# Fine-Tuning MusicGen for Composer Style Transfer: Leveraging Style Embeddings for Music Generation

Yanny Gao [1] Sara Kothari [1]

[1]Department of Computer Science, Stanford

## Introduction

Generating **music that mimics a composer's style** is challenging.
While models like MusicGen [2] produce coherent music, they often **miss subtle stylistic traits**.
We aim to achieve **composer style transfer**: given a one-minute audio recording (e.g., Bach) and a text prompt like "Convert this music to Chopin's style," our fine-tuned model generates a new audio piece that reflects Chopin's characteristics.

We achieve three key advancements:

- **Style Encoder:** We trained a style encoder using contrastive loss to produce embeddings that cluster audio samples from the same composer.
- **Fine-Tuning:** We fine-tuned MusicGen using parameter-efficient fine-tuning to integrate style embeddings and align generated audio with a target composer's style by minimizing the mean squared error (MSE) between the generated audio's embedding and the target composer's mean embedding.
- **Composer Classifier:** We developed a classifier using transfer learning with the MusicNet[5] dataset that predicts the composer of a one-minute audio sample, which we used to evaluate the generated music.

We also employed a **cycle consistency evaluation**, reconstructing music through **Composer A** $\rightarrow$ **Composer B** $\rightarrow$ **Composer A** and measuring similarity using **Frechet Audio Distance (FAD)**[4].

Our novel training objective demonstrates promising results for fine-tuning MusicGen, improving its ability to generate music that effectively aligns with target composer style while maintaining the same structure.

## Dataset and Features

We use the MusicNet dataset [5], which contains 330 classical recordings with over one million annotated labels.

1. **Composer Classifier:** We extracted one-minute audio snippets starting after the first minute, processed them with YAMNet [1], and created a dataset of 1524 samples (80/20 train/val split).
2. **Style Encoder:** We used the first 60 seconds (960,000 features at 16kHz) of recordings with corresponding composer labels.
3. **Fine-tuning:** We created 2880 paired samples by pairing each source composer with a target composer. Each sample includes a one-minute audio snippet and a tokenized text description (title + target composer) processed using MusicGen's Autoprocessor

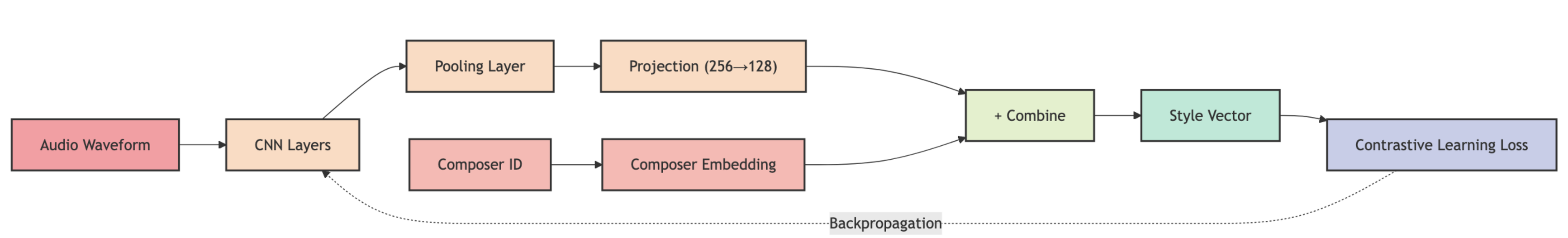



Figure 1. Architecture of style encoder

## Models

- **Style Encoder**: A CNN-based model processing raw waveforms through three 1D convolutional layers with ReLU activations, trained using **InfoNCE contrastive loss**:

$$\mathcal{L} = -\log \frac{\exp(q \cdot k_+)/\tau}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)}$$

- **Composer Classifier**: After training a ResNet-based model and 2 MLP variants, we finally chose the MLP with regularization (batch normalization, L2 regularization, dropout) trained on YAMNet embeddings.
- **MusicGen Fine-tuning**: Used Low-Rank Adaptation (LoRA)[3] with MSE loss between generated and target composer style embeddings:

## Results and Discussions

| Model | Accuracy. (%) | Precision. (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| ResNet-Based | 82.85 | 88.69 | 82.85 | 83.39 |
| MLP w/ Regularization | **90.29** | **90.74** | **90.29** | **90.40** |
| Basic MLP Classifier | 88.35 | 90.33 | 88.35 | 88.68 |

