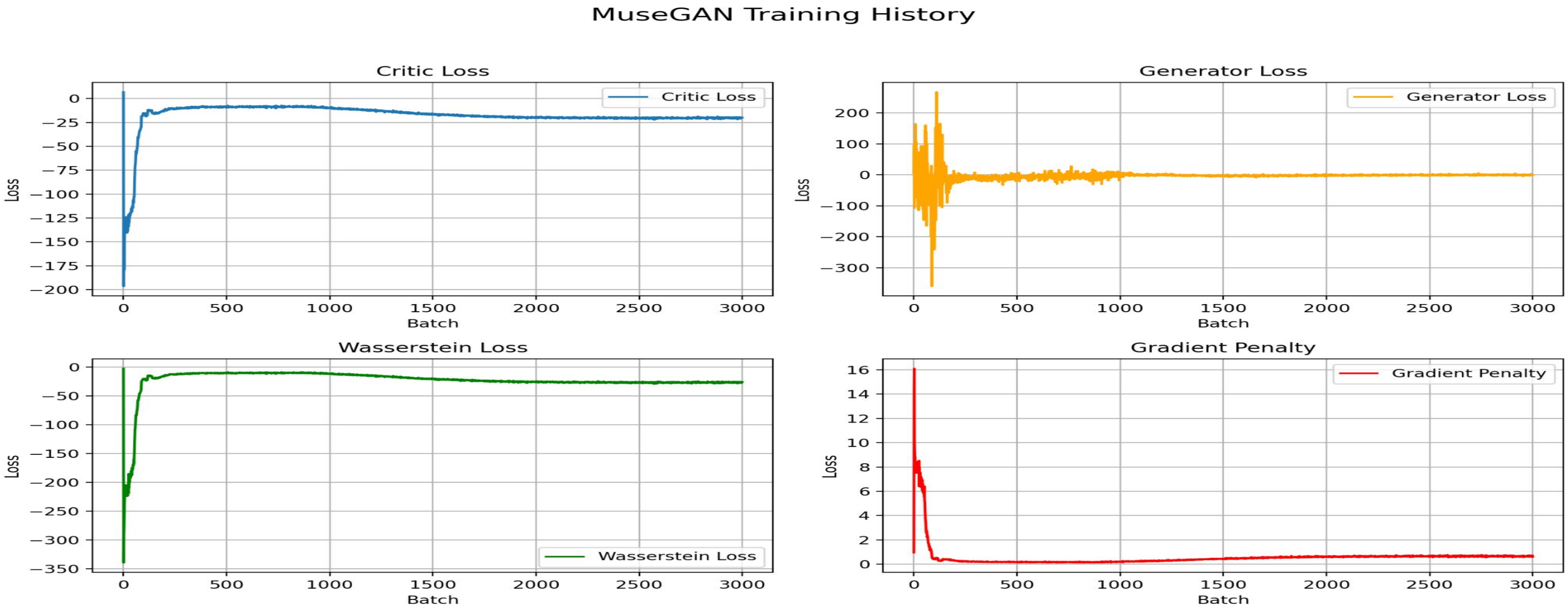
Generator: Generator: Outputs 4-track music

Critic: Judges harmonic and temporal realism

MuseGAN

Music files here: [Music_Generation-/MuseGan/output at main · Micro046/Music_Generation-]

Observation: losses are decreasing and can be seen stabilized.



Diffusion model for music generation

The model was trained on bach-cello dataset (67 MIDI files)

