CONTENT-BASED CONTROLS FOR MUSIC LARGE LANGUAGE MODELING

Liwei Lin[1,2] **Gus Xia**[1,2] **Junyan Jiang**[1,2] **Yixiao Zhang**[3]
[1] Music X Lab, New York University Shanghai
[2] Mohamed bin Zayed University of Artificial Intelligence
[3] C4DM, Queen Mary University of London

ABSTRACT

Recent years have witnessed a rapid growth of large-scale language models in the domain of music audio. Such models enable end-to-end generation of higher-quality music, and some allow conditioned generation using text descriptions. However, the control power of text controls on music is intrinsically limited, as they can only describe music *indirectly* through meta-data (such as singers and instruments) or high-level representations (such as genre and emotion). *We aim to further equip the models with direct and* ***content-based*** *controls on innate music languages* such as pitch, chords and drum track. To this end, we contribute *Coco-Mulla*, a **co**ntent-based **co**ntrol method for **mu**sic **l**arge **la**nguage modeling. It uses a parameter-efficient fine-tuning (PEFT) method tailored for Transformer-based audio models. Experiments show that our approach achieves high-quality music generation with **low-resource** semi-supervised learning. We fine-tune the model with less than 4% of the orignal parameters on a small dataset with fewer than 300 songs. Moreover, our approach enables effective content-based controls. We illustrate its controllability via chord and rhythm conditions, two of the most salient features of pop music. Furthermore, we show that by combining content-based controls and text descriptions, our system achieves flexible music variation generation and arrangement. Our source codes and demos are available online [1] [2].

1. INTRODUCTION

Controllable music generation encompasses the creation of music under various controls, such as musical or textual descriptions [1–4]. It enables amateur users to create customized music and helps professional musicians explore new ideas for composition and arrangement. Remarkable advancements have been made in this field in recent years, particularly in text-to-music generation [3–6]. These cross-modality models are trained on extensive sets of parallel text-audio data pairs, utilizing pre-trained large language models [7] or multi-modal embeddings [8, 9] to establish a mapping between natural language and music. They enable high-level controls such as mood and tempo by incorporating them into a text prompt.

However, not all music information can be expressed via text. Existing models cannot yet apply effective controls on intricate musical languages (e.g., chord progressions) or directly refer to musical contents from other audio recordings. The ability to accommodate such *content-based* controls is crucial for tasks such as music editing, music variation generation, and arrangement. For example, in AI-assistant composition systems, the generative models are required to compose based on a rough idea like motifs or counter-melodies; in music re-instrumentation, we tend to keep the underlying harmony unchanged and reinterpret the song with new instruments.

Historically speaking, the shift from text-based (and metadata-based) models to content-based models has once happened in the realm of music information retrieval (MIR), which dramatically improved the model performances and also expanded the boundary of music understanding. In a similar fashion, we aim to push the boundary of music audio generation domain by further equipping off-shelf generative models with content-based controls.

A simple approach to incorporating content-based control is to train a separate generative model conditioned on the provided control content [4, 10]. For example, both the MusicGen and MusicLM systems offer two versions of model: a vanilla text-to-music version, and a melody-conditioned version for accompaniment generation. The main issue with this simple conditioning approach lies in the high training cost, which is at the same scale as the base model in terms of both computational resources and training data. Moreover, each conditioned model can only deal with *one type* of content-based control input. This rigid setting is not practical in real music production and arrangement scenarios where multiple control contents are often required for satisfactory results.

To solve the aforementioned problems, we contribute a unified approach to incorporating different content-based controls with music-audio generative models. Particularly,

[1] https://github.com/Kikyo-16/coco-mulla-repo.
[2] https://kikyo-16.github.io/coco-mulla/.

we see the immense potential of large-scale pre-trained models in their semantic-level understanding of music and therefore propose a novel parameter-efficient fine-tuning condition adaptor based on llama adaptor [11].

