MuseControlLite: Multifunctional Music Generation with Lightweight Conditioners

Fang-Duo Tsai [1] **Shih-Lun Wu** [2] **Weijaw Lee** [1] **Sheng-Ping Yang** [1] **Bo-Rui Chen** [1] **Hao-Chung Cheng** [1] **Yi-Hsuan Yang** [1]

Abstract

We propose MuseControlLite, a lightweight mechanism designed to fine-tune text-to-music generation models for precise conditioning using various time-varying musical attributes and reference audio signals. The key finding is that positional embeddings, which have been seldom used by text-to-music generation models in the conditioner for text conditions, are critical when the condition of interest is a function of time. Using melody control as an example, our experiments show that simply adding rotary positional embeddings to the decoupled cross-attention layers increases control accuracy from 56.6% to 61.1%, while requiring 6.75 times fewer trainable parameters than state-of-the-art fine-tuning mechanisms, using the same pre-trained diffusion Transformer model of Stable Audio Open. We evaluate various forms of musical attribute control, audio inpainting, and audio outpainting, demonstrating improved controllability over MusicGen-Large and Stable Audio Open ControlNet at a significantly lower fine-tuning cost, with only 85M trainable parameters. Source code, model checkpoints, and demo examples are available at: https://MuseControlLite.github.io/web/.

1. Introduction

Text-to-music generation models have recently gained popularity as they hold the promise of empowering everyone to create high-quality and expressive music without much musical training and a reduced time cost (Copet et al., 2024). However, for people who desire to be more deeply involved in the creation process itself rather than the final output only, mechanisms to exert controllability going beyond the simple text prompt have been considered critical. As such, recent months have seen an increasing body of research concerning the addition of fine-grained, time-varying control to text-to-music generation, such as those related to musical aspects of chords, rhythm, melody, and dynamics. Exemplars include Music ControlNet (Wu et al., 2024), MusiConGen (Lan et al., 2024), JASCO (Tal et al., 2024), and DITTO (Novack et al., 2024b), to name just a few.

Despite the exciting progress that has been made, we find two avenues for improvement. First, existing models might be over-parameterized. In view of the success of ControlNet (Zhang et al., 2023; Zhao et al., 2024) in adding spatial control for text-to-image generation, a prominent approach has been to use similar idea to fine-tune pre-trained text-to-music models to add conditioners for time-varying conditions. For example, treating mel-spectrograms as images, Music ControlNet (Wu et al., 2024) offers multiple conditions of melody, rhythm and dynamics. Stable Audio Open ControlNet (Hou et al., 2025) further improves audio quality with latent diffusion, adapting the original U-Net-based ControlNet encoder to a diffusion Transformer architecture (Peebles & Xie, 2023). However, the ControlNet approach requires duplicating half of the diffusion model as a trainable copy (Zhang et al., 2023), leading to increased training and inference times. Lighter alternatives for fine-tuning, such as the idea of *decoupled cross-attention* presented in IP-adapter (Ye et al., 2023), can be studied.

Second, while fine-tuning a text-to-music generation model to consider conditions of musical attributes or conditions of reference audio signals (Tsai et al., 2024) have been studied separately, little work has been done to consider both types of conditions at the same time. This capability may enable various creative use cases, such as adding melody conditions while performing audio inpainting or outpainting.

We propose MuseControlLite, a lightweight fine-tuning mechanism with significantly fewer trainable parameters for controllable text-to-music generation. The key finding is that the decoupled cross-attention mechanism (Ye et al., 2023), when augmented with positional embeddings, achieves superior performance while using nearly an order

[1]National Taiwan University, Taipei, Taiwan. [2]Massachusetts Institute of Technology, Cambridge, MA, United States. Correspondence to: Fang-Duo Tsai <fundwotsai2001@gmail.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

of magnitude fewer trainable parameters than ControlNet-based mechanisms (85M vs. 572M). Built on the simple idea of combining the rotary positional embeddings (RoPE) (Su et al., 2024) with decoupled cross-attention, MuseControlLite is a novel framework for controlling time-varying conditions. Our implementation introduces only an additional 8% trainable parameters relative to the diffusion Transformer backbone (Evans et al., 2024c).

Unlike our approach, ControlNet-based models (Wu et al., 2024; Hou et al., 2025) incorporate positional embeddings in the self-attention layers of the Transformer blocks. These models process the conditioning input in the latent space—matching the shape of the pre-trained backbone—and reintroduce it into the frozen U-Net decoder (Wu et al., 2024) or Transformer blocks (Hou et al., 2025) to guide the generation toward the desired output. We find empirically that our approach is more parameter-efficient.

Our model supports multi-attribute control similarly to Music ControlNet (Wu et al., 2024). Moreover, by introducing "audio conditioning," MuseControlLite is able to replicate the reference audio in full resolution while providing partial control capability, enabling both audio inpainting and outpainting. Additionally, we adopt multiple classifier-free guidance (Liu et al., 2022; Brooks et al., 2023) to flexibly regulate the strength of the time-varying conditions, preventing the quality degradation that can result from over-fixation.

The main contributions are three-fold:

- The first investigation of using positional embeddings for decoupled cross-attention layers in diffusion Transformers for controllable text-to-music generation.
- The first fine-tuning method that handles both attribute and audio control ('text+attribute+audio'). In contrast, existing finetuning methods only take either attribute control ('text+attribute') (Wu et al., 2024), or audio control ('text+audio') (Tsai et al., 2024).
- Demonstration on the public evaluation benchmark of the Song Describer dataset (Manco et al., 2023), showing that MuseControlLite outperforms existing ControlNet-based approaches (Wu et al., 2024; Hou et al., 2025) in melody control, achieving a 4.5% improvement in melody accuracy.

2. Related Work

Controllable music generation aims to produce music that aligns with human requirements. On a global scale, controls may include text prompts, tempo (BPM), instrumentation, timbre, or mood. On a more fine-grained or local level, control can be exercised over chords, rhythm, dynamics, melodic lines, and other structural elements, usually as time-varying conditions. We review some existing work below.

Training-time control with global conditions. One of the most common methods to steer a music generation model is direct fine-tuning. Plitsis et al. (2024) explores personalized techniques from image generation, such as DreamBooth (Ruiz et al., 2023) and textual inversion (Gal et al., 2022). Mustango (Melechovsky et al., 2023), trained with music-focused textual captions, learns to understand instructions about chords, tempo, and key. Instruct-MusicGen (Zhang et al., 2024) fine-tunes MusicGen, enabling music editing with text prompts. MusicGen-style (Rouard et al., 2024) trains MusicGen from scratch, incorporating a text conditioner and an audio feature extractor to fuse text and audio during inference.

Training-time control with local conditions. For more granular controls, MusicGen (Copet et al., 2024) prepends a melody-based conditioning tensor (along with text embeddings) and trains the entire model from scratch. Similarly, JASCO (Tal et al., 2024) appends all conditions to the model's input across the feature dimension but also requires training from scratch—demanding significant computational resources and making it less flexible to add new conditions. MusiConGen (Lan et al., 2024) and Coco-Mulla (Lin et al., 2023) fine-tune MusicGen, enabling chord and rhythm conditions without full retraining. Music ControlNet (Wu et al., 2024) uses zero-initialized convolution layers and a trainable copy as an adapter for fine-tuning.

Inference-time optimization. Training-free approaches for controllable generation also gain attention for their computational efficiency and adaptability (Levy et al., 2023; Novack et al., 2024b; Kim et al., 2024). These methods leverage pre-trained models directly, avoiding additional model training. However, training-free methods often encounter inherent quality limitations compared to fully trained models. In our work, we address similar objectives with MuseControlLite, which is trained on open-source data and provides functionalities similar to DITTO (Novack et al., 2024b).