The data underwent the padding, cutting and augmentation

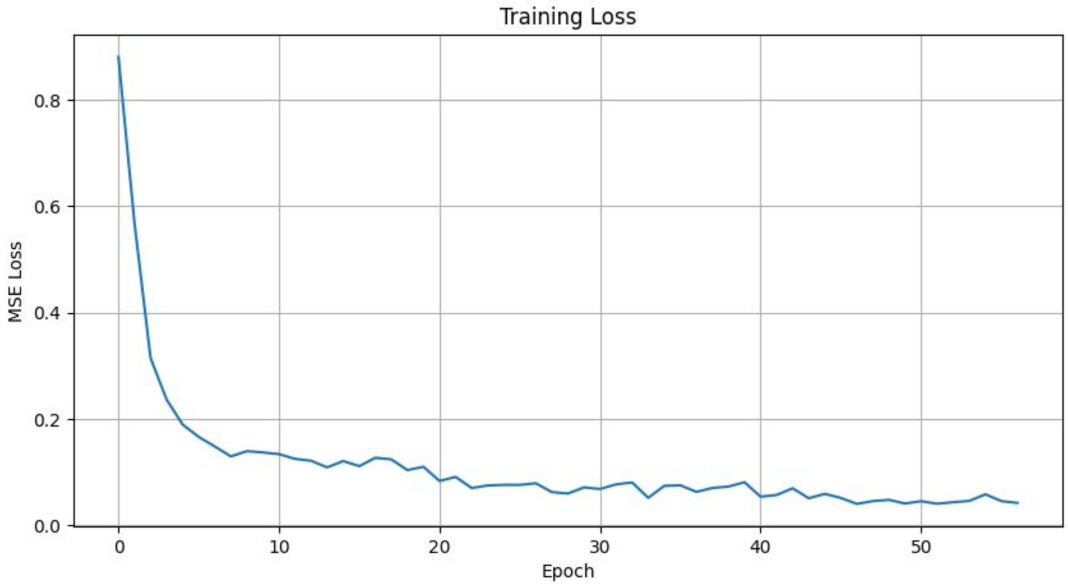
The chosen model architecture is UNet-2D

Hyperparameter name	Value
# of epochs	100
Batch size	8
Learning rate	8e-5
Patience interval	10
Pitch range	128
Number of diffusion steps	1000
Velocity bins	32
Block channels in UNet	(32, 64, 96)

Diffusion model for music generation

Fast convergence (pseudo-convergence)

Poor quality of the generated MIDI samples



Variational Autoencoder

– Input: piano roll sequences (binary grid).

– Encoder compresses to latent vector z.

– Decoder reconstructs sequence from z.

– Trained using reconstruction and KL loss.

Variational Autoencoder: Encoder

– 1D CNN extracts time-local features.

– Bidirectional GRU captures temporal context.

– Fully connected layers output mean and std.

– Latent vector generated via reparametrization.

Variational Autoencoder: Decoder

– MLP maps latent z to notes.

– GRU was too strong, lead to model collapse

– Linear → ReLU → Linear → Sigmoid.

– No recurrence, encourages latent use.

Variational Autoencoder: Loss

– Binary cross-entropy for reconstruction.
– KL divergence regularizes latent space.
– KL warm-up prevents posterior collapse.
– Free bits ensure each dimension contributes.

$$\mathrm{KL} = \sum_{i=1}^{d} \max\left(\tau,\ -\frac{1}{2}\left(1 + \log\sigma_i^2 - \mu_i^2 - \sigma_i^2\right)\right)$$

$$\mathcal{L}_{\text{recon}} = -\sum_{t=1}^{T}\sum_{d=1}^{D}\left[x_{t,d}\log\hat{x}_{t,d} + (1 - x_{t,d})\log(1 - \hat{x}_{t,d})\right]$$