The current design of the adaptor integrates the joint embeddings of symbolic music chords and piano roll and acoustic drum tracks with the pre-trained MusicGen model. In theory, the adaptor can perform on joint embeddings of any combination of content-based controls and can be integrated with any Transformer-based generative model. In short, the main contribution of this work is as follows:

- **A unified approach on content-based controls**: Our model enables chord and drum pattern controls via acoustic hints, achieving an arbitrary combination of textual, harmonic, and rhythmic description for the controlled generation process.
- **Low-resource fine-tuning on pseudo-labeled datasets**: We provide a method to fine-tune a large auto-regressive audio generative model with a small-size, pseudo-labeled dataset in which all the pseudo labels are extracted using existing MIR tools. We fine-tune MusicGen [4], an excellent text-to-music model, on 4% trainable parameters of the original model with a training set of fewer than 300 songs without text or other annotations.
- **Flexible variation generation and arrangement**: Our model achieves flexible variation generation and arrangement of the given polyphonic piano roll by combining text prompts and content-based controls. This enables numerous downstream music-editing applications.

2. RELATED WORK

We review two realms of related works: 1) large-language models (LLMs) for music audio and 2) existing methods of parameter-efficient fine-tuning.

2.1 Music Audio Generation

Music audio generation necessitates extensive contextual modeling to account for the intricate structure of musical language. Recent large-scale music audio generative models, encompassing auto-regressive and diffusion-based approaches, have made remarkable strides in capturing such a long-term structure while introducing cross-modality conditions. For example, Jukebox [1] leverages VQ-VAE [12] and transformer decoders to achieve lyrics- and genre-based generation; Diffusion-based Moûsai [3] adopts the pre-trained frozen T5 encoder [7] to summarize text conditions; auto-regressive MusicGen [4] realizes monophonic melody and text controls by assembling EnCodec [13], T5 encoder, and an acoustic transformer decoder. Specifically, MusicGen is the first text- and melody-conditioned model, limited to a monophonic melody condition, and it does not accommodate drum tracks. Additionally, a contemporaneous work, Music ControNet [14], shares a similar goal with ours but is based on a diffusion model rather than on pretrained large language models (LLMs).

2.2 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) methods adapt pre-trained language models (PLMs) without fine-tuning all model parameters [15, 16], significantly reducing computational and storage expenses. [17] and [18] tailor PLMs for specific tasks by appending task-specific prefixes to input sequences. [19] employs low-rank adaption (LoRA) to fine-tune pre-trained linear transformations in PLMs. [11] and [20] propose LLaMA-Adapter, adjusting attention outputs using prompt adaptors and zero gate scalars. LLaMA-Adapter also introduces a multi-modal conditional variant by incorporating *global* image representations into prompt adaptors. In this study, we present a novel PEFT method, inspired by LLaMA-Adapter, designed to fine-tune large-scale models while accommodating external *sequential* multi-modal conditions.

3. BASE MODEL

In this work, we choose MusicGen [4], an excellent Transformer-based music audio language model, as our base model. MusicGen offers two variants: a text-only model and a melody-based model. The melody-based model conditions its generation on the dominant time-frequency bin of the audio chromagram and text prompts, limiting it to monophonic conditioning and preventing it from incorporating rhythmic drum patterns. In this study, we adopt the text-only MusicGen as our base model, augmenting it with content-based controls via the proposed PEFT method.

The largest text-only MusicGen consists of 3 components: a pre-trained EnCodec, a pre-trained T5 encoder, and an acoustic transformer decoder. The transformer decoder comprises $N = 48$ layers, each including a causal self-attention block and a cross-attention block to handle condition text prompts.

MusicGen tokenizes audio signals using EnCodec [13], a Residual Vector Quantization (RVQ) [21] auto-encoder, compressing signals of sample rate $S_r = 32000$ into discrete codes of a low frame rate $f_s = 50$. The acoustic transformer decoder takes these tokens as its input.

4. METHODOLOGY

Our approach consists of 2 components: 1) a joint embedding encoder to integrate content-based controls, and 2) a condition adaptor to fine-tune MusicGen by incorporating the learned joint embeddings. To maintain the ability of the vanilla MusicGen to associate text with music, we train the adapter for only the self-attention blocks of the acoustic transformer decoder. During training, all the parameters of MusicGen are frozen.

