Compositional Textual Inversion for Controllable Music Generation

Tewodros Kederalah Idris Maurine Gatimu
Christine Muthee *
Carnegie Mellon University
10-423/623 Generative AI Course Project

May 1st, 2025

1 Introduction

Controlling musical style in text-to-music generation remains a challenge when relying solely on natural language prompts. In this project, we applied Textual Inversion (TI) to MusicGen, a pre-trained autoregressive model, to introduce new prompt tokens (e.g., "<jazz>") learned from a small number of style-specific audio clips. These tokens enabled more consistent and controllable generation. We further investigated style control by exploring whether multiple learned tokens could be composed within a single prompt to express hybrid musical styles. To evaluate the impact of TI-enhanced prompts, we used CLAP similarity on audio samples of 30 and 150 seconds. The results showed improved performance in 4 out of 8 genres (50%)—particularly Jazz, Pop, Reggae, and Afrobeat—when training on 30-second clips. For 150-second clips, TI led to improvements in 6 out of 8 genres (75%), including Lofi and Ambient, suggesting that longer audio inputs yield more expressive and stylistically rich tokens. In summary, TI offers a lightweight and modular approach to enhance controllability in music generation.

2 Dataset / Task

We used a compact, representative dataset curated to support *textual inversion* for music generation. The dataset included 5 to 6 high-quality audio clips per genre, covering 8 diverse musical styles including Lofi, Hiphop, Pop, Reggae, Country, Jazz, Afrobeat and Ambient music.

Each audio clip was paired with one descriptive text prompt that reflected its stylistic attributes. The prompts were used to run the baseline and the prompt-audio pairs were used during training and evaluation to fine-tune a generative model capable of synthesizing music excerpts from 1–2 sentence style prompts (e.g., "A street heartfelt Hip-hop with calm lo-fi instrumental beats").

*Project code available at https://github.com/teddyk251/Controllable-Music-Generation

The audio clips used in this project were sourced from two publicly available platforms, YouTube and Tubidy, according to the fair use and open license guidelines to ensure ethical data collection. The dataset was curated to be style-representative, enabling us to explore stylistic conditioning using a minimal yet meaningful set of audio examples.

Task Definition

The task was to learn new token embeddings that represent specific musical styles (e.g., "<jazz>"), such that when these tokens are included in text prompts to MusicGen (e.g., "a lo-fi beat with <jazz> saxophone"), the model generates music that reflects the desired style more accurately than with natural language alone.

We formulated this as a style-conditioning problem using textual inversion, where the model was frozen and only the new embedding vector(s) were optimized based on alignment between the generated audio and the target concept.

Evaluation Metrics

We evaluated the learned token embeddings with the following criteria

- **CLAP Similarity:** We used CLAP (Contrastive Language–Audio Pretraining) embeddings to assess semantic similarity between the generated audio and paraphrased natural language descriptions (e.g., "a dancable Afrobeat track"). This allowed us to evaluate how well the learned token captures the intended concept, even if the token itself (e.g., "<Afrobeat>") is out-of-vocabulary for CLAP.

All evaluations were performed relative to a baseline where MusicGen is used with descriptive prompts only, without any fine-tuning or token learning.

3 Related Work

Text-to-Music Generation

Recent models like MusicGen [Copet et al., 2024], MusicLM [Agostinelli et al., 2023], and AudioLDM [Liu et al., 2023] have pushed the boundaries of text-conditioned music generation. These advancements build on earlier breakthroughs such as the Music Transformer [Huang et al., 2018], which was among the first to demonstrate the effectiveness of transformer-based architectures in capturing long-term structure in symbolic music. By introducing relative self-attention, the Music Transformer enabled coherent generation over extended musical sequences—laying the groundwork for applying transformers to musical tasks. Among these, MusicGen stands out as an efficient, autoregressive model that uses discrete audio tokens and a frozen text encoder to generate music from textual prompts. Its transformer-based architecture allows fast inference and flexible prompt control.