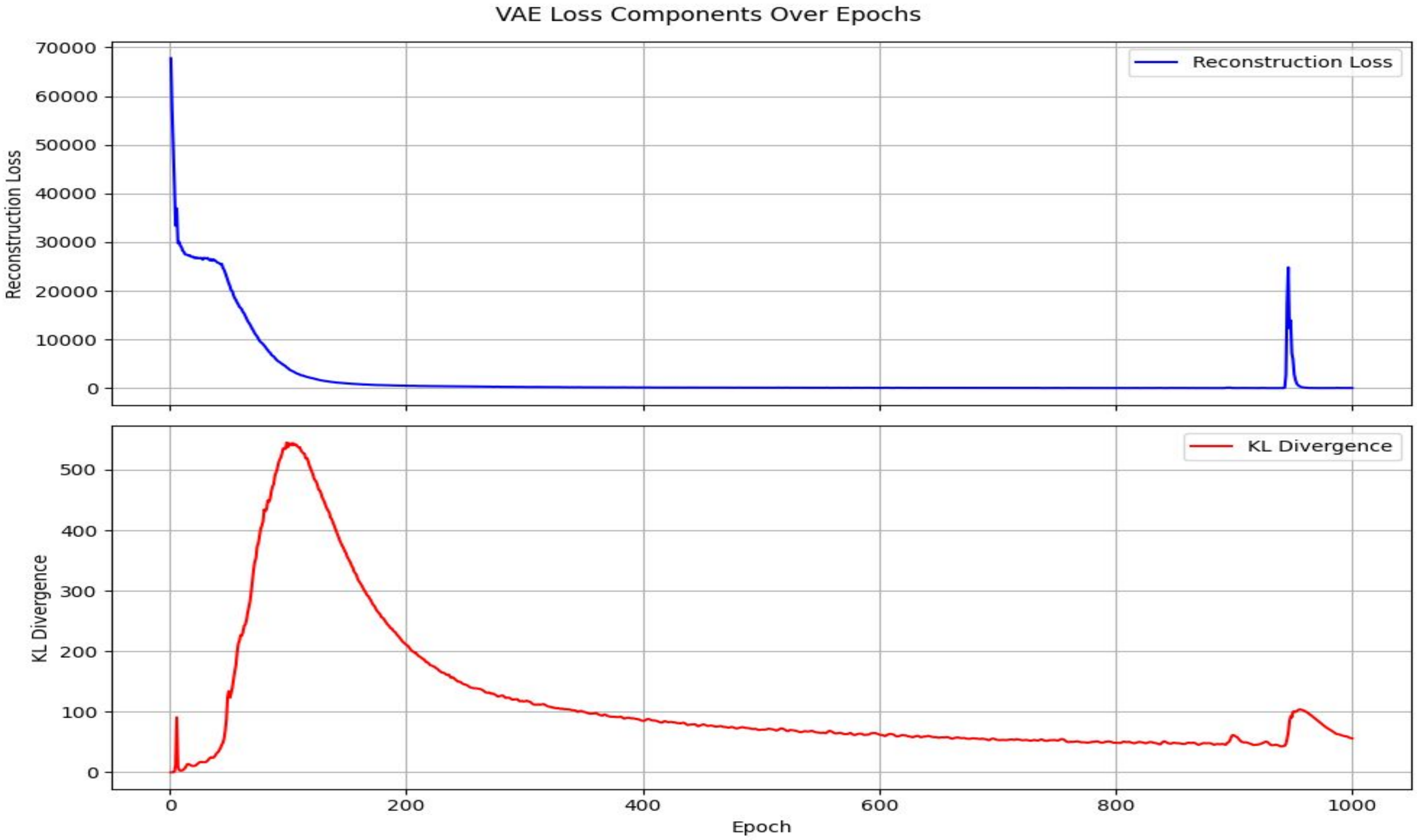
Variational Autoencoder: Loss

– Binary cross-entropy for reconstruction.
– KL divergence regularizes latent space.
– KL warm-up prevents posterior collapse.
– Free bits ensure each dimension contributes.