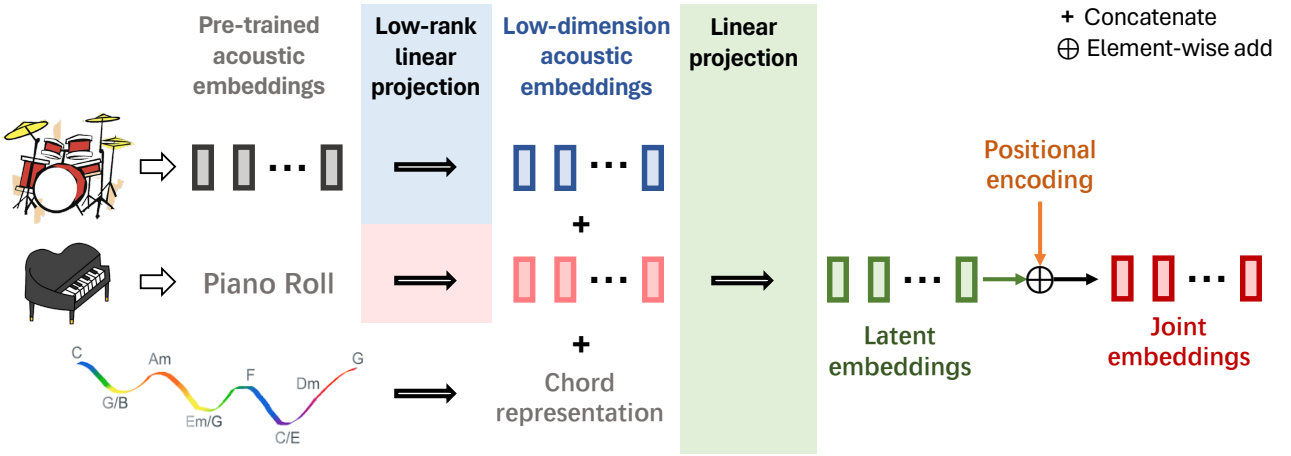


Figure 1. The joint embedding module. We randomly mask acoustic or piano roll embedding with probability r during training.

Chord Name	Pitch	Root	Bass	Pitch Indices
C:maj	C, E, G	C (0)	C (0)	0, 4, 7
C:maj/E	C, E, G	C (0)	E (4)	0, 4, 7
D:min	D, F, A	D (2)	D (2)	0, 3, 7

Table 1. Symbolic chord data structure.

4.1 Joint Symbolic and Acoustic Embedding

To incorporate the desired chord progression and to borrow musical content from another audio recording simultaneously, we design a joint symbolic and acoustic embedding. For precise alignment of generated music with acoustic hints, we use a frame-wise representation for both symbolic and acoustic data.

4.1.1 Symbolic Chord and MIDI Representation

We describe a chord as a combination of a root pitch class, the bass pitch class, and the chroma representation of the chord quality. As shown in Table 1, we represent a chord as $\{\text{root}, \text{bass}, \boldsymbol{m}\}$ $(\text{root}, \text{bass} \in \{0, 1, ..., 11\})$, with $\boldsymbol{m} \in \mathbb{R}^{12}$ being a multi-hot positional vector indicating the active pitches in the octave starting from the root note. Define $\boldsymbol{c}_i \in \mathbb{R}^{12+12+12+1}$ to represent the i^{th}-frame chord:

$$\boldsymbol{c}_i = \begin{cases} [e(\text{root}); e(\text{bass}); \boldsymbol{m}; 0], & \text{if } i^{\text{th}} \text{ frame has a chord} \\ [\mathbf{0}; \mathbf{0}; \mathbf{0}; 1], & \text{otherwise} \end{cases}, \tag{1}$$

where e is a function from an index $j \in \{0, 1, ..., 11\}$ to its one-hot vector $e(j) \in \mathbb{R}^{12}$.

We represent MIDI using the piano roll format. Assume $\boldsymbol{p}_i \in \{0, 1\}^{128}$ is the i^{th} frame in the piano roll indicating the presence of each pitch. We further compress the sparse piano roll into a low-dimension MIDI representation $\boldsymbol{p}'_i$ using a trainable matrix $\boldsymbol{W}_{\text{p}} \in \mathbb{R}^{d_1 \times 128}$:

$$\boldsymbol{p}'_i = \boldsymbol{W}_{\text{p}}^{\mathsf{T}} \boldsymbol{p}_i \in \mathbb{R}^{d_1}. \tag{2}$$

Throughout the training process, we use pseudo chord annotations obtained through a chord recognition model from [22] and MIDI annotations via an automatic music transcription model from [23].

4.1.2 Acoustic Representation

We convert the separated drum stem to discrete codes using EnCodec [13]. Instead of directly modeling these discrete codes, we pass the i^{th}-frame codes through the frozen input embedding layer of MusicGen transformer decoder to obtain a pre-trained acoustic embedding $\boldsymbol{h}_i \in \mathbb{R}^{2048}$. Such a continuous pre-trained representation is more robust since it can address the issue of utilizing discrete codes not present in the training data during the inference stage.