Textual Inversion for Generative Models

Textual Inversion (TI) was originally introduced for text to image generation [Gal et al., 2022], where it enables learning new visual concepts from a few examples. TI has since been adapted for text-to-audio models. Plitsis et al. [Plitsis et al., 2023] applied TI to the diffusion-based AudioLDM, learning new musical styles as pseudo-words. While effective, their approach was limited to single-token personalization and a diffusion backbone.

In contrast, Thomé [Thomé et al., 2024] applied TI to MusicGen, showing that it can learn personalized audio concepts efficiently via frozen-weight training. His work demonstrated that TI improves MusicGen's editability and controllability compared to AudioLDM and DreamSound. However, the study did not explore combining multiple tokens or prompt composability.

Controllable and Modular Prompt Design

Most generative music models rely on descriptive natural language to control style and genre, with little support for explicit, composable control tokens. Prompt modularity—a well-known concept in text and image generation—remains underexplored in the music domain. While prior work focuses on adding specific style tokens, we extend the idea to multi-token prompt composition, where users can combine learned style tokens (e.g., "<jazz>" and "<reggae>") to create richer, hybrid musical outputs.

Our Contribution in Context

Our work builds directly on Thomé [Thomé et al., 2024], confirming the viability of TI in autoregressive music models. We contribute by learning multiple TI tokens, exploring prompt composability, and evaluating performance through CLAP [Elizalde et al., 2022] and human judgments.

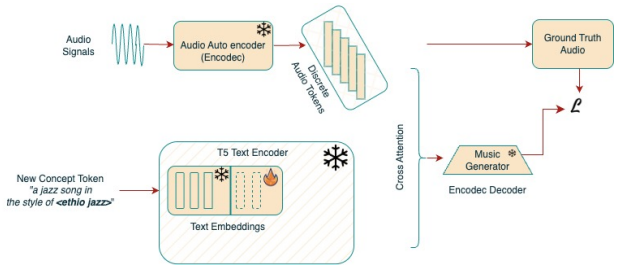

Figure 1: Training Pipeline. *The snowflakes represent the frozen parts of the model and the flame represents the updatable parts of the model*

4 Approach

Our approach builds on MusicGen [Copet et al., 2024], a pre-trained autoregressive model for controllable music generation. MusicGen tokenizes audio into discrete representations using an autoencoder (EnCodec) and generates these tokens conditioned on a text prompt using a Transformer decoder with cross-attention on the encoded prompt. In this project, we aim to improve style control in generation by introducing learned prompt tokens via Textual Inversion. As shown in Figure 1.

Baseline Setup

As a baseline, we used the original MusicGen model (`facebook/musicgen-large`) without any fine-tuning. We generate music using prompts that included descriptive text phrases such as `1970s-style reggae groove with emotional soul influence` or `energetic yet calm lo-fi groove with soulful harmonies, and mellow ambiance`. These prompts were intended to express musical style through natural language alone. This setup allowed us to evaluate how well MusicGen responds to descriptive prompts without introducing any new vocabulary or learned concepts.

Training Prompt Generation

For each music genre listed in Section 2, we crafted multiple prompts that describe distinct stylistic variations within the same genre. The intuition behind this was that no single prompt can fully encapsulate the breadth of a musical style. By prompting the pretrained MusicGen model with multiple nuanced descriptions, we aimed to generate a diverse set of audio samples that better reflect the richness of each genre. These generated clips were then used as part of the baseline analysis. Below are two distinct prompts designed to

capture stylistic variation in the `lo-fi` genre:

- **Prompt A:** Lo-fi hip hop beat with warm vinyl crackle, jazzy piano chords, mellow boom-bap drums, and nostalgic melodic loops — perfect for study and relaxation.
- **Prompt B:** Lo-fi music with icy piano tones, gentle percussion, low analog warmth, and somber emotion — for winter moods and focus flow.