3. MuseControlLite

3.1. Diffusion Background

Diffusion models (Ho et al., 2020; Song et al., 2020) operate in two stages: a forward process that progressively corrupts a clean sample $\mathbf{x}_0$ over T time steps with noise (forming $\mathbf{x}_1, \ldots, \mathbf{x}_T$), and a reverse process that is learned to invert the corruption step by step. Diffusion models are often trained to predict the noise $\boldsymbol{\epsilon}$ added at each time step by minimizing a mean squared error denoising loss.

While early diffusion models use U-Net-like architectures for denoising, diffusion Transformers (Peebles & Xie, 2023; Xie et al., 2024) emerge as a more effective way to capture long-range dependencies in data through the attention mechanisms. During training, each noisy sequence $\mathbf{x}_t$ is passed

through the Transformer along with a time-step embedding to predict ϵ, $\mathbf{x}_0$ or other intermediate variables (Salimans & Ho, 2022), depending on the chosen parameterization. By iteratively refining $\mathbf{x}_t$ through this denoising loop, a coherent musical sequence can be reconstructed.

In our implementation, we use Stable Audio Open (Evans et al., 2024c), an open-source text-to-music generation model based on a diffusion Transformer with 24 diffusion blocks. Each block contains both self-attention and cross-attention layers. As will be described later, we modify the cross-attention layers to consider time-varying conditions.

3.2. Rotary Position Embedding (ROPE) Background

Absolute positional embeddings (Radford et al., 2019; Clark, 2020) add a learned or sinusoidal vector (Vaswani, 2017) to each Transformer token, and relative positional embeddings (Shaw et al., 2018) incorporate distance offsets between tokens in attention. ROPE (Su et al., 2024) instead rotates query and key vectors by a position-dependent angle, embedding both absolute and relative information more directly. This is encapsulated by the following equations:

$$\mathbf{q}_m^T \mathbf{k}_n = \left(\mathbf{R}^d_{\Theta,m} \mathbf{W}^q \mathbf{x}_m\right)^\top \left(\mathbf{R}^d_{\Theta,n} \mathbf{W}^k \mathbf{x}_n\right), \tag{1}$$

$$\mathbf{R}^d_{\Theta,m}\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{d-1} \\ x_d \end{pmatrix} \begin{pmatrix} \cos m\theta_1 \\ \cos m\theta_1 \\ \vdots \\ \cos m\theta_{d/2} \\ \cos m\theta_{d/2} \end{pmatrix} + \begin{pmatrix} -x_2 \\ x_1 \\ \vdots \\ -x_d \\ x_{d-1} \end{pmatrix} \begin{pmatrix} \sin m\theta_1 \\ \sin m\theta_1 \\ \vdots \\ \sin m\theta_{d/2} \\ \sin m\theta_{d/2} \end{pmatrix} \tag{2}$$

where $\mathbf{x}_m$ and $\mathbf{x}_n$ in Eq. (1) are embeddings for tokens at positions m and n coming from the last layer, $\mathbf{q}$ and $\mathbf{k}$ the resulting query and key vectors, $\mathbf{W}^q$ and $\mathbf{W}^k$ the projection matrices, and $\mathbf{R}^d_{\Theta,i}$ the rotation matrix with $\Theta = \left\{\theta_i = 10000^{-\frac{2(i-1)}{d}}, \quad i \in \left[1, 2, \ldots, \frac{d}{2}\right]\right\}$, where θ_i for each position index i follows an exponential scaling pattern based on the dimension d. Note that in Eq. (2) we drop the subscript of $\mathbf{x}$ for simplicity and use x_j to indicate the j-th entry ($j \in [1, 2, \ldots, d]$) of the vector.

3.3. Proposed Adapter Design

To prevent over-parametrization yet still achieve the desired fine-tuning performance, Custom Diffusion (Kumari et al., 2023) demonstrates that fine-tuning only $\mathbf{W}^k$ (for key vectors) and $\mathbf{W}^v$ (for value vectors) can suffice for producing personalized outputs with a given new image condition. IP-Adapter (Ye et al., 2023) further improves this approach by adopting a **decoupled cross-attention** mechanism that trains only $\mathbf{W}'^k$ and $\mathbf{W}'^v$ in the "decoupled" layers to learn new conditions related to image generation, where $\mathbf{W}'^k$ and $\mathbf{W}'^v$ are learnable copy of (and with parameters initialized from) $\mathbf{W}^k$ and $\mathbf{W}^v$, respectively. In view that time-varying conditions in the music/audio domain may be treated similarly as spatial conditions in the image domain, we propose to employ decoupled cross-attention to music generation.

For the original cross-attention layers handling the text condition (i.e., the pink-shaded component in the middle of Figure 1), we leave it unchanged:

$$\mathbf{x}_{\text{text}} := \text{Attention}\left(\mathbf{x}\,\mathbf{W}^q,\ \mathbf{c}_{\text{text}}\,\mathbf{W}^k,\ \mathbf{c}_{\text{text}}\,\mathbf{W}^v\right), \tag{3}$$

where $\mathbf{c}_{\text{text}}$ denotes the embedding representing the text condition.

Transferring the technique from the traditional U-Net, we adapt the decoupled cross-attention layers to fit the diffusion Transformer, which calls for additional positional encodings. Specifically, in MuseControlLite, beyond the text condition $\mathbf{c}_{\text{text}}$, we introduce to the decoupled layers (i.e., the musical attribute pipeline within the green-shaded component in Figure 1) an additional musical attribute condition $\mathbf{c}_{\text{attr}} = \{\mathbf{c}_n\}_{n=1}^N$, which is a function of time with N time points, and $\mathbf{c}_n$ is at the time position n of the sequence $\mathbf{c}_{\text{attr}}$. We apply ROPE to the query, key, and value vectors to enhance the model's positional awareness. We then train the duplicated $\mathbf{W}'^k$ and $\mathbf{W}'^v$ to handle $\mathbf{c}_{\text{attr}}$:

$$\mathbf{q}_m = \mathbf{R}^d_{\Theta,m} \mathbf{W}^q \mathbf{x}_m, \tag{4}$$

$$\mathbf{k}_n = \mathbf{R}^d_{\Theta,n} \mathbf{W}'^k \mathbf{c}_n, \tag{5}$$

$$\mathbf{v}_n = \mathbf{R}^d_{\Theta,n} \mathbf{W}'^v \mathbf{c}_n. \tag{6}$$

For the decoupled cross-attention, we combine the rotated sequences and calculate the attention:

$$\mathbf{x}_{\text{attr}} := \text{Attention}\left(\mathbf{Q}_{\text{rope}}, \mathbf{K}_{\text{rope}}, \mathbf{V}_{\text{rope}}\right), \tag{7}$$

where $\mathbf{Q}_{\text{rope}} = \{\mathbf{q}_m\}_{m=1}^M$, $\mathbf{K}_{\text{rope}} = \{\mathbf{k}_n\}_{n=1}^N$, $\mathbf{V}_{\text{rope}} = \{\mathbf{v}_n\}_{n=1}^N$, where M is the length of the musical audio sequence (N is proportional to M but they can be different in general). Finally, following Zhang et al. (2023), we add the cross-attention outputs together and connect them with a zero-initialized (zero-init) 1D convolutional layer Z_{CNN} to eliminate initial noise at the start of training. We regard this linear superposition as a correction to the query representation based on the given condition:

$$\mathbf{x} = Z_{\text{CNN}}(\mathbf{x}_{\text{text}} + \mathbf{x}_{\text{attr}}). \tag{8}$$

3.4. Applications for Controls and Manipulations

We first demonstrate that our model is applicable to all time-varying signals used in Music ControlNet (Wu et al., 2024). Next, we incorporate an additional audio condition to enable audio inpainting and outpainting, treating audio signals as another type of control signals.

