MuseControlLite: Multifunctional Music Generation with Lightweight Conditioners

Anonymous Authors[1]

Abstract

We propose MuseControlLite, a lightweight mechanism with only 85M trainable parameters to fine-tune text-to-music generation models to gain precise controllability over various time-varying conditions of musical attributes and from reference audio signals. The key finding is that positional embeddings, which have been seldom used in the conditioner of text-to-music generation models for text conditions, are critical when the condition of interest is a function of time. Taking melody control as an example, our experiment shows that the simple idea of adding rotary positional embeddings to the so-called decoupled cross-attention layers greatly improves the control accuracy from 56.6% to 70.9%, with 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 provide evaluation of various musical attribute controls, audio inpainting, and audio outpainting, demonstrating improved controllability over Music ControlNet and Stable Audio Open ControlNet with lower cost of fine-tuning. We will open source the code and model checkpoint, along with demo examples which can be found 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 can 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) offer multiple conditions of melody, rhythm and dynamics. Stable Audio Open ControlNet (Hou et al., 2024) 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 the generation results are influenced by the provided condition, the control accuracy (i.e., how well the generated pieces follow the control signal) is insufficient. For example, many studies have attempted to exert controllability over the melody part of the generated music, but the melody control accuracy reported in the literature has been around 50% only (Copet et al., 2024; Hou et al., 2024) (see Table 3). For a better user experience, a model with higher control accuracy might be needed.

We propose MuseControlLite, a lightweight fine-tuning mechanism with fewer trainable parameters yet empirically more accurate control of time-varying conditions than existing approaches, for controllable text-to-music generation. The key finding is that the decoupled cross-attention mechanism (Ye et al., 2023) can use nearly an order magnitude

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

fewer trainable parameters than ControlNet-based mechanisms (572M vs 85M), but it needs *positional embeddings* to work well. Interestingly, our pilot study shows that decoupled cross-attention alone yields very inaccurate control, possibly because the trainable parameters are too few. While positional embeddings are not used in the conditioners of ControlNet-based models (Wu et al., 2024; Hou et al., 2024), it turns out that they are critical inductive bias when the trainable parameters are few, likely because the content to be generated (i.e., music signals) and the conditions of interest (e.g., melody) here are both functions of time.

Based on the simple idea of combining positional embeddings and decoupled cross-attention, MuseControlLite is a novel framework for time-varying condition control. Specifically, we employ rotary positional embedding (ROPE) (Su et al., 2024), incorporating a modification where we rotate all queries, keys, and values to enhance position-aware attention. Our implementation uses only 8% trainable parameters of a diffusion Transformer backbone (Evans et al., 2024c) to learn additional controls, while maintaining the inference speed of the pretrained model, which does not offer fine-grained controllability.

Our model supports multi-attribute control similarly to Music ControlNet (Wu et al., 2024). Moreover, treating the reference audio signals as another type of time-varying control signals, we also train a separate set of adapters for audio conditioning, enabling audio inpainting and outpainting applications. The two adapter sets are fully compatible and can be combined with the original text condition. Additionally, we adopt multiple classifier-free guidance (Liu et al., 2022; Brooks et al., 2023) to flexibly regulate the strength of each condition, preventing the quality degradation that can result from over-fixation on any single condition.

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 trainable model that handles both attribute and audio control ('text+attribute+audio'). In contrast, existing trainable models only take either attribute control ('text+attribute') (Wu et al., 2024), or audio control ('text+audio') (Rouard et al., 2024; 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., 2024) in melody control, achieving a 14% improvement in melody accuracy.

We will share code and model checkpoint upon publication.