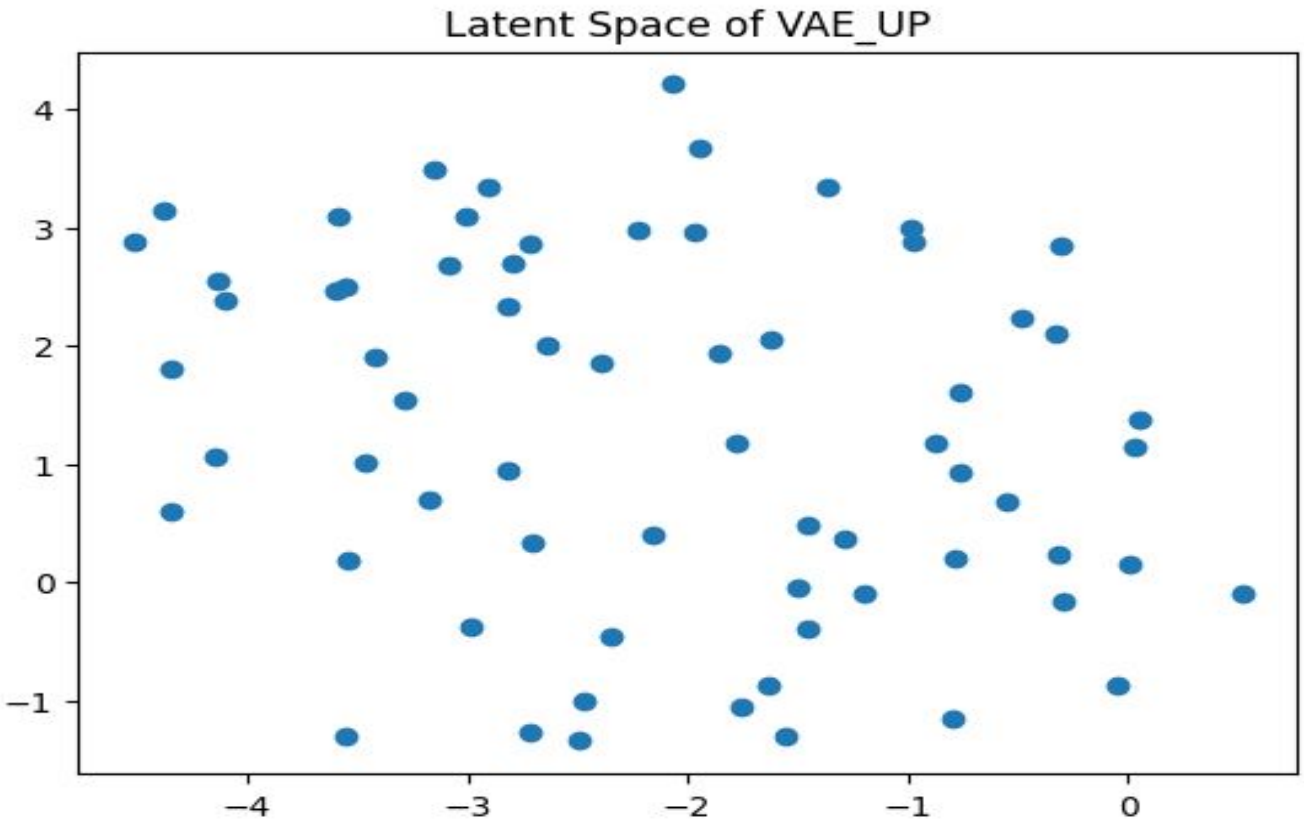
$$\mathrm{KL} = \sum_{i=1}^{d} \max\left(\tau,\ -\frac{1}{2}\left(1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2\right)\right)$$

$$\mathcal{L}_{\text{recon}} = -\sum_{t=1}^{T}\sum_{d=1}^{D}\left[x_{t,d}\log \hat{x}_{t,d} + (1 - x_{t,d})\log(1 - \hat{x}_{t,d})\right]$$

Variational Autoencoder: Training



Variational Autoencoder: Training



Variational Autoencoder: Observations

– VAE can learn meaningful music representations.

– Architecture balance is critical.

– Future: conditional generation and style control.

Conclusion

- Transformers, which excel at capturing long-term dependencies and musical structure
- MuseGAN, which enables multi-track, polyphonic generation using GANs
- Diffusion model shows the worst results from the implemented models (due to the poor translation of image-specific architecture to audio generation, not sufficient number of training samples,inappropriate loss function)

GitHub Repository

[Micro046/Music_Generation-: Music generation project for deep learning]