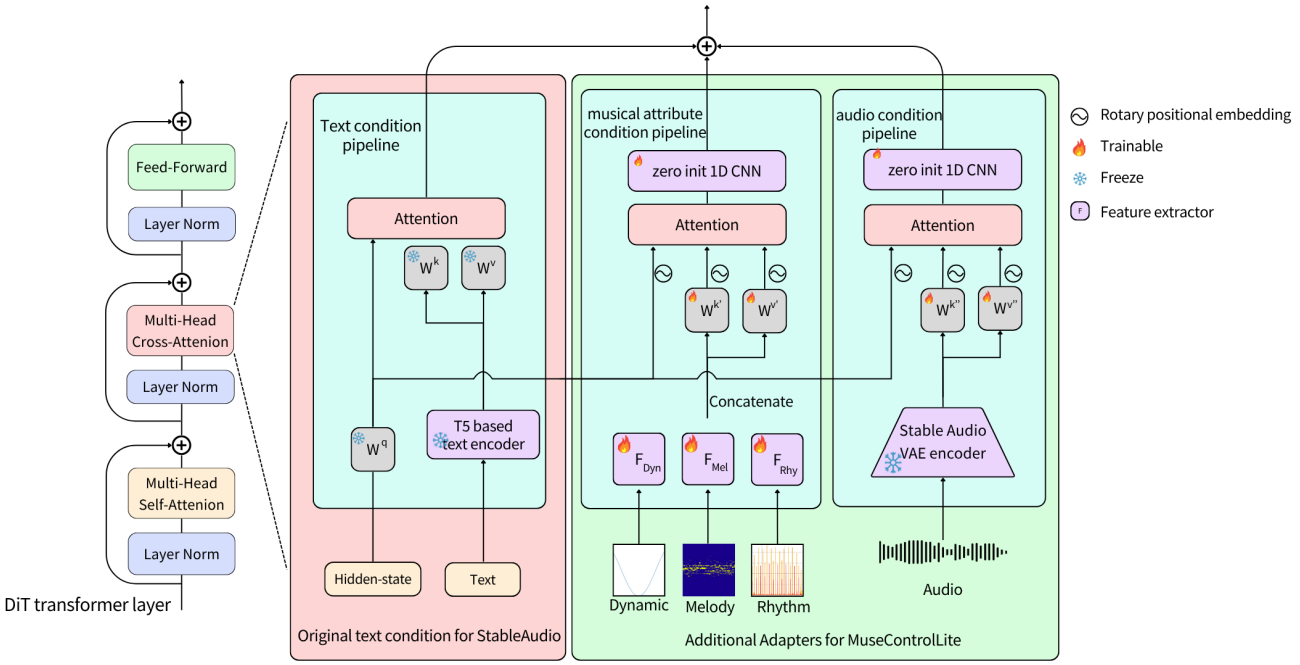

Figure 1. MuseControlLite uses multiple condition extractors to handle all time-varying controls. The musical attribute offers control over elements such as melody, rhythm, and dynamics, whereas the audio condition enables audio inpainting and outpainting. We train the two pipelines separately with two sets of adapters, but users can choose to use either one or both at the inference time.

Table 1. We trained both models for 70,000 steps with a batch size of 32 using the melody condition and found that the one without ROPE struggles to learn the new condition.

	FD↓	KL↓	CLAP↑	Mel acc.↑
w/o ROPE	113.13	0.58	**0.41**	10.7%
w/ ROPE	**78.50**	**0.29**	0.38	**58.6%**

Musical Attribute Control We follow Music ControlNet (Wu et al., 2024) to extract rhythm and dynamics, but adopt a method similar to that of Hou et al. (2025) for melody extraction. For melody $\mathbf{c}_{\text{mel}} \in \mathbb{R}^{N_{\text{melody}} \times 128}$, we first compute the CQT with 128 bins for the mean of the left and right audio channels,[1] apply an argmax operation, then to retain the four most prominent pitches per frame, followed by a high-pass filter (cutoff at 261.2 Hz) to suppress the bass. Dynamics $\mathbf{c}_{\text{dyn}} \in \mathbb{R}^{N_{\text{dynamics}} \times 1}$ are derived from the spectrogram energy, converted to decibels, and smoothed with a Savitzky-Golay filter to align with perceived intensity. Rhythm $\mathbf{c}_{\text{rhy}} \in \mathbb{R}^{N_{\text{rhythm}} \times 2}$ is extracted using a recurrent network-based beat detector (Böck et al., 2016) that estimates beat and downbeat probabilities, enabling precise synchronization and rhythmic nuance.

We use separate 1D CNN layers to extract features from each condition and expand their channel dimensions to $C_r/3$, where C_r is the cross-attention dimension. We then use PyTorch's `interpolate` function to match the sequence lengths of these features, setting $N_{\text{interpolate}}$ equal to the query length M. Finally, we concatenate the features along the last dimension, resulting in the condition representation $\mathbf{c}_{\text{attr}} \in \mathbb{R}^{N_{\text{interpolate}} \times C_r}$ for the decoupled cross-attention input.

During training, we apply a masking strategy similar to Music ControlNet (Wu et al., 2024), which randomly masks 10% to 90% of the condition, and the masks are independent for the three conditions (i.e., melody, dynamics, rhythm). By using such partial conditioning, we find that the model learns to "disentangle" these conditions and can improvise for the unconditioned segments. For example, we can specify music attribute conditions only for the 10–20-second segment, leaving the 0–10-second and 20–47-second segments blank for the model to improvise.

Audio Inpainting and Outpainting We directly use the VAE-encoded latent $\mathbf{x}_0 \in \mathbb{R}^{N_{audio} \times A}$, referred to as the "clean latent" for StableAudio, as $\mathbf{c}_{\text{audio}}$, where N_{audio} is the length of the encoded audio and A is the number of audio latent channels. Since $\mathbf{c}_{\text{audio}}$ provides far more information than $\mathbf{c}_{\text{attr}}$, we found that training with both $\mathbf{c}_{\text{attr}}$ and $\mathbf{c}_{\text{audio}}$ simultaneously can cause the model to ignore $\mathbf{c}_{\text{attr}}$. Thus,

[1] We noticed that Stable Audio Open ControlNet computes the CQT for the left and right channels separately yields a finer-grained melody condition. Adopting this method in our implementation improved melody accuracy to 64.5% after only 18,000 training steps.

the audio condition $\mathbf{c}_{\text{audio}}$ is trained separately using a distinct set of adapters, while the decoupled cross-attention layers remain the same as those used in the musical attribute conditioning pipeline. The audio condition is applied with a complementary mask to $\mathbf{c}_{\text{attr}}$ so that there is no overlap between $\mathbf{c}_{\text{attr}}$ and $\mathbf{c}_{\text{audio}}$.

By applying masks to $\mathbf{c}_{\text{audio}}$ during training, $\mathbf{W}''^{k}$ and $\mathbf{W}''^{v}$ learn not only to reflect the condition $\mathbf{k}_n$ on the same position $\mathbf{q}_n$ but also to attend to distant tokens $\mathbf{k}_m$, as shown in Figure 2 in the appendix. This yields smooth transitions at the boundary where the audio condition is given and where it is masked. MuseControlLite can thus simultaneously achieve partial audio control and music-attribute control. Since the segments controlled by $\mathbf{c}_{\text{audio}}$ are more rigid, we propose to use musical attribute conditions to flexibly control the masked audio segments.