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 for adding new conditions. MusiconGen (Lan et al., 2024) and Coconulla (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 pretrained 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

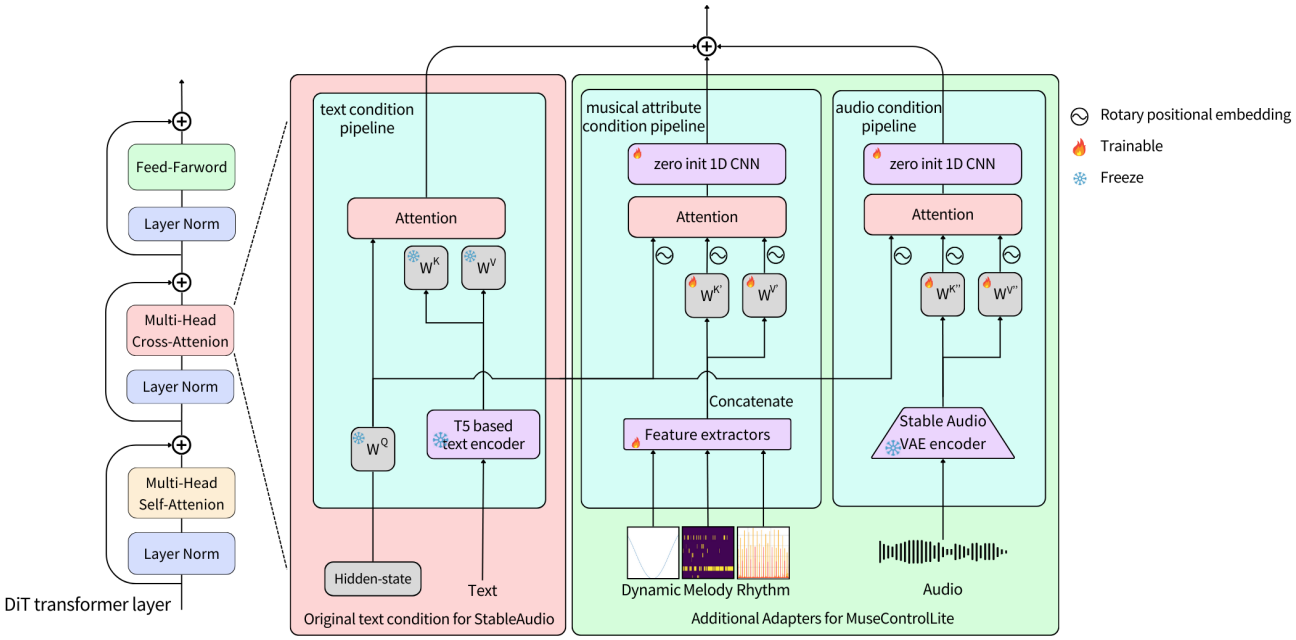

Figure 1. MuseControlLite incorporates two trainable pipelines to handle musical attribute conditions and audio conditions. The musical attribute pipeline offers control over elements such as melody, rhythm, and dynamics, whereas the audio condition pipeline facilitates audio inpainting and outpainting. Although the pipelines operate independently, they can integrate and cooperate seamlessly when needed.

$\mathbf{x}_1, \ldots, \mathbf{x}_T$), 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 Transformer (Peebles & Xie, 2023) emerges 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 $\boldsymbol{\epsilon}$, $\mathbf{x}_0$ 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 take 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:

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

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

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 $\mathbf{R}_{\Theta,i}^d$ 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 handing the text condition (i.e., the part with pink shed in the middle of Figure 1), we leave it unchanged:

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

where $\mathbf{c}_{\text{text}}$ denotes the embedding representing the text condition. Transferring the technique from the traditional U-Net, we adapt it 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 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 not only to the query and key vectors but also to the value vectors to enhance the model's positional awareness. We then train the duplicated $\mathbf{W}'^k$ and $\mathbf{W}'^v$ to handle $\mathbf{c}_{\text{attr}}$:

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

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

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

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

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

where $\mathbf{Q}_{\text{rope}} = \{\mathbf{q}_m\}_{m=1}^{M}$, $\mathbf{K}_{\text{rope}} = \{\mathbf{k}_m\}_{m=1}^{M}$, $\mathbf{V}_{\text{rope}} = \{\mathbf{v}_m\}_{m=1}^{M}$, 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:

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

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.

Table 1. We trained our models for 27,000 steps with the melody condition and found that, without ROPE, they struggle to learn the new condition and exhibit poorer audio realism.