This approach was repeated across other genres such as pop, jazz, and gospel, where multiple prompts help capture a broader range of stylistic attributes relevant to each category.

For each audio sample, we generated a detailed descriptive prompt that was used as input to the MusicGen model during inference. These prompts captured key characteristics of the music—such as rhythm, instrumentation, vocal presence, and other relevant stylistic features—while explicitly specifying the genre. Including the genre was particularly important, as it guided the model in adopting to an appropriate musical style during generation. The quality of the generated music was then evaluated using CLAP, which serves as our baseline metric for performance.

Textual Inversion for Style Token Learning

To extend MusicGen with more controllable style prompts, we applied Textual Inversion to learn new token embeddings that represent specific musical styles or artist characteristics (e.g., `"<jazz>"`, `"<reggae>"`). This was done by:

- Adding new placeholder tokens to the tokenizer vocabulary,
- Freezing all model weights, including the text encoder and transformer decoder,
- Optimizing only the new token embeddings using a small set of style-specific audio clips, and
- Using prompts such as `"a jazz song in the style of <jazz>"` during training.

The optimization objective was to adjust the embedding v^* of the new token such that the generated audio matches the style of the reference clips. We minimized the following loss:

$$\mathcal{L} = \sum_{i=1}^{N} \| f_{\text{musicgen}}(T_{\text{base}} + v^*) - z_i \|^2 \qquad (1)$$

Here, T_{base} is the base prompt embedding, v^* is the learnable token embedding, and z_i is the codebook token sequence of the i^{th} target audio clip.

The objective was to make these new tokens learnable musical concepts that, when included in a prompt, steer generation toward the desired style more reliably and consistently than descriptive text alone.

Prompt Composition Experiments

In addition to learning individual style tokens, we explored prompt composability—the ability to combine multiple learned tokens within a single prompt. For instance:

```
"a lo-fi beat with <jazz> and
<reggae>" vibe
```

This setup allowed us to test whether the learned tokens can act as modular building blocks that interact meaningfully with each other and with standard natural language prompts.

5 Experiments

5.1 Baseline

To establish a foundation for comparison, we conducted a baseline experiment using the pretrained `facebook/musicgen-large` model without any fine-tuning or token modifications. The goal was to evaluate how well the base model aligns audio generation with textual prompts, using CLAP similarity as our metric.

Procedure. 100 text prompts were selected from the MusicCaps dataset [Agostinelli et al., 2023], which contains human-written descriptions of music clips. For each caption, we generated a 10-second music sample using MusicGen. The choice of 10 seconds follows the standard evaluation window from prior works and ensures consistency with the reference music clips in the dataset.

Evaluation. We used CLAP (Contrastive Language-Audio Pretraining) [Elizalde et al., 2022], a multimodal model that embeds audio and text in a shared space. We computed the cosine similarity between the CLAP embeddings of each prompt and its corresponding generated audio.

Results. The average CLAP similarity across the 100 generations was **0.4225**. This serves as our prompt-only baseline.

Real-world Reference. For comparison, we computed CLAP similarity on 38 curated real music-text pairs. The average was **0.4404**, indicating that even real audio rarely achieves perfect alignment, and that our baseline score is reasonable.

5.2 Our Training Approach

1. Procedure We trained and evaluated Textual Inversion (TI) for learning style-controlling tokens by injecting learned embeddings directly into MusicGen's audio

Evaluation Setting	Avg. CLAP
Prompt-only (MusicGen Large)	0.4225
Real Music-Text Pairs	0.4404

Table 1: Baseline CLAP similarity scores.

token codebooks. We trained new embeddings (e.g., "<jazz>", "<pop>") using 3–5 audio clips for each genre (with Ambient having only 3 samples). Importantly, we did not update any of MusicGen's model weights. Instead, we optimized one learnable token per codebook, resulting in four trainable vectors per genre. The model was trained to minimize Mean Squared Error (MSE) between predicted and actual codebook embeddings from the EnCodec audio representations.