3.5. Multiple Classifier-Free Guidance

Classifier-free guidance (Ho & Salimans, 2022) is an inference-time method that trades off sample quality and diversity in diffusion models. It is often used in text-conditional audio or image generation to improve text adherence. Song et al. (2020) provides a crucial interpretation that each denoising step can be viewed as ascending along $\nabla_x \log p_\theta(\mathbf{x})$, the *score* of $p_\theta(\mathbf{x})$. Additionally, any input condition $\mathbf{c}$ can be incorporated into a diffusion model by injecting the embeddings of $\mathbf{y}$ into cross-attention (Rombach et al., 2022), thus modeling $p_\theta(\mathbf{x}|\mathbf{c})$ (and $\nabla_x \log p_\theta(\mathbf{x}|\mathbf{c})$). Ho & Salimans (2022) shows that we can train a model by randomly dropping the condition, thereby learning both $\nabla_x \log p_\theta(\mathbf{x})$ and $\nabla_x \log p_\theta(\mathbf{x}|\mathbf{c})$. Specifically, this procedure enables $\nabla_x \log p_\lambda(\mathbf{x}|\mathbf{c}) = \nabla_x \log p(\mathbf{x}) + \lambda_{\text{text}}\big(\nabla_x \log p(\mathbf{x}|\mathbf{c}) - \nabla_x \log p(\mathbf{x})\big)$. In our work, we use a single pipeline to model $p_\theta(\mathbf{x}|\mathbf{c}_{\text{attr}}, \mathbf{c}_{\text{audio}})$ during training. Our pilot study finds that a fine-tuned model often over-fits to the additional conditions $\mathbf{c}_{\text{attr}}$ or $\mathbf{c}_{\text{audio}}$. Therefore, we adopt a separated guidance scale (Brooks et al., 2023), λ_{attr} for musical attributes and λ_{audio} for audio:

$$\nabla_x \log p_\lambda(\mathbf{x}|\mathbf{c}) = \nabla_x \log p(\mathbf{x}) + \sum_{i \in \{\text{text,attr audio}\}} \lambda_i \Big(\nabla_x \log p(\mathbf{x}|\mathbf{c}_{\le i}) - \nabla_x \log p(\mathbf{x}|\mathbf{c}_{<i})\Big). \tag{9}$$

As detailed in Appendix B, we expand the equation from a single or two conditions (Ho & Salimans, 2022; Brooks et al., 2023) to a general form of multiple conditions.

4. Experimental setup

4.1. Dataset

For training, we utilize the open-source MTG-Jamendo dataset (Bogdanov et al., 2019) and preprocess the data following this pipeline: (i) Resample the audio to 44.1 kHz and segmented into fixed-length clips compatible with the maximum input shape of the Stable Audio Open VAE encoder; (ii) we employ the sound event detection capabilities of PANNs (Kong et al., 2020) to exclude any audio samples containing vocals; (iii) captions for each audio clip are generated using the Qwen2-Audio-7B-Instruct model (Chu et al., 2024). Additionally, we remove any samples that overlap with the Song Describer dataset (Manco et al., 2023), as this dataset is reserved exclusively for evaluation purposes. We will also release the code[2] for this data processing pipeline, aiming to facilitate and accelerate future developments in text-to-music model training and fine-tuning. For evaluation, we adopt the methodology outlined by Evans et al. (2024a;c;b). Specifically, we utilize the instrumental subset of the Song Describer dataset, explicitly excluding any prompts associated with vocals. This yields an evaluation set comprising 586 audio clips. All 586 clips are used in the melody-conditioned generation experiments described in Section 5.1, aligning with the settings in Stable Audio Open ControlNet (Hou et al., 2025). However, using musical attributes and text captions from the same audio does not reflect real-world use cases, as users may wish to generate music conditioned on a melody using arbitrary text prompts, rather than the exact caption of the melody source—a process known as *style transfer*. To demonstrate the style transfer capability of MuseControlLite, the tasks described in Section 5.2 employ a different setup. Ideally, we would have followed the same approach in Section 5.1, but this was not possible due to the closed-source nature of Stable Audio Open ControlNet (Hou et al., 2025). To enable style transfer evaluation, we split the 586 audio clips into two disjoint subsets and generated samples by pairing text prompts from the first subset with musical attributes extracted from the second, ensuring that the attributes were independent of the ground-truth audio. The generated outputs were then evaluated against the first subset as the reference. For the audio inpainting and outpainting tasks in Section 5.3, we randomly selected a smaller subset of 50 clips from the original 586, without applying the style transfer setting. In Section 5.4, to compare with Stable Audio Open ControlNet, which is open-source, we downloaded example outputs from their project website[3] and generated corresponding samples using the same melody condition audio and text prompts as used by other baselines.

4.2. Training and Inference Specifics

We fine-tune the model using the following objective:

$$\mathbb{E}_{t\sim[0,1],x_t} \left\| f_\theta\left(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, t\right) - \mathbf{v}_\mathbf{t} \right\|_2^2 , \tag{10}$$

[2] https://github.com/fundwotsai2001/Text-to-music-dataset-preparation

[3] https://stable-audio-control.github.io/

Table 2. Separated guidance scales used in different tasks

Guidance	λ_{text}	λ_{attr}	λ_{audio}
Musical attribute control	7.0	2.0	✗
Audio inpainting	7.0	2.0	1.0
Audio outpainting	7.0	2.0	1.0

with the velocity term $\mathbf{v_t} = \alpha_t \boldsymbol{\epsilon} + \beta_t \mathbf{x}_0$ following the v-prediction parameterization (Salimans & Ho, 2022) to improve training stability, where α_t and β_t are time-dependent coefficients that balance the contributions of noise and clean data. That is, rather than predicting either the noise or the clean data directly, this parameterization predicts the intermediate variable $\mathbf{v_t}$. The time variable t is sampled from a noise schedule within the interval $t \sim [0, 1]$. The scaling functions are defined as $\alpha_t = \cos(0.5\pi t)$ and $\sigma_t = \sin(0.5\pi t)$.

We freeze all model components except for the adapters, feature extractors, and the zero-initialized 1D convolution layers (as shown in Figure 1). We train two sets of adapters: one conditioned on all musical attributes and the other conditioned only on audio, using identical training configurations. We use a batch size of 128, a constant learning rate of 10^{-4}, and a weight decay of 10^{-2}. To encourage the model to focus on $\mathbf{c}_{\text{attr}}$ or $\mathbf{c}_{\text{audio}}$, we drop the text condition in 30% of training iterations. Additionally, each condition is independently dropped with a probability of 50% and subjected to random masking. The model is trained for 40,000 steps with an effective batch size of 128 on a single NVIDIA RTX 3090. For inference, we fix the separate guidance scales as shown in Table 2. We use 50 denoising steps to generate 47-second audio clips. When both a musical attribute condition and an audio condition are applied, we use a complementary masking strategy. This means that the model is exposed to only one type of condition at a time, as the audio condition can be overly dominant and may cause the model to ignore the musical attribute condition.

4.3. Baselines

Although musical attribute control, audio inpainting, and outpainting have been explored in many prior works, they are either not open-source (e.g., Music ControlNet (Wu et al., 2024), DITTO (Novack et al., 2024b;a), JASCO (Li et al., 2024)) or generate a relatively short audio (Tal et al., 2024). MusiConGen (Lan et al., 2024) and Coco-Mulla (Lin et al., 2023) provide rhythm control; however, *MusiConGen* represents rhythm using a constant beats-per-minute (BPM) value, while *Coco-Mulla* relies on a separate drum track for rhythm representation. We consider both approaches unsuitable for direct comparison with our method.