	FD↓	KL↓	CLAP↑	Mel acc.↑
w/o ROPE	245.53	0.58	0.34	13.8%
w/ ROPE	**135.46**	**0.37**	**0.40**	**70.9%**

Musical Attribute Control We follow Music ControlNet (Wu et al., 2024) to extract melody, rhythm, and dynamics. For melody ($\mathbf{c}_{\text{mel}} \in \mathbb{R}^{N_{\text{melody}} \times 12}$), we compute a chromagram using Librosa (McFee et al., 2015), convert it to one-hot via `argmax`, and apply a high-pass filter (cutoff at 261.2 Hz) to suppress bass. Dynamics ($\mathbf{c}_{\text{dyn}} \in \mathbb{R}^{N_{\text{dynamics}} \times 1}$) are derived from the spectrogram energy, mapped to decibels, and smoothed with a Savitzky-Golay filter to align with perceived intensity. Rhythm ($\mathbf{c}_{\text{rhy}} \in \mathbb{R}^{N_{\text{rhythm}} \times 2}$) (Böck et al., 2016) uses a recurrent network-based beat detector for beat/downbeat probabilities, enabling precise synchronization and rhythmic nuance. We use separate 1D CNN layers to extract features from each condition and expand their channel sizes to $C_r/3$, where C_r is the cross-attention dimension. We then use PyTorch's `interpolate` functions to match the sequence length among these features and concatenate them in the last dimension, resulting in the condition $\mathbf{c}_{\text{all}} \in \mathbb{R}^{N_{\text{interpolate}} \times C_r}$ for the decoupled cross-attention input, where $N_{\text{interpolate}}$ is the same as the Stable Audio's latent sequence length.

During training, we apply a masking strategy that followed 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 20–30-second segments blank for the model to improvise.

Audio Inpainting and Outpainting Since audio provides far more information than the musical attribute conditions, we found that training with both audio and musical attribute conditions simultaneously can cause the model to ignore the musical attribute conditions. Thus, we train another pair of $\mathbf{W}''^v$ and $\mathbf{W}''^k$ (i.e., another learnable copy of $\mathbf{W}^k$ and $\mathbf{W}^v$) solely for the audio condition $\mathbf{c}_{\text{audio}}$ using the same masking strategy. We directly use the VAE-encoded latent ($\mathbf{x}_0 \in \mathbb{R}^{A \times T_{\text{audio}}}$), referred to as the "clean latent" for StableAudio, as the audio condition, where T_{audio} is the length of the encoded audio and A is the number of audio latent channels.

By applying masks to $\mathbf{c}_{\text{audio}}$ during training, $\mathbf{W}''^k$ and $\mathbf{W}''^v$

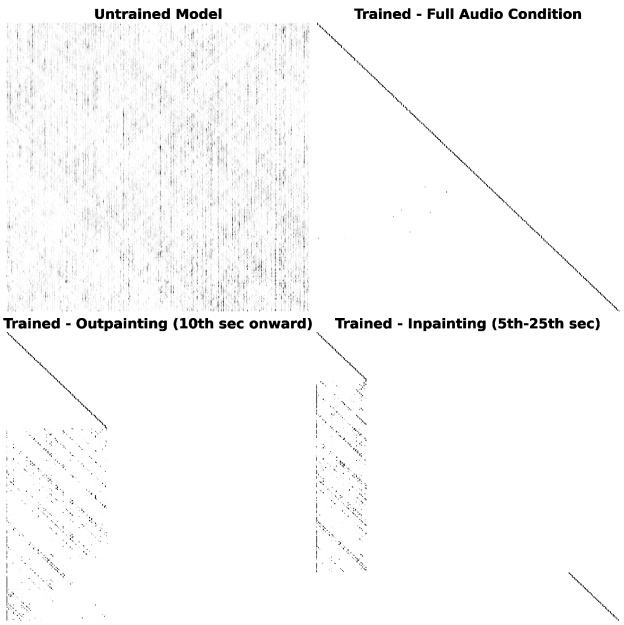

Figure 2. This figure consists of four attention maps: i) untrained, ii) trained, given full audio condition, iii)trained, inpainting 5th-25th seconds, and iv) trained, outpainting 10th seconds onward. After training, the attention map exhibits a perfectly diagonal pattern when given the full audio condition. 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.