2. Token Evaluation. To evaluate the effectiveness of the trained tokens, we generated music using the learned token in prompts (e.g., "a smooth <jazz> track") and computed CLAP similarity against paraphrased textual descriptions of the genre. These were compared against a baseline using prompt-only generation.

3. Prompt Composability. We also explored multi-token prompts (e.g., "a lo-fi beat with <jazz> and <reggae> vibe") to evaluate the compositional behavior of learned tokens and assess stylistic blending. Evaluation included both CLAP similarity and human judgments.

Training Regimes. We experimented with two training regimes:

1. **30 sec clips for 800 epochs** — a shorter context emphasizing convergence over many iterations.
2. **150 sec clips for 600 epochs** — a longer, richer context with fewer total updates.

5.3 Results

Our results, summarized in Table 2 and Table 3, indicate that Textual Inversion (TI) yields genre-specific improvements in CLAP similarity compared to baseline prompt-based generation. In the 30-second, 800-epoch setting (Table 2), TI notably improved similarity for genres such as Pop **(0.547 vs. 0.504)**, Reggae **(0.623 vs. 0.595)**, Jazz **(0.698 vs. 0.606)**, and Afrobeat **(0.403 vs. 0.367)**. However, performance dropped slightly for Hip-hop, Lofi, and Ambient.

In the 60-second, 600-epoch setting (Table 3), TI achieved even more consistent gains, outperforming the baseline in six out of eight genres, including Lofi, Pop, Reggae, and Ambient. Lofi **(0.469 vs. 0.352)** benefited the most from longer audio training samples suggesting that longer training embeds significant information into the "<lofi>" token minimizing the noise that maynot be necessary for training.

Table 2: CLAP Similarity (30 secs, 800 epochs)

Genre	Baseline	TI
Lofi	0.452	0.439
Hip-hop	0.583	0.494
Pop	0.504	**0.547**
Reggae	0.595	**0.623**
Country	0.354	0.347
Jazz	0.606	**0.698**
Afrobeat	0.367	**0.403**
Ambient	0.481	0.407

Table 3: CLAP Similarity (60 secs, 600 epochs)

Genre	Baseline	TI
Lofi	0.352	**0.469**
Hip-hop	0.576	0.482
Pop	0.487	**0.547**
Reggae	0.605	**0.644**
Country	0.288	0.251
Jazz	0.606	**0.618**
Afrobeat	0.358	**0.376**
Ambient	0.431	**0.475**

Overall, our results demonstrate that Textual Inversion (TI) can be effectively applied to auto-regressive music generation, and that learned prompt tokens offer enhanced control over musical style compared to prompt-only generation. The results aligned with our expectations in that **Learned prompt tokens effectively encoded stylistic control** For most of the genres that we trained, the tokens learned via TI, reliably guide the model to generate music in the corresponding style when included in prompts, outperforming general descriptive phrases in terms of consistency and controllability. **Prompt composition exhibited modular control**. Combining multiple learned tokens in a prompt (e.g., "a lo-fi beat with <jazz> and <reggae>") yield blended or distinguishable stylistic features in the generated output. This demonstrates that TI can support compositional control at the prompt level. **We obtain quantitative improvements over the baseline.** Using paraphrased descriptive prompts to generate audio,the audio generated with learned tokens has a higher similarity with the prompt than that generated by the prompt-only baseline. **Efficiency and scalability of TI is an efficient and scalable form of music generation.** Since our approach does not require retraining MusicGen or modifying its weights, we demonstrate that TI is a lightweight and scalable method for personalizing and expanding controllable text-to-music generation. With only 5- 6 training samples, we can capture style nuances and provide an efficient and scalable way to compose music styles

using learnt tokens

6 Code Overview

Our project codebase consists of modular training, inference, and evaluation scripts designed to support Textual Inversion by injecting learned token embeddings directly into MusicGen's codebooks. Below we summarize each major component.