- **MusicGen** (Copet et al., 2024): MusicGen is a Transformer-based autoregressive model for text-to-music generation. We adopt the MusicGen-Stereo-Large for audio inpainting and outpainting, MusicGen-Stereo-Large-Melody for melody comparison.
- **Stable Audio Open ControlNet** (Hou et al., 2025): The ControlNet structure, widely used in text-to-image generation, can also be applied for text-to-music control. Although Stable Audio Open ControlNet is not open-source (Hou et al., 2025), we employed exactly the same metrics and dataset for comparability. In addition, we contacted the authors to ensure that we followed the same method for extracting melodies and calculating accuracy.
- **Naïve masking**: An inference-time method used in DITTO (Novack et al., 2024b), we implemented it with Stable Audio Open (Evans et al., 2024c) for audio inpainting and outpainting. We initiate the denoising process with random noise $\mathbf{x}_t$, and after each denoising step, we immediately overwrite the "reference" region of $\mathbf{x}_{t-1}$ with a noisy version of a given reference audio.

4.4. Objective Evaluation Metrics

We use the following metrics to evaluate the musical attribute controllability, text adherence, and audio realism. In order to benchmark against Stable Audio Open ControlNet (Hou et al., 2025), we use the same open-source code[4] for calculating $\text{FD}_{\text{openl3}}$, KL_{passt} (Koutini et al., 2021), and $\text{CLAP}_{\text{score}}$ (Wu et al., 2023). $\text{FD}_{\text{openl3}}$ (Cramer et al., 2019) extends Fréchet Distance (FD) to full-band stereo audio using OpenL3 (48kHz) features, enabling more comprehensive similarity evaluation. KL_{passt} measures semantic alignment via KL divergence using PaSST, an AudioSet-trained tagger (Gemmeke et al., 2017), adapting it for long-form audio by segmenting and averaging logits. $\text{CLAP}_{\text{score}}$ (Wu et al., 2023) assesses text-audio correspondence using CLAP embeddings with a feature fusion approach, ensuring robust evaluation of long-form, high-resolution audio.

Melody Accuracy We directly use the code for calculating melody accuracy from Stable Audio Open ControlNet (Hou et al., 2025), provided by the authors, to ensure a fair comparison. The chromagram $C \in \mathbb{R}^{12\times T}$ is computed via STFT with a window size of 2048 and a hop size of 512, and we use `argmax` to select the most prominent pitch. We measure the agreement between the frame-wise pitch classes (C, C#, ..., B; 12 in total) of the reference audio and the generated audio. Higher accuracy indicates better preservation of the intended melody.

[4] https://github.com/Stability-AI/stable-audio-metrics

Table 3. Melody control comparing with state-of-the-art controllable text-to-music generation models. The proposed model achieves the best melody accuracy and acceptable musical quality metrics with fewer trainable parameters and training data.

Model	Trainable Parameters	Total Parameters	Training Data	FD ↓	KL ↓	CLAP ↑	Mel acc. ↑
MusicGen-Stereo-Large-Melody	3.3B	3.3B	20K hr	193.66	0.436	0.354	43.1%
Stable Audio Open ControlNet	572M	1.9B	2.2K hr	97.73	**0.265**	**0.396**	56.6%
Ours (MuseControlLite-Melody)	**85M**	**1.4B**	1.7K hr	**76.42**	0.289	0.372	**61.1%**
Ours (MuseControlLite-Attr)	**85M**	**1.4B**	1.7K hr	80.79	0.271	0.373	60.6%

Table 4. Performance of all combinations of controls using text and $\mathbf{c}_{\text{attr}}$ from different subsets of Song Describer Dataset (Manco et al., 2023) (style transfer task). Controllability significantly improves when the relevant condition is provided to the model (✓).

	melody	rhythm	dynamics	FD ↓	KL ↓	CLAP ↑	Mel acc. ↑	Rhy F1 ↑	Dyn cor. ↑
Text condition only	✗	✗	✗	124.77	0.59	**0.42**	0.09	0.21	0.05
Text+single musical attribute	✓	✗	✗	**89.89**	0.54	0.28	0.60	0.76	0.66
	✗	✓	✗	111.03	**0.45**	0.38	0.09	0.89	0.42
	✗	✗	✓	136.40	0.54	0.39	0.09	0.30	0.92
Text+double musical attributes	✓	✓	✗	90.00	0.56	0.28	**0.61**	0.89	0.73
	✓	✗	✓	92.73	0.55	0.28	0.60	0.78	0.94
	✗	✓	✓	119.20	0.46	0.35	0.08	0.89	0.93
Text+all musical attributes	✓	✓	✓	94.50	0.55	0.28	**0.61**	**0.90**	**0.95**

Table 5. Performance of all combinations of controls using text and $\mathbf{c}_{\text{attr}}$ from the same subset of Song Describer Dataset (Manco et al., 2023) (non-style transfer task).

	melody	rhythm	dynamics	FD ↓	KL ↓	CLAP ↑	Mel acc. ↑	Rhy F1 ↑	Dyn cor. ↑
Text condition only	✗	✗	✗	124.77	0.58	**0.42**	0.09	0.22	0.06
Text+single musical attribute	✓	✗	✗	**83.89**	0.31	0.37	**0.61**	0.77	0.67
	✗	✓	✗	104.83	0.41	0.40	0.09	0.86	0.48
	✗	✗	✓	133.43	0.52	0.40	0.09	0.32	0.92
Text+double musical attributes	✓	✓	✗	86.62	0.30	0.39	**0.61**	**0.89**	0.74
	✓	✗	✓	88.30	**0.28**	0.38	0.60	0.78	0.94
	✗	✓	✓	110.60	0.36	0.40	0.09	0.88	0.93
Text+all musical attributes	✓	✓	✓	90.28	0.31	0.39	**0.61**	**0.89**	**0.95**

Dynamics Correlation Following Wu et al. (2024), we compute Pearson's correlation to measure the strength of the linear relationship between the dynamics curve of the generated audio and the ground truth condition.

Rhythm F1 Following Wu et al. (2024), the Rhythm F1 score is computed following standard beat and downbeat detection evaluation methods (Davies et al., 2009; Raffel et al., 2014; Wu et al., 2024). It measures the alignment between beat and downbeat timestamps estimated from the input rhythm control and those extracted from the generated audio. These timestamps are obtained using a Hidden Markov Model (HMM) post-filter (Krebs et al., 2015) applied to frame-wise beat and downbeat probabilities. In accordance with (Raffel et al., 2014; Wu et al., 2024), an input and generated (down)beat timestamp are considered aligned if their temporal difference is less than 70 milliseconds.

Smoothness Value To evaluate the smoothness of boundaries in audio inpainting and outpainting tasks, we adopt the Novelty-Based Segmentation approach (Müller, 2015). First, we compute a linear spectrogram and construct the self-similarity matrix (SSM) by measuring pairwise similarities between feature vectors at different time frames. The SSM is then convolved with a checkerboard-shaped kernel centered at each diagonal position. This kernel is designed to emphasize local changes by contrasting regions of high and low similarity. We select a kernel size of 3 to detect novelty on a short time scale, with other hyperparameters same as the tutorial[5] of Müller (2015). Finally, the convolution values are summed for each time position, forming a

[5] https://www.audiolabs-erlangen.de/resources/MIR/FMP/C4/C4S4_NoveltySegmentation.html

one-dimensional curve known as the novelty curve V. This curve represents abrupt changes in musical content over time, with peaks typically indicating segment boundaries. We define the Smoothness Value as the second finite difference of the position V_i, which is $V_{i-1} - 2V_i + V_{i+1}$, where i represents an inpainting or outpainting boundary position. A lower Smoothness Value implies non-smooth transition.