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. 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 MuseControlLite, we utilize several pipelines to model $p_\theta(\mathbf{x}|\mathbf{c}_{\text{attr}}, \mathbf{c}_{\text{audio}})$ during training. Our pilot studies finds that after fine-tuning, the model often over-fits to the additional conditions $\mathbf{c}_{\text{attr}}$ or $\mathbf{c}_{\text{audio}}$. Therefore, we adopt separate guidance scales (Brooks et al., 2023) λ_{attr} and λ_{audio}, yielding

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

Detailed formulation can be found in Appendix B.

4. Experimental setup

4.1. Dataset

We used an internal training dataset (which will be shared publicly for reproducibility upon paper publication) consisting of 30-second audio clips with LLM-generated captions derived from the MTG-Jamendo dataset (Bogdanov et al., 2019), while filtering out those included in the Song Describer Dataset (Manco et al., 2023), totaling 3,500 hours worth of data. For evaluation, we follow the methodology of the Stable Audio papers (Evans et al., 2024a;c;b), utilizing the Song Describer Dataset but restricting it to instrumental music (i.e., excluding samples with singing voice prompts), resulting in a total of 589 audio clips. We use all 589 audio clips in Section 5.1, while for other tasks in Section 5.2 and Section 5.3, we use a smaller test set with 50 audio randomly sampled from this collection.

4.2. Training and Inference Specifics

We fine-tuned the model using the following objective:

$$\mathbb{E}_{t\sim[0,1],x_t} \left\| f_\theta\left(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}, t\right) - \mathbf{v_t} \right\|_2^2 , \tag{9}$$

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). The musical attribute pipeline and the audio condition pipeline are trained separately. We used a constant learning rate of 10^{-4}, a weight decay of

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	2.5	0.5
Audio outpainting	7.0	1.5	1.0

10^{-2}, and dropped the text condition for 50% of training iterations to encourage the model to focus on $\mathbf{c}_{\text{attr}}$ or $\mathbf{c}_{\text{audio}}$. Additionally, the other conditions were independently dropped with a probability of 5%. Training was conducted for 27,000 steps with an effective batch size of 64 on a single NVIDIA RTX 3090, using the same settings for both pipelines.

The model generates 30-second audio samples, aligning with the duration of the training dataset. To accommodate this, we adjusted the pretrained Stable Audio Open (Evans et al., 2024c) to operate within a shorter latent space corresponding to 30 seconds, without further fine-tuning, as we found that latent space length has only a minor influence on audio quality.

For inference, we fixed the separate guidance scales as shown in Table 2. To ensure a fair comparison with Stable Audio Open ControlNet (Hou et al., 2024), we used 250 denoising steps for the evaluations reported in Section 5.1. For other tasks, we used 50 denoising steps.

4.3. 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., 2024), we use the same open-source code[1] for calculating $\text{FD}_{\text{openl3}}$, KL_{passt}(Koutini et al., 2021), and $\text{CLAP}_{\text{score}}$(Wu et al., 2023). $\text{FD}_{\text{openl3}}$ 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, adapting it for long-form audio by segmenting and averaging logits. CLAPscore assesses text-audio correspondence using LAION CLAP embeddings with a feature fusion approach, ensuring robust evaluation of long-form, high-resolution audio.

Following (Wu et al., 2024), we use the metrics below to evaluate musical attribute controllability.

Melody Accuracy Melody accuracy measures the agreement between the frame-wise pitch classes (C, C#, ..., B; 12 in total) of the input melody control and those extracted from the generated output. A higher accuracy indicates better preservation of the intended melody.

Dynamics Correlation 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 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.

Novelty 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. Finally, the convolution values are summed for each time position, forming a one-dimensional curve known as the novelty curve N. This curve represents abrupt changes in musical content over time, with peaks typically indicating segment boundaries.

We define the Novelty Value as $N_i - N_{i-1} - N_{i+1}$, where i represents the boundary position. A lower Novelty Value indicates smoother transitions, implying the absence of abrupt changes in the surrounding time frames.

