SongCreator: Lyrics-based Universal Song Generation

Shun Lei[1], **Yixuan Zhou**[1†], **Boshi Tang**[1], **Max W. Y. Lam**[2], **Feng Liu**[2], **Hangyu Liu**[2], **Jingcheng Wu**[2], **Shiyin Kang**[2], **Zhiyong Wu**[1,3*], **Helen Meng**[3]
[1] Shenzhen International Graduate School, Tsinghua University, Shenzhen
[2] Independent Researcher
[3] The Chinese University of Hong Kong, Hong Kong SAR
`{leis21, yx-zhou23}@mails.tsinghua.edu.cn, zywu@sz.tsinghua.edu.cn`

Abstract

Music is an integral part of human culture, embodying human intelligence and creativity, of which songs compose an essential part. While various aspects of song generation have been explored by previous works, such as singing voice, vocal composition and instrumental arrangement, etc., generating songs with both vocals and accompaniment given lyrics remains a significant challenge, hindering the application of music generation models in the real world. In this light, we propose SongCreator, a song-generation system designed to tackle this challenge. The model features two novel designs: a meticulously designed dual-sequence language model (DSLM) to capture the information of vocals and accompaniment for song generation, and a series of attention mask strategies for DSLM, which allows our model to understand, generate and edit songs, making it suitable for various song-related generation tasks by utilizing specific attention masks. Extensive experiments demonstrate the effectiveness of SongCreator by achieving state-of-the-art or competitive performances on all eight tasks. Notably, it surpasses previous works by a large margin in lyrics-to-song and lyrics-to-vocals. Additionally, it is able to independently control the acoustic conditions of the vocals and accompaniment in the generated song through different audio prompts, exhibiting its potential applicability. Our samples are available at https://thuhcsi.github.io/SongCreator/.

1 Introduction

Music is an integral part of human culture, embodying human intelligence and creativity. Songs combining vocals and accompaniment compose an essential part of it, whose generation has been a hotspot in both academia and industry in recent years. Although with the rapid advancements in generative models, communities have witnessed the applications of Artificial Intelligence Generated Content (AIGC) models in the generation of texts [1–3], images [4–6] and speeches [7–11], it still remains a big question whether we can replicate the successes in song generation, which demands coordination among various complex elements such as instruments, rhythm, melody and vocals. Currently creating high-level songs with both vocals and accompaniment still requires substantial human effort in composition, instrument arrangement, singing, and so on, a process requiring a great deal of time and expertise. Lyrics-to-song generative models could lower the barrier to entry for novices and improve the workflow of experienced artists.

Previous works mostly explored specific aspects of song generation, as listed in Table 1. Although they exhibit abilities in vocal composition, instrumental arrangement and harmonious generation, none of them is able to combine these three for high-quality lyrics-to-song generation. To this end, Jukebox [12] can be seen as the first and only attempt from published literature so far to

[*]Corresponding author, [†] Equal contribution.

Table 1: A comparison of song generation with related tasks in the literature. We use **Composition** to denote whether the model can complete vocal composition, **Arrangement** to denote whether the model can arrange the instrumental accompaniment, and **Harmony** to denote whether vocals and accompaniment sound harmonious and pleasant together.

Tasks	**Inputs**	**Outputs**	**Composition**	**Arrangement**	**Harmony**
Singing Voice Synthesis [15–20]	Scores	Vocals	✗	✗	✗
SongComposer [21]	Lyrics	Vocals	✓	✗	✗
Text-to-Music [22–25]	Text Description	Music	✗	✓	✗
Accompaniment Generation [26–30]	Vocals	Music	✗	✓	✓
Song Generation	Lyrics	Song	✓	✓	✓