To mitigate overfitting and reduce training complexity, we employ a trainable low-rank matrix $\boldsymbol{W}_{\text{a}} \in \mathbb{R}^{2048 \times d_2}$ to map $\boldsymbol{h}_i$ into a lower-dimensional space:

$$\boldsymbol{h}'_i = \boldsymbol{W}_{\text{a}}^{\mathsf{T}} \boldsymbol{h}_i \in \mathbb{R}^{d_2}. \tag{3}$$

We set $d_2 = d_1 = 12$ in our experiments.

4.1.3 Masking Scheme and Positional Encoding

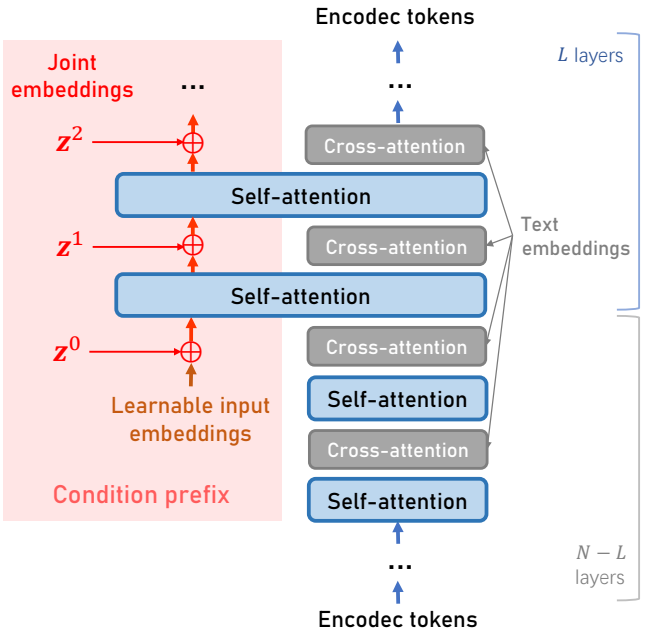
During training, we randomly mask MIDI and acoustic representation with a probability r independently:

$$\boldsymbol{z}_i^{\text{p}} = \begin{cases} \boldsymbol{p}'_i, & \text{if not masked} \\ \boldsymbol{s}_i^{\text{p}}, & \text{otherwise} \end{cases}, \tag{4}$$

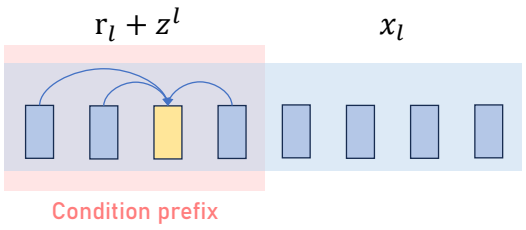
$$\boldsymbol{z}_i^{\text{a}} = \begin{cases} \boldsymbol{h}'_i, & \text{if not masked} \\ \boldsymbol{s}_i^{\text{a}}, & \text{otherwise} \end{cases}. \tag{5}$$

Here, $\boldsymbol{s}_i^{\text{p}}$ and $\boldsymbol{s}_i^{\text{a}}$ are learnable masked embeddings. The masking strategy trains the model to follow an arbitrary combination of conditional tracks during inference. We set $r = 0.4$ in our experiments.

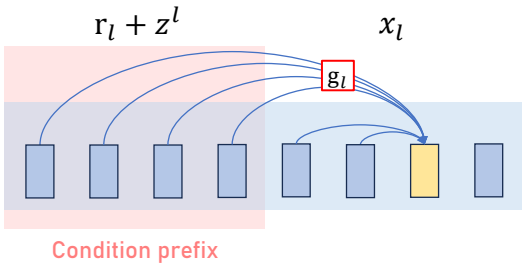
We introduce a learnable matrix $\boldsymbol{W}_{\text{e}} \in \mathbb{R}^{(d_1+d_2+37) \times d}$ to incorporate the above embeddings and a learnable positional embedding $\boldsymbol{z}_i^{\text{pos}} \in \mathbb{R}^{d_1+d_2+37}$ to facilitate sequen-



(a) To incorporate content-based control from the joint embeddings, the condition prefix joins into the last L frozen self-attention layers of the MusicGen decoder.



(b) Inside the condition prefix, each position can attend to all other positions within the condition prefix.