4.4. Baselines

Although musical attribute control, audio inpainting, and outpainting have been done in many prior works, they are either not open source (Wu et al., 2024; Novack et al., 2024b;a; Li et al., 2024) or not generating a relatively short audio (Tal et al., 2024). MusiConGen (Lan et al., 2024) and Coco-Mulla provides (Lin et al., 2023) rhythm control, but MusiconGen uses a constant bpm to represent rhythm, and CoCo-mulla uses MIDI-like conditions which is far from our us, thus we consider not suitable for comparison.

- **MusicGen** (Copet et al., 2024): MusicGen is a

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

Model	Trainable Parameters	Total Parameters	FD ↓	KL ↓	CLAP ↑	Mel acc. ↑
MusicGen-Stereo-Large-Melody	3.3B	3.3B	187.022	0.471	0.362	43.7%
Stable Audio Open ControlNet	572M	1.9B	**97.734**	**0.265**	0.396	56.6%
Ours (MuseControlLite)	**85M**	**1.4B**	135.458	0.376	**0.397**	**70.9%**

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 nearly an order magnitude fewer trainable parameters.

	melody	rhythm	dynamics	FD ↓	KL ↓	CLAP ↑	Mel acc. ↑	Rhy F1 ↑	Dyn cor. ↑
No condition	✗	✗	✗	226.06	0.61	0.36	0.10	0.21	0.17
Single condition	✓	✗	✗	152.30	0.40	0.37	**0.70**	0.47	0.31
	✗	✓	✗	171.63	0.51	0.34	0.09	0.86	0.48
	✗	✗	✓	219.53	0.61	0.32	0.10	0.52	0.92
Double conditions	✓	✓	✗	**124.73**	0.32	0.38	**0.70**	0.86	0.54
	✓	✗	✓	144.45	0.31	0.39	**0.70**	0.69	0.94
	✗	✓	✓	187.93	0.53	0.32	0.10	**0.87**	0.94
All conditions	✓	✓	✓	134.53	**0.28**	**0.39**	**0.70**	**0.87**	**0.95**

Table 4. Performance of all combinations of controls using conditions extracted from Song Describer Dataset (Manco et al., 2023).

transformer-based auto-regressive 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., 2024): 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., 2024), 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 we implemented 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.

5. Result

5.1. Melody Comparison with ControlNet

To compare with Stable Audio Open ControlNet (Hou et al., 2024) and MusicGen-Melody (Copet et al., 2024), we trained another set of adapters that learn only melody conditions, keeping all other settings the same as in Section 4.2. The results are shown in Table 3. With fewer trainable parameters, MuseControlLite achieves intermediate performance in **FD** and **KL** between the two baselines. However, it outperforms both Stable Audio Open ControlNet and MusicGen-Melody in melody accuracy by 14% and 27%, respectively, demonstrating the precise controllability of our proposed model.

5.2. Ablation Study for Musical Attribute Conditions

We evaluate all combinations of controls for dynamics, melody, and rhythm. Our findings indicate that after training with musical-attribute conditions, unconditional generation results in the worst audio-realistic (**FD**), which is consistent with observations in Music ControlNet (Wu et al., 2024).

For single-condition controls, performance improves when the corresponding condition is provided. Additionally, we observed that the conditions are not independent. When the melody condition is given, the dynamics correlation and rhythm F1 score also increase. A similar phenomenon is observed between rhythm and dynamics.

For multiple controls, the controllability metrics remain largely the same compared to single-control scenarios while achieving a lower **FD** and **KL** score. This indicates improved audio realism and better semantic alignment with the reference dataset. These results suggest that the model effectively learns to respond to multiple controls simultaneously, despite the added complexity.

	Model	FD ↓	KL ↓	CLAP ↑	Mel acc. ↑	Rhy F1 ↑	Dyn cor. ↑	Novelty value (10s) ↓
Baseline	MusicGen-Stereo-Large	248.99	0.84	0.23	0.34	0.70	0.48	–1.66
	Naïve masking	271.95	1.40	0.11	0.30	0.46	0.46	–1.16
Ours	Unconditional	199.93	0.56	0.29	0.37	0.63	0.59	**–2.10**
	Melody control	**193.60**	**0.29**	0.35	**0.59**	0.60	–0.08	–1.76
	Rhythm control	205.33	0.30	0.35	0.13	**0.87**	0.08	–1.77
	Dynamics control	198.04	0.37	**0.36**	0.12	0.45	**0.84**	–1.42

Table 5. Result for audio outpainting task, our model outperforms the baselines in all aspects.