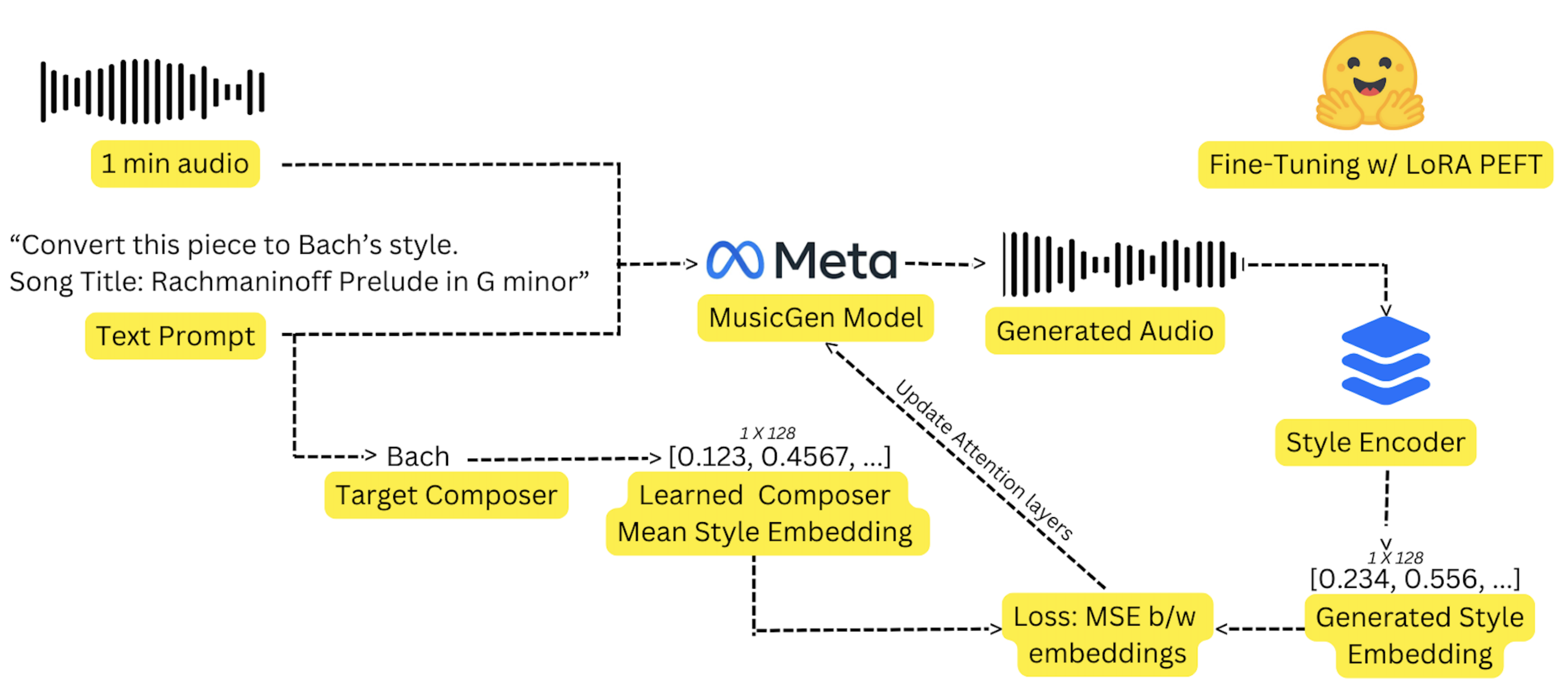
Table 1. Comparison of 3 Composer Classifiers

Our composer classifier experiments showed that the MLP with batch normalization, L2 regularization, and increased dropout performed best (90.29% accuracy), outperforming both the ResNet-Based model and Basic MLP. The confusion matrix reveals strong discriminative power across composer classes.

| Model | Top-1 Acc. (%) | Top-2 Acc. (%) | FAD(↓) |
|---|---|---|---|
| Baseline MusicGen | 9.00 | 13.00 | 482.24 |
| Fine-Tuned MusicGen (LoRA) | **16.00** | **23.00** | **329.13** |

Table 2. Comparison of Fine-Tuned and Baseline MusicGen Models

Our results demonstrate significant improvements in composer style transfer through fine-tuning. The fine-tuned model achieved a **77.78%** improvement in Top-1 accuracy and a **76.92%** improvement in Top-2 accuracy compared to the baseline. The **23.14%** decrease in FAD score indicates better audio quality and style matching.