(c) Outside the condition prefix are the hidden embeddings corresponding to Encodec tokens. They attend to all preceding positions (following the causal mask) and the entire condition prefix.

Figure 2. Condition adaptor. The condition prefix is injected to the self-attention mechanism of the MusicGen transformer decoder. All transformation matrices in MusicGen are frozen. Only the input embeddings, joint embedding encoders, and the gate factors are trainable.

tial modeling. The final joint symbolic and acoustic embedding is as follows:

$$\boldsymbol{z}_i = \boldsymbol{W}_{\mathrm{e}}^{\mathsf{T}}([\boldsymbol{c}_i; \boldsymbol{z}_i^{\mathrm{p}}; \boldsymbol{z}_i^{\mathrm{a}}] + \boldsymbol{z}_i^{\mathrm{pos}}) \in \mathbb{R}^d. \tag{6}$$

Finally, let T be the total number of frames. The complete sequential joint embedding is then:

$$\boldsymbol{z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_T\} \in \mathbb{R}^{T \times d}. \tag{7}$$

4.2 Condition Adaptor

To plug the joint embeddings into MuiscGen, we present a novel condition adaptor that can take time-varying sequential conditions. In the vanilla Transformer, each self-attention layer operates on a sequence of T hidden embeddings corresponding to T frames of Encodec tokens. In the proposed condition adaptor, as shown in Fig 2, for the last L layers of the MusicGen decoder, we expand the sequence of hidden embeddings to $2T$, where T new positions take on the task of incorporating and processing condition-related information. We call the newly introduced positions the *condition prefix*.

Specifically, we insert a sequence of learnable input embeddings into the $(N - L + 1)^{\text{th}}$ MusicGen transformer decoder layer, initiating the condition prefix. Inside the condition prefix, we pass the hidden states only through self-attention layers, skipping cross-attention layers.

Let $\boldsymbol{H}_l^{\mathrm{p}} \in \mathbb{R}^{T \times d}$ $(N - L + 1 \leq l \leq N)$ represent the output of the l^{th}-layer attention layer for the condition prefix. $\boldsymbol{H}_0^{\mathrm{p}}$ is a sequence of learnable input embeddings. We compute the condition prefix as follows:

$$\boldsymbol{Q}_l^{\mathrm{p}}, \boldsymbol{K}_l^{\mathrm{p}}, \boldsymbol{V}_l^{\mathrm{p}} = \text{QKV-projector}(\boldsymbol{H}_l^{\mathrm{p}} + \boldsymbol{Z}_l), \tag{8}$$

$$\boldsymbol{H}_{l+1}^{\mathrm{p}} = \text{Self-Attention}(\boldsymbol{Q}_l^{\mathrm{p}}, \boldsymbol{K}_l^{\mathrm{p}}, \boldsymbol{V}_l^{\mathrm{p}}), \tag{9}$$

where the sequential joint embeddings $\boldsymbol{Z}_l$ is defined in Eq (7). Note that we learn distinct joint embeddings for each decoder layer. Since the adaptor aims at capturing the long-term contextual information of the sequential joint embeddings, it *does not* employ a causal attention mask for the condition prefix. Additionally, the condition prefix does not attend to the Encodec token frames, as shown in Fig 2(b).

For the non-prefix part, hidden states are passed through both self-attention and cross-attention layers. Let $\boldsymbol{H}_l \in \mathbb{R}^{T \times d}$ $(1 \leq l \leq N)$ represent the output of the l^{th} attention layer for the encoded tokens. we compute vanilla attention output $\boldsymbol{S}_l$ as follows:

$$\boldsymbol{Q}_l, \boldsymbol{K}_l, \boldsymbol{V}_l = \text{QKV-projector}(\boldsymbol{H}_l), \tag{10}$$

$$\boldsymbol{S}_l = \text{Self-Attention}(\boldsymbol{Q}_l, \boldsymbol{K}_l, \boldsymbol{V}_l). \tag{11}$$

To incorporate condition information, in the last L layers, we compute cross attention $\boldsymbol{S}_l^{'}$ between $\boldsymbol{Q}_l$ and $\{\boldsymbol{K}_l^{\mathrm{p}}, \boldsymbol{V}_l^{\mathrm{p}}\}$. We leverage Self-Attetion layers to compute $\boldsymbol{S}_l^{'}$ rather than Cross-Attention layers since the controls are closer to the audio modality than the textual modality. To