		FD ↓	KL ↓	CLAP ↑	Mel acc. ↑	Rhy F1 ↑	Dyn cor. ↑	Novelty value (5s) ↓	Novelty value (25s) ↓
Baseline	Naïve masking	701.88	1.75	0.05	0.30	0.48	0.40	–1.17	–1.24
Ours	Unconditional	188.62	0.59	0.30	**0.39**	0.65	0.66	–1.38	–1.48
	Melody control	**188.53**	0.46	0.36	0.34	0.59	0.05	**–1.72**	**–1.81**
	Rhythm control	191.66	**0.42**	**0.36**	0.05	**0.88**	0.13	–1.20	–1.24
	Dynamics control	224.55	0.56	0.32	0.05	0.27	**0.75**	–1.33	–1.48

Table 6. Results for audio inpainting task.

5.3. Audio Outpainting and Inpainting

Audio Outpainting We mask a 30-second audio clip, retaining only the first 10 seconds, and experiment with our model using both unconditional generation and single musical attribute control. To ensure that evaluation metrics are not influenced by the reference audio, we trim out the first 10 seconds before computing them. Musical attribute evaluations are based on the trimmed audio. As shown in Table 5, our unconditional outpainting outperforms MusicGen-Large in all aspects except for Rhythm F1, demonstrating superior audio realism and consistency. Notably, our non-autoregressive model achieves better 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.

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 evaluation and assess only the generated inpainted part. The results, presented in Table 6, 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 complex.

6. Conclusion

In this paper, we have introduced MuseControlLite, a lightweight training method that not only enables precise control of music generation under specified musical attribute conditions but also supports audio outpainting and inpainting, either through unconditional generation or with musical attribute control. We benchmark our approach against the state-of-the-art ControlNet-based approaches (Zhang et al., 2023; Hou et al., 2024) 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. ii) Enhancing control over conditions that can not 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.

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 the 24th 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 (ICML 2019)*, Long Beach, CA, United States, 2019. URL http://hdl.handle.net/10230/42015.

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.

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.

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.

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.

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. *arXiv preprint arXiv:2410.05151*, 2024.

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.

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. *arXiv preprint arXiv:2407.15060*, 2024.

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*, 2023.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. librosa: Audio and music signal analysis in python. In *SciPy*, pp. 18–24, 2015.

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.

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 *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

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:

$$p(x, c_1, \ldots, c_n) = p(x) \prod_{i=1}^{n} p\big(c_i | x,\, c_1, \ldots, c_{i-1}\big)\,. \tag{10}$$

We simply apply the Bayes' rule:

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

Use log scale to convert multiplications to additions:

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

Take the derivative to eliminate the constant term:

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

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

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.

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

leading to the equation shown in Section 3.5.

B. Results for using different guidance scale formulation

We have tested the musical attribute pipeline with different λ_{attr} and found out that there is a trade-off between the original model's ability and the additional control. The line where $\lambda_{\text{attr}} = 0$, is using the pretrained Stable Audio Open for 30-second audio generation. As the λ_{attr} goes larger, **FD** also rises, but **KL** and **CLAP** seems to have the optimal between $\lambda_{\text{attr}} = 1.5$ and $\lambda_{\text{attr}} = 2.0$. We pick $\lambda_{\text{attr}} = 2.0$ for the evaluation in Section 5.2.

λ_{attr}	**FD** ↓	**KL** ↓	**CLAP** ↑	**Mel acc.** ↑	**Rhy f1** ↑	**Dyn cor.** ↑
0.0	**113.73**	0.51	0.39	0.10	0.22	0.16
1.5	128.29	**0.28**	**0.40**	0.66	**0.87**	0.94
2.0	134.53	**0.28**	0.39	0.70	**0.87**	**0.95**
2.5	146.07	0.31	0.35	**0.72**	**0.87**	**0.95**
3.0	157.65	0.34	0.36	**0.72**	**0.87**	**0.95**

Table 7. Results for λ_{attr} comparison