5. Result

5.1. Melody conditioned generation Comparison

To compare melody-conditioned generation with Stable Audio Open ControlNet (Hou et al., 2025) and MusicGen-Melody (Copet et al., 2024), we train another version of MuseControlLite that learns only from melody conditions. This version, referred to as **MuseControlLite-Melody**, is trained for 40,000 steps with a batch size of 128. Additionally, we report results for adapters trained with all musical attributes but used for melody-only conditioned generation, referred to as **MuseControlLite-Attr**. The results are presented in Table 3. Despite having fewer trainable parameters and less training data, both MuseControlLite-Melody and MuseControlLite-Attr outperform other baselines in terms of FD and melody accuracy, demonstrating superior controllability and realism. However, our models perform slightly worse than Stable Audio Open ControlNet (Hou et al., 2025) on KL and CLAP metrics. This may be due to our training data being sourced solely from the MTG-Jamendo dataset (Bogdanov et al., 2019), whereas Stable Audio Open ControlNet (Hou et al., 2025) uses a more diverse dataset consisting of MTG-Jamendo (Bogdanov et al., 2019), FMA (Defferrard et al., 2016), MTT (Law et al., 2009), and Wikimute (Weck et al., 2024).

5.2. Ablation Study for Musical Attribute Conditions

We evaluate all combinations of controls on both style transfer and non-style transfer tasks using MuseControlLite-Attr, with the results presented in Tables 4 and 5. While MuseControlLite-Attr is trained on all musical attribute conditions, it allows users to perform inference with any combination of these conditions.The style transfer task uses musical attributes and text from different subsets, while the non-style transfer task uses musical attributes and text from the same subset. The results in Table 4 are consistent with what reported in Music ControlNet (Wu et al., 2024). When the condition is provided, the melody accuracy, rhythm F1, and dynamics correlation are significantly higher. Additionally, we found that when only the melody condition is given, the generated audio still achieves a high rhythm F1 and dynamics correlation, suggesting that the melody condition provides rich information that may include rhythm and dynamics cues. Comparing Tables 4 and 5, we find that MuseControlLite-Attr performs worse on the style transfer task for FD, KL, and CLAP. The degradation of CLAP and KL in style transfer when using melody controls may indicate that the melody condition itself includes the timbre or genre information of the original audio. Hence, the generated audio still retains a certain level of information relevant to the original audio. When rhythm and/or dynamics controls are applied, the degradation of CLAP and KL is milder. On the other hand, musical attribute controllability remains approximately the same for both the style transfer and non-transfer tasks.

5.3. Audio Outpainting and Inpainting

Audio Outpainting We mask a 47-second audio clip, retaining only the first 24 seconds, and experiment with no attribute controls and single musical attribute controls using MuseControlLite (i.e., the one trained to consider both attribute controls and audio controls) and other baselines. We trim the first 24 seconds and retain the outpainting parts, as we aim to evaluate only the newly generated segments. As shown in Table 6, our text condition only outpainting outperforms MusicGen-Large in all aspects except for Rhythm F1, demonstrating superior audio realism and consistency. Notably, our non-autoregressive model achieves better objective results than the state-of-the-art autoregressive model, despite autoregression being intuitively preferred for continuation tasks. This suggests that our model effectively learns to improvise missing segments using cross-attention layers, even when no audio condition is provided. While the naive masking method achieves better text adherence according to the CLAP score, it has the lowest smoothness value, indicating abrupt changes at the boundaries.

Audio Inpainting To evaluate audio inpainting, we retain only the first and last 5 seconds of the reference audio, testing the model's ability to fill in the missing middle portion. Similar to the outpainting task, we exclude the reference segments from the evaluation and assess only the generated inpainted part (i.e., the generated audio from 16s to 32s). The results, presented in Table 7, show that in terms of audio realism and text adherence, the performance is similar to that of audio outpainting. However, musical attribute control appears to be more challenging. A possible reason is that, in audio inpainting, the model must handle two transitions—one at the beginning and one at the end—rather than simply continuing the sequence, making the task inherently more complicated.

5.4. Subjective Evaluation

The task for subjective evaluation is similar to Section 5.1, but the dataset consists of samples from the demo website of Stable Audio ControlNet (Hou et al., 2025). We apply the

Table 6. Result for audio outpainting task.

	Model	FD ↓	KL ↓	CLAP ↑	Mel acc. ↑	Rhy F1 ↑	Dyn cor. ↑	Smoothness (at 24s) ↑
Baseline	MusicGen-Stereo-Large-Melody	207.48	0.38	0.32	0.19	0.71	0.13	–0.18
	Naïve masking	193.92	0.64	**0.39**	-0.07	0.23	0.11	–0.40
Ours	Text+Audio	184.55	0.37	0.34	0.14	0.45	0.14	–0.16
	Text+Audio+Melody	**138.07**	**0.27**	0.35	**0.57**	0.81	0.43	–0.26
	Text+Audio+Rhythm	165.55	0.32	0.35	0.15	**0.95**	0.26	–0.16
	Text+Audio+Dynamics	189.15	0.32	0.35	0.14	0.55	**0.77**	**0.10**

Table 7. Results for audio inpainting task.

		FD ↓	KL ↓	CLAP ↑	Mel acc. ↑	Rhy F1 ↑	Dyn cor. ↑	Smoothness (at 16s) ↑	Smoothness (at 32s) ↑
Baseline	Naïve masking	193.92	0.64	**0.39**	0.12	0.29	0.07	–0.62	–0.53
Ours	Text+Audio	144.23	**0.26**	0.30	0.17	0.51	0.14	–0.36	–0.30
	Text+Audio+Melody	**127.00**	0.33	0.31	**0.57**	0.80	0.41	–0.56	**–0.13**
	Text+Audio+Rhythm	144.96	**0.26**	0.31	0.17	**0.89**	0.21	**–0.08**	–0.77
	Text+Audio+Dynamics	150.90	0.29	0.29	0.17	0.61	**0.74**	–0.40	–0.59

Table 8. User study mean opinion scores (1–5) comparing text adherence (T), melody similarity (M), and overall preference (O).

Model	T	M	O
MusicGen-Stereo-Large-Melody	3.12	2.67	3.06
Stable Audio Open ControlNet	**3.69**	4.17	**3.65**
Ours (MuseControlLite)	3.58	**4.21**	3.63

same text and melody conditions as demonstrated on their site to generate music using both our model and MusicGen. These outputs are then compared to the samples retrieved from their demo page. Participants rate each audio on a 5-point Likert scale based on the following three aspects:

- **Text adherence (T)**: Does the generated audio match the text prompt?
- **Melody similarity (M)**: Does the generated audio faithfully preserve the melody from the reference audio?
- **Overall preference (O)**: Overall, how much do you like the generated audio?

The Stable Audio ControlNet (Hou et al., 2025) demo page provides eight music editing samples. We divided these into four questionnaires, each containing two melody-conditioned audio samples and three audio samples generated by our baseline methods. A total of 34 participants were recruited from our social circle and randomly assigned to one of the four questionnaires. According to subjective results, MuseControlLite demonstrates performance comparable to that of Stable Audio Open ControlNet (Hou et al., 2025) while using fewer training parameters and fewer training data.