	$\textbf{Chord}_{\text{rec}}\uparrow$	$\textbf{Chord}^{*}_{\text{rec}}\uparrow$	$\textbf{Beat}_{F_1}\uparrow$	$\textbf{CLAP}_{\text{scr}}\uparrow$	$\textbf{FAD}^{*}_{\text{vgg}}\downarrow$	$\textbf{FAD}_{\text{vgg}}\downarrow$
Chord-only	0.412	0.195	-	**0.401**	6.209	6.695
MIDI-only	0.649	0.406	-	0.381	7.105	7.094
Drums-only	0.530	0.267	0.856	0.360	3.845	4.933
Full	**0.791**	**0.524**	**0.864**	0.351	**3.697**	**4.370**
MusicGen	-	-	-	0.441	6.434	6.847
Oracle	0.885	0.695	0.898	-	-	

Table 2. The performance of the model with $L = 48$ on RWC-POP-100 subset. Oracle scores gauge the performance of the chord recognition model [22] and beat tracking model [24] on the ground-truth audio.

L	**Total**	**Trainable**	$\textbf{CLAP}^{*}_{\text{scr}}\uparrow$		$\textbf{Chord}_{\text{rec}}\uparrow$	
			Chord-only	**Full**	**Chord-only**	**Full**
12	3.29B	0.87%	**0.428**	**0.371**	0.239	0.672
24	3.31B	1.66%	0.408	0.358	0.397	0.747
36	3.33B	2.44%	0.396	0.344	0.410	0.772
48	3.36B	3.20%	0.401	0.351	**0.412**	**0.791**

Table 3. The performance of models with different L values under the chord-only condition.

make {query, key, value} more compatible, we use the fusion of $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_l^{\text{p}}$ as the query instead of a single $\boldsymbol{Q}_l$:

$$\boldsymbol{S}_l^{'} = \text{Self-Attention}(\boldsymbol{Q}_l + \boldsymbol{Q}_l^{\text{p}}, \boldsymbol{K}_l^{\text{p}}, \boldsymbol{V}_l^{\text{p}}). \quad (12)$$

Following this, we combine $\boldsymbol{S}_l$ and $\boldsymbol{S}_l^{'}$ using a zero-initialized learnable gating factor g_l. Additionally, to maintain the text controllability of the model, we then compute the cross attention between textual embedding and them:

$$\boldsymbol{H}_{l+1} = \text{Cross-Attention}(\boldsymbol{S}_l + g_l \cdot \boldsymbol{S}_l^{'}, text). \quad (13)$$

All the layers in MusicGen are frozen, including the QKV-projector, Self-Attention, and Cross-Attention layers. Hence, the total trainable parameters only comprise $\boldsymbol{H}_0^{\text{p}}$, $\boldsymbol{W}_{\text{p}}$, $\boldsymbol{W}_{\text{a}}$, $\boldsymbol{W}_{\text{e}}$, $\boldsymbol{z}^{\text{pos}}$, and g_l. Moreover, the proposed adaptor can learn in a semi-supervised manner using pseudo-separated tracks, pseudo MIDI, and pseudo chord labels. Additionally, during training, each music piece is assigned to a vague text description randomly sampled from a small set of predefined phrases, eliminating the requirement for text-audio data pairs.

5. EXPERIMENT

5.1 Datasets

The training dataset consists of 299 *unannotated instrumental* songs. We collect 150 of them from an open-source dataset MUSDB18 [25] and download the remain 149 songs from the internet. The latter subset predominantly consists of Pop songs, with a limited data of other genres such as Jazz and Rock. We omit the silent start and end segments of each training song, yielding 17.12 hours of audio. We employ Demucs to extract drum tracks, a chord recognition model [22], an automatic transcription model [26], and a beat tracking model [24] to generate pseudo chord, MIDI, and beat labels.

The test set comprises 50 songs with chord, beat, and MIDI annotations from RWC-POP-100 [27]. Vocals are excluded by a music source separation model Demucs [28].