Training (train_tokens.py)

The training script learns one token embedding per codebook (4 total) for each genre:

Initialization: One learnable vector per codebook is initialized with small Gaussian noise. MusicGen is loaded in inference mode with all weights frozen.

Target Extraction: For each audio clip, the frozen EnCodec encoder produces target codebook indices.

Loss Computation: For each codebook, we retrieve pretrained embeddings corresponding to the target indices and minimize MSE with the learned vector (repeated over time).

Optimization: Vectors are optimized using Adam with ReduceLROnPlateau scheduling and early stopping. Training progress is logged via Weights & Biases.

Trained vectors are saved in `trained_tokens/{genre}/`, and the script is located at `scripts/train_tokens.py`.

Inference (inference.py)

This script injects the trained vectors into MusicGen's embedding layers and generates music using prompts like `"a happy <afrobeat> track"`. It loads the base model, appends the trained token to each codebook's vocabulary, and performs generation with the desired prompt.

The script used for generation is `scripts/inference.py`.

Evaluation (evaluate.py)

Evaluation uses CLAP to measure semantic alignment between generated audio and its corresponding prompt. For each genre:

Generation: Audio is generated using both baseline and TI-enhanced prompts.

CLAP Similarity: CLAP embeddings are computed for audio and text, and cosine similarity is used to measure alignment.

Logging: Results are saved in CSV and visualized using a bar chart. The evaluation code is in `scripts/evaluate.py`. Results are saved in the CSV file `similarity_results.csv` and the image file `clap_similarity_barchart.png`, both located in the `evaluation_outputs/` folder.

7 Timeline

Throughout the project, each member actively contributed to the development process. Task distribution was based on individual strengths and interests to maximize efficiency and learning. In addition to asynchronous work, we held biweekly meetings to provide progress updates and collaborate on critical milestones. Table 4 summarizes the key activities, average time spent, and number of contributors involved in each task.

Activity	People	Avg. Time (hrs)
Reading papers	3	21
Understanding the baseline code	3	12
Collecting audio dataset	2	12
Generating baseline prompts	1	12
Writing the proposal document	3	24
Running the baseline code	3	22
Writing and understanding new code	3	48
Writing the midterm report	3	24
Experiments and results	3	42
Creating the poster	2	12
Work on human evaluations	1	12
Final report	3	15
Total Time		**256**

Table 4: Team Activity Breakdown with Average Time and Participation

8 Research Log

8.1 Initial Direction and Proposal

The objective of this project was to develop a method for improving stylistic control in text-to-music generation by applying Textual Inversion (TI). Standard natural language prompts often lack the precision needed to express specific musical styles in a consistent way. To address this limitation, we explored whether introducing new learned tokens—each representing a musical style—could help guide generation more effectively. Our main hypothesis was that these tokens, once trained, would allow MusicGen to produce musical outputs that are more consistent, controllable, and composable across different prompts.

8.2 Running the base code

Our first task was to re-run the baseline using MusicGen to compute CLAP similarity scores. This step was successful and yielded a strong reference score for comparison with subsequent TI-enhanced generations.

8.3 Data Collection

The process of collecting data, however, proved to be a significant challenge. Our objective was to curate a set of audio clips that were both genre-representative and diverse in musical expression. We ultimately selected eight distinct genres with well-defined stylistic features. Identifying subgenre-representative samples, particularly for broad genres like country music, was

especially difficult due to the diversity of musical styles within them. We addressed this by carefully selecting 5 to 6 audio samples per genre that captured a variety of substyles to the best extent possible, ensuring each style had meaningful representation for training.

8.4 Code Implementation

With the dataset finalized, we began adapting the codebase to support Textual Inversion. Our initial plan was to insert a new learnable token into the prompt vocabulary of the T5 text encoder used in MusicGen. However, despite significant effort, this approach encountered persistent integration issues: during inference, the new token (e.g., "<jazz>") was not effectively represented in the encoder outputs, making optimization ineffective. After multiple failed attempts to align the token inside the frozen T5 model, we abandoned this approach.