6. Limitations

Although MuseControlLite has demonstrated superior performance in controllable music generation, there are still a few weaknesses: (i) Using multiple classifier-free guidance mechanisms for flexibility slightly slows inference due to multi-batch processing. (ii) For audio inpainting and outpainting, if the text prompt deviates significantly from the reference audio, MuseControlLite struggles to generate smooth transitions. (iii) MuseControlLite is trained solely on the public MTG-Jamendo dataset (Bogdanov et al., 2019), which contains mostly electronic music. Therefore, MuseControlLite would perform better on other genres after being fine-tuned with a more diverse dataset.

7. Conclusion

In this paper, we have introduced MuseControlLite, a lightweight training method that not only enables precise control over music generation under specified musical attribute conditions, but also supports audio outpainting and inpainting, either through text-only condition generation or with musical attribute control. We benchmark our approach against state-of-the-art ControlNet-based methods (Zhang et al., 2023; Hou et al., 2025) for structural control. MuseControlLite demonstrates superior results, suggesting that it is a powerful alternative to competing models.

Promising future directions include: (i) manipulating the attention mechanism for more efficient training and improved control precision, and (ii) enhancing control over conditions that cannot be accurately extracted using current feature extraction methods.

Impact Statement

MuseControlLite lowers the barrier for precise, time-varying control in text-to-music generation, making advanced creative tools more accessible to a broader range of artists, hobbyists, and researchers. By introducing a lightweight fine-tuning mechanism with fewer trainable parameters, we enable resource-constrained developers to integrate powerful controllability features into their systems without requiring large-scale computational infrastructures. This democratization of sophisticated AI-driven music creation can foster new waves of artistic experimentation, support rapid prototyping for commercial applications, and reduce the technical overhead that often limits innovation.

At the same time, these capabilities raise important questions about intellectual property, cultural expression, and the changing role of human creators. While MuseControlLite offers transformative advantages—such as facilitating customized soundtracks, educational tools for music learning, and expanded accessibility for individuals with limited musical training—it also highlights the need for responsible use. We encourage practitioners to adopt transparent data governance and to respect copyright laws and cultural contexts when generating music. By balancing creative freedom with ethical considerations, MuseControlLite has the potential to advance the state of music AI while emphasizing responsible and respectful innovation.

Acknowledgment

This work is supported by grants from Google Asia Pacific and the National Science and Technology Council of Taiwan (NSTC 112-2222-E002-005-MY2, NSTC 114-2124-M-002-003, NSTC 113-2628-E-002-029, NTU-113V1904-5). The authors sincerely appreciate the valuable feedback provided by the anonymous reviewers. We also thank Prof. Chris Donahue from Carnegie Mellon University for providing insights for this work. Additionally, we are grateful to the authors of Hou et al. (2025) for helping us align our training and evaluation settings.

References

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

Bogdanov, D., Won, M., Tovstogan, P., Porter, A., and Serra, X. The MTG-Jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning*, 2019.

Brooks, T., Holynski, A., and Efros, A. A. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.

Chu, Y., Xu, J., Yang, Q., Wei, H., Wei, X., Guo, Z., Leng, Y., Lv, Y., He, J., Lin, J., et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Clark, K. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Cramer, A. L., Wu, H.-H., Salamon, J., and Bello, J. P. Look, listen, and learn more: Design choices for deep audio embeddings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3852–3856. IEEE, 2019.

Davies, M. E., Degara, N., and Plumbley, M. D. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.

Defferrard, M., Benzi, K., Vandergheynst, P., and Bresson, X. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

Elizalde, B., Deshmukh, S., Al Ismail, M., and Wang, H. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Evans, Z., Carr, C., Taylor, J., Hawley, S. H., and Pons, J. Fast timing-conditioned latent audio diffusion. *arXiv preprint arXiv:2402.04825*, 2024a.

Evans, Z., Parker, J. D., Carr, C., Zukowski, Z., Taylor, J., and Pons, J. Long-form music generation with latent diffusion. *arXiv preprint arXiv:2404.10301*, 2024b.

Evans, Z., Parker, J. D., Carr, C., Zukowski, Z., Taylor, J., and Pons, J. Stable audio open. *arXiv preprint arXiv:2407.14358*, 2024c.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. Audio set: An ontology and human-labeled dataset

for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hou, S., Liu, S., Yuan, R., Xue, W., Shan, Y., Zhao, M., and Zhang, C. Editing music with melody and text: Using ControlNet for diffusion Transformer. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025.

Kim, S., Kwon, J., Wang, H., Yoo, S., Lin, Y., and Cha, J. A training-free approach for music style transfer with latent diffusion models. *arXiv preprint arXiv:2411.15913*, 2024.

Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., and Plumbley, M. D. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

Koutini, K., Schlüter, J., Eghbal-Zadeh, H., and Widmer, G. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

Krebs, F., Böck, S., and Widmer, G. An efficient state-space model for joint tempo and meter tracking. In *ISMIR*, pp. 72–78, 2015.

Kumari, N., Zhang, B., Zhang, R., Shechtman, E., and Zhu, J.-Y. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1931–1941, 2023.

Lan, Y.-H., Hsiao, W.-Y., Cheng, H.-C., and Yang, Y.-H. MusiConGen: Rhythm and chord control for Transformer-based text-to-music generation. In *ISMIR*, 2024.

Law, E., West, K., Mandel, M. I., Bay, M., and Downie, J. S. Evaluation of algorithms using games: The case of music tagging. In *ISMIR*, pp. 387–392. Citeseer, 2009.

Levy, M., Di Giorgi, B., Weers, F., Katharopoulos, A., and Nickson, T. Controllable music production with diffusion models and guidance gradients. *arXiv preprint arXiv:2311.00613*, 2023.

Li, P. P., Chen, B., Yao, Y., Wang, Y., Wang, A., and Wang, A. Jen-1: Text-guided universal music generation with omnidirectional diffusion models. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pp. 762–769. IEEE, 2024.

Lin, L., Xia, G., Jiang, J., and Zhang, Y. Content-based controls for music large language modeling. *arXiv preprint arXiv:2310.17162*, 2023.

Liu, N., Li, S., Du, Y., Torralba, A., and Tenenbaum, J. B. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pp. 423–439. Springer, 2022.

Manco, I., Weck, B., Doh, S., Won, M., Zhang, Y., Bogdanov, D., Wu, Y., Chen, K., Tovstogan, P., Benetos, E., Quinton, E., Fazekas, G., and Nam, J. The Song Describer Dataset: a corpus of audio captions for music-and-language evaluation. In *Machine Learning for Audio Workshop at NeurIPS*, 2023.

Melechovsky, J., Guo, Z., Ghosal, D., Majumder, N., Herremans, D., and Poria, S. Mustango: Toward controllable text-to-music generation. *arXiv preprint arXiv:2311.08355*, 2023.

Müller, M. *Fundamentals of music processing: Audio, analysis, algorithms, applications*, volume 5. Springer, 2015.

Novack, Z., McAuley, J., Berg-Kirkpatrick, T., and Bryan, N. DITTO-2: Distilled diffusion inference-time t-optimization for music generation. *arXiv preprint arXiv:2405.20289*, 2024a.

Novack, Z., McAuley, J., Berg-Kirkpatrick, T., and Bryan, N. J. Ditto: Diffusion inference-time t-optimization for music generation. *arXiv preprint arXiv:2401.12179*, 2024b.