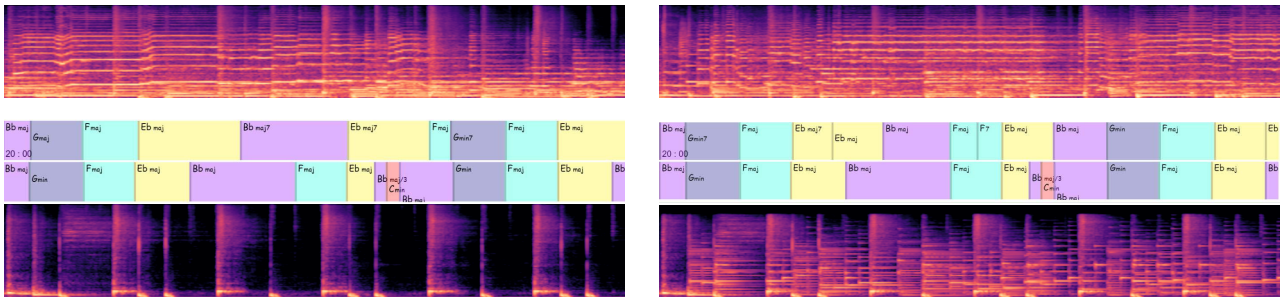
5.2 Training Configuration

We train the proposed model using 4 RTX8000s with an initial learning rate of 2e-3 and a batch of 24 20-second samples for 10 epochs. We set the warm-up epoch to 2 and update the model using a cross-entropy reconstruction loss. During training, for each audio sample, we simply sample a text prompt from a text description set for each music segment: {*melodic music, catchy song, a song, music tracks*}.

5.3 Evaluation

We separate each audio clip in the test set into 4 stems using Demucs and discard the vocal track. As shown in Table 2, *"Chord-only"* signifies no drums and MIDI controls, while *"Full"* indicates both drums and MIDI controls. Within each group, we generate 16 20-second audio samples per given chord progression, employing various text prompts while keeping the same chord unchanged. In total, we have 4 test groups, each with 800 generated audio samples.

We report weighted recall score for chord accuracy, standard F-measure for rhythm control, CLAP [29] score for text control evaluation, and Fréchet Audio Distance (FAD) [30] for audio quality measure. As depicted in Table 2, $\text{Chord}_{\text{rec}}$ represents chord root accuracy, and $\text{Chord}^{*}_{\text{rec}}$ assesses full chord accuracy. $\text{FAD}^{*}_{\text{vgg}}$ quantifies the audio dissimilarity between generated samples



(a) Chord and rhythm control

(b) Chord and rhythm control with MIDI hints

Figure 3. Comparison of generated samples and groundtruth. The top two rows are generated samples, while the bottom rows are reference soundtracks. The text prompt is *"lazy jazz composition features a captivating saxophone solo that effortlessly melds with piano chords, skillfully weaving its way through the melody with languid grace. Instruments: saxophone, piano, drums"*.

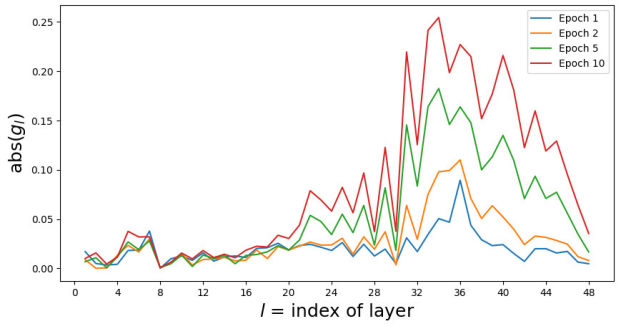


Figure 4. The variation of $|g_l|$ during training.

and groundtruth audio from the RWC-POP-100 subset, whereas FAD_{vgg} assesses this dissimilarity using the remaining 50 audios within RWC-POP-100.

5.4 Results

As illustrated in Table 2 and Figure 3, our model excels in chord and rhythm control while maintaining the text-conditioned ability, even though we do not train the model with real text annotations. We do not report chord and beat accuracy for the baseline model, as it cannot use these controls. Code details and more demos are publicly available online.

5.4.1 Low-resource Fine-tuning

During fine-tuning, our observations suggest that our model works better with a smaller training dataset characterized by high-quality audio fidelity than a larger one featuring pseudo-separated instrumental ground truth. As indicated in Table 3, we discern a trade-off between controllability and semantic correlation. As the number of trainable layers increases, the model achieves a simultaneous improvement in chord recall score while witnessing a reduction in CLAP_{src}. In addition, as shown in Fig 4, as the layers go deeper, the absolute value of gate factor g_l increases. It demonstrates the proposed adapter primarily affects the topmost layers of the decoder transformer, implying that the lower layers are likely responsible for modeling high-level semantics, while the upper layers shape the finer details of the content.