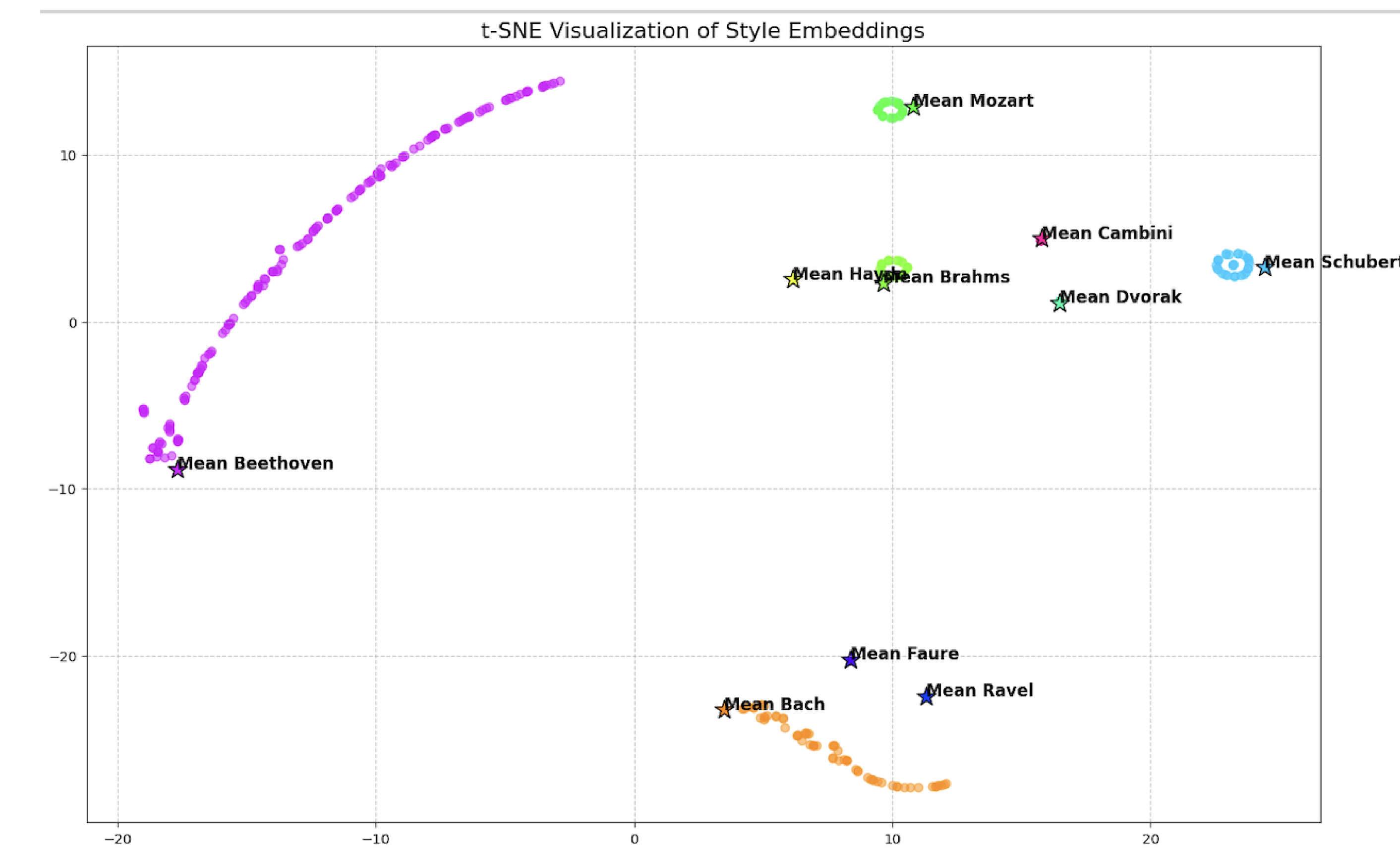




Figure 2. t-SNE visualization of style embeddings generated by Style Encoder of first 60 seconds of each recording in dataset

The t-SNE visualization of our style embeddings demonstrates that the **encoder successfully captured stylistic relationships** between composers - notably placing Faure and Ravel (teacher and student) in close proximity while maintaining clear separation between distinct styles. The MLP trained on these embeddings achieved **95.54%** validation accuracy, confirming their effectiveness at representing stylistic nuance.

Despite computational limitations constraining our fine-tuning, these results validate our approach of using style embeddings and MSE loss to guide music generation toward specific composer styles.

## Future

With additional resources, we would fine-tune the model with the original 30GB dataset, targeting more layers beyond just the attention modules. We would also explore full fine-tuning of the model rather than just using LoRA and experiment with different style encoder architectures to further improve style representation.

## References

[1] models/research/audioset/yamnet at master · tensorflow/models.

[2] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation.

[3] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[4] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms, 2019.

[5] John Thickstun, Zaid Harchaoui, and Sham M. Kakade. Learning features of music from scratch. In *International Conference on Learning Representations (ICLR)*, 2017.