Peebles, W. and Xie, S. Scalable diffusion models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Plitsis, M., Kouzelis, T., Paraskevopoulos, G., Katsouros, V., and Panagakis, Y. Investigating personalization methods in text to music generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1081–1085. IEEE, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., Ellis, D. P., and Raffel, C. C. Mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, volume 10, pp. 2014, 2014.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Rouard, S., Adi, Y., Copet, J., Roebel, A., and Défossez, A. Audio conditioning for music generation via discrete bottleneck features. *arXiv preprint arXiv:2407.12563*, 2024.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced Transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Tal, O., Ziv, A., Gat, I., Kreuk, F., and Adi, Y. Joint audio and symbolic conditioning for temporally controlled text-to-music generation. *arXiv preprint arXiv:2406.10970*, 2024.

Tsai, F.-D., Wu, S.-L., Kim, H., Chen, B.-Y., Cheng, H.-C., and Yang, Y.-H. Audio Prompt Adapter: Unleashing music editing abilities for text-to-music with lightweight finetuning. *arXiv preprint arXiv:2407.16564*, 2024.

Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Weck, B., Kirchhoff, H., Grosche, P., and Serra, X. Wikimute: A web-sourced dataset of semantic descriptions for music audio. In *International Conference on Multimedia Modeling*, pp. 42–56. Springer, 2024.

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

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

Xie, E., Chen, J., Chen, J., Cai, H., Tang, H., Lin, Y., Zhang, Z., Li, M., Zhu, L., Lu, Y., and Han, S. SANA: Efficient high-resolution image synthesis with linear diffusion Transformers. *arXiv preprint arXiv:2410.10629*, 2024.

Ye, H., Zhang, J., Liu, S., Han, X., and Yang, W. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Zhang, Y., Ikemiya, Y., Choi, W., Murata, N., Martínez-Ramírez, M. A., Lin, L., Xia, G., Liao, W.-H., Mitsufuji, Y., and Dixon, S. Instruct-musicgen: Unlocking text-to-music editing for music language models via instruction tuning. *arXiv preprint arXiv:2405.18386*, 2024.

Zhao, S., Chen, D., Chen, Y.-C., Bao, J., Hao, S., Yuan, L., and Wong, K.-Y. K. Uni-ControlNet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

A. Separated guidance scale formulation

To expand the classifier-free guidance from a single condition to a general form, we start from:

$$p(x, c_1, \ldots, c_n) = p(x) \prod_{i=1}^{n} p\big(c_i|x,\, c_1, \ldots, c_{i-1}\big)\,. \tag{11}$$

We simply apply the Bayes' rule:

$$p(x|c_1, \ldots, c_n) = \frac{p(x) \prod_{i=1}^{n} p\big(c_i|x,\, c_1, \ldots, c_{i-1}\big)}{p(c_1, \ldots, c_n)}\,. \tag{12}$$

Use log scale to convert multiplications to additions:

$$\log p(x|c_1, \ldots, c_n) = \log(x) + \sum_{i=1}^{n} \log p\big(c_i|x,\, c_1, \ldots, c_{i-1}\big) - \log p(c_1, \ldots, c_n)\,. \tag{13}$$

Take the derivative to eliminate the constant term:

$$\nabla_x \log p(x|c_1, \ldots, c_n) = \nabla_x \log p(x) + \sum_{i=1}^{n} \nabla_x \log p\big(c_i|x,\, c_1, \ldots, c_{i-1}\big)\,, \tag{14}$$

$$\sum_{i=1}^{n} \nabla_x \log p\big(c_i|x,\, c_1, \ldots, c_{i-1}\big) = \sum_{i=1}^{n} \nabla_x \log p(\big(c_i, x,\, c_1, \ldots, c_{i-1}\big) - \log p\big(x,\, c_1, \ldots, c_{i-1}\big)\big)\,. \tag{15}$$

Then we scale the condition term with guidance λ_i. The guidance λ_i can be different according to the control strength that is required.

$$\nabla_x \log p(x|c_1, \ldots, c_n) = \nabla_x \log p(x) + \sum_{i=1}^{n} \lambda_i \nabla_x \log p(\big(c_i, x,\, c_1, \ldots, c_{i-1}\big) - \log p\big(x,\, c_1, \ldots, c_{i-1}\big)\big)\,, \tag{16}$$

leading to Equation (9) shown in Section 3.5.

B. Ablation study for all key module in MuseControlLite

We conducted an ablation study on the key modules of our proposed method. Without rotary positional embeddings (RoPE) (Su et al., 2024), the model cannot perform melody control, although it still achieves the highest CLAP (Elizalde et al., 2023) score. Removing the 1D-CNN condition extractor causes all metrics to drop, and omitting the zero-initialized 1D-CNN layers used to sum cross-attention outputs similarly degrades performance. Finally, doubling the number of attention heads by scaling W'^k and W'^v yields no improvement compared to the base configuration.

Table 9. Performance of all combinations of controls using conditions extracted from Song Describer Dataset (Manco et al., 2023).

	Extractor	**ROPE**	**Scale up**	**CNN**	**FD** ↓	**KL** ↓	**CLAP** ↑	**Mel acc.** ↑
Ours	✓	✓	✗	✓	**78.50**	**0.29**	0.38	**58.6%**
Ours w/o ROPE	✓	✗	✗	✓	113.13	0.57	**0.41**	10.7%
Ours w/o Extractor	✗	✓	✗	✓	94.50	0.33	0.38	57.2%
Ours w/o CNN	✓	✓	✗	✗	93.30	0.30	0.37	56.7%
Ours w/ double heads	✓	✓	✓	✓	80.25	**0.29**	0.38	58.4%

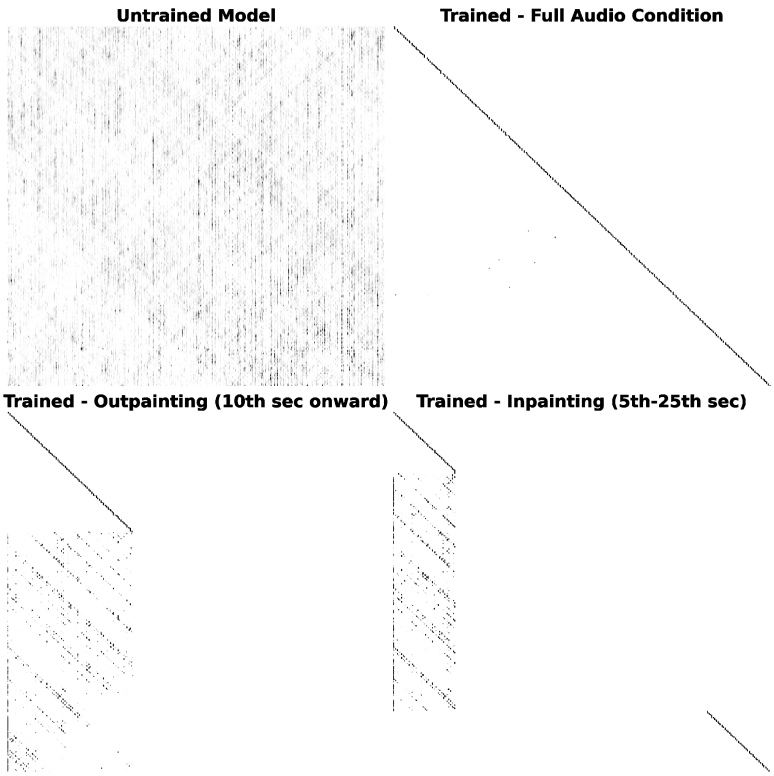

Figure 2. This figure consists of four attention maps: i) untrained, ii) trained, given full audio condition, iii) trained, inpainting 5th-25th seconds, and iv) trained, outpainting 10th seconds onward. After training, a portion of the attention maps exhibits a perfectly diagonal pattern when given the full audio condition. When performing inpainting or outpainting (i.e., only partial audio condition is given), the model tends to reference previous keys, exhibiting effective usage of given audio context.