5.4.2 Chord and Rhythm Control

As shown in Table 2 and Fig 3(a), the chord control capability of our model strengthens as MIDI hints are provided. Additionally, the results highlight the model's adeptness at rhythm control when a drum track is included. However, in cases of semantic conflict between the content-based condition and the text prompt, we observe that the model tends to prioritize the former, leading to music that aligns with the drum pattern rather than the prompt.

5.4.3 Variation Generation and Arrangement

As depicted in Fig 3(b), our model, with the assistance of MIDI hints, can produce variations by integrating musical elements from conditioned MIDI tracks, such as motifs and walking bass. Furthermore, we have observed instances where the generated audio aligns with the original main or counter melodies. This facilitates idea-driven variation generation and semantic-based arrangement within the provided polyphonic piano roll.

6. CONCLUSION

We present a content-based music generative model, achieved by fine-tuning a pre-trained Transformer-based audio language model using the proposed condition adaptor. Our experimental results substantiate its proficiency in seamlessly integrating chord progressions, rhythm patterns, MIDI, and text prompts into the generated music. Our work bridges the gap of direct control via musical elements and audio conditions in the music audio generation field. Furthermore, the proposed condition adaptor facilitates efficient low-resource fine-tuning, even with a relatively small unannotated training set. Nonetheless, results generated with conflicting audio, MIDI, and text prompts may lack musicality and may not fully meet semantic control expectations. In the future, we aim to explore further enhancements in the areas of harmonic direct control and content-based generation.

7. ACKNOWLEDGMENTS

This work was supported in part through the NYU and NYUSH IT High Performance Computing resources, services, and staff expertise.

8. REFERENCES

[1] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[2] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," *arXiv preprint arXiv:2301.12503*, 2023.

[3] F. Schneider, Z. Jin, and B. Schölkopf, "Moûsai: Text-to-music generation with long-context latent diffusion," *arXiv preprint arXiv:2301.11757*, 2023.

[4] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *arXiv preprint arXiv:2306.05284*, 2023.

[5] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank *et al.*, "Noise2music: Text-conditioned music generation with diffusion models," *arXiv preprint arXiv:2302.03917*, 2023.

[6] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "MusicLM: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[8] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending clip to image, text and audio," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 976–980.

[9] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis, "Mulan: A joint embedding of music audio and natural language," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, P. Rao, H. A. Murthy, A. Srinivasamurthy, R. M. Bittner, R. C. Repetto, M. Goto, X. Serra, and M. Miron, Eds., 2022, pp. 559–566.

[10] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour *et al.*, "Singsong: Generating musical accompaniments from singing," *arXiv preprint arXiv:2301.12662*, 2023.

[11] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. Qiao, "Llama-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv preprint arXiv:2303.16199*, 2023.

[12] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[13] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[14] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2692–2703, 2024.

[15] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[17] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4582–4597.

[18] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang, "P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 61–68.

[19] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2021.

[20] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue *et al.*, "Llama-adapter v2: Parameter-efficient visual instruction model," *arXiv preprint arXiv:2304.15010*, 2023.

[21] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.

[22] J. Jiang, K. Chen, W. Li, and G. Xia, "Large-vocabulary chord transcription via chord structure decomposition," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*,

A. Flexer, G. Peeters, J. Urbano, and A. Volk, Eds., 2019, pp. 644–651.

[23] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I. Wei, J.-Y. Hsu, Y.-C. Chuang, L. Su *et al.*, "Omnizart: A general toolbox for automatic music transcription," *The Journal of Open Source Software*, vol. 6, no. 68, p. 3391, 2021.

[24] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new python audio and music signal processing library," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1174–1178.

[25] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-a corpus for music separation," 2017.

[26] Y.-T. Wu, Y.-J. Luo, T.-P. Chen, I.-C. Wei, J.-Y. Hsu, Y.-C. Chuang, and L. Su, "Omnizart: A general toolbox for automatic music transcription," *Journal of Open Source Software*, vol. 6, no. 68, p. 3391, 2021. [Online]. Available: https://doi.org/10.21105/joss.03391

[27] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases," in *Proceedings of the 3rd International Society for Music Information Retrieval Conference*, 2002.

[28] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep extractor for music sources with extra unlabeled data remixed," *arXiv preprint arXiv:1909.01174*, 2019.

[29] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[30] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2018.