Instead, we shifted to a more robust method by injecting learned embeddings directly into the audio token codebooks of MusicGen's transformer decoder. This approach did not require modifying the tokenizer or T5 encoder. We created four learnable vectors per genre—one for each of MusicGen's internal codebooks and optimized these vectors to match the true EnCodec representations for genre-specific clips.

This alternative path required refactoring the training loop to allow for embedding updates without affecting the rest of the model. The approach yielded strong results and aligned better with MusicGen's architecture, which already supports modular injection of audio embeddings at generation time.

8.5 Evaluation

Our evaluation strategy involved two main approaches. First, we used CLAP similarity scores to quantitatively assess the alignment between the generated audio and the corresponding text prompts, and to compare the quality of our outputs with those produced by the baseline models. Second, we planned to conduct a human evaluation study to collect subjective feedback on the stylistic fidelity and coherence of the hybrid audio samples generated using learned tokens. However, due to time constraints and implementation challenges in the model, we were unable to complete this component. Despite this, we believe that human feedback would have provided valuable insights into the perceived effectiveness and musical coherence of our method.

9 Conclusion

In this work, we explored the use of Textual Inversion as a lightweight and effective approach for controlling style in text-to-music generation. Our findings demonstrate that introducing learned tokens enables reliable steering of musical outputs, even with as few as 5–6 training examples per style. This highlights the potential of TI as an efficient method for style conditioning. Our experiments further show that training on longer audio sequences leads to richer token embeddings, improving text-audio alignment as measured by CLAP similarity. Overall, our results suggest that Textual Inversion offers a promising direction for enhancing controllability, consistency, and composability in music generation systems like MusicGen.

9.1 Future Works

Future work will focus on expanding the scope and robustness of our approach. First, we plan to train on a larger number of audio samples per style and explore finer-grained sub-genres within broader categories (e.g., distinguishing between smooth jazz and bebop). This will help us evaluate the granularity of control Textual Inversion can offer. Second, we aim to design genre-specific prompt structures that align more closely with the stylistic characteristics of the target genre, potentially improving prompt-model alignment. This could potentially guide users in generating high quality music that is controllable and coherent. We also acknowledge the value of human evaluation as a reliable benchmark in generative approaches, therefore we plan to incorporate listening surveys to quantitatively and qualitatively evaluate the quality of our generated music. Lastly, we recognize the importance of diversity and inclusiveness in generative systems, and therefore intend to incorporate underrepresented genres, particularly from non-Western musical traditions, to enhance the cultural breadth of stylistic conditioning in music generation.

References

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text, 2023. URL https://arxiv.org/abs/2301.11325. arXiv:2301.11325.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation, 2024. URL https://arxiv.org/abs/2306.05284. arXiv:2306.05284.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap: Learning audio concepts from natural language supervision, 2022. URL https://arxiv.org/abs/2206.04769.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. URL https://arxiv.org/abs/2208.01618. arXiv:2208.01618.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer, 2018. URL https://arxiv.org/abs/1809.04281. arXiv:1809.04281.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. Audioldm: Text-to-audio generation with latent diffusion models, 2023. URL https://arxiv.org/abs/2301.12503. arXiv:2301.12503.

Manos Plitsis, Theodoros Kouzelis, Georgios Paraskevopoulos, Vassilis Katsouros, and Yannis Panagakis. Investigating personalization methods in text-to-music generation, 2023. URL https://arxiv.org/abs/2309.11140. arXiv:2309.11140.

Carl Thomé, Bob L. Sturm, Jonas Pertoft, and Niklas Jonason. Applying textual inversion to control and personalize text-to-music models. In *Proceedings of the 15th International Workshop on Machine Learning and Music (MML 2024)*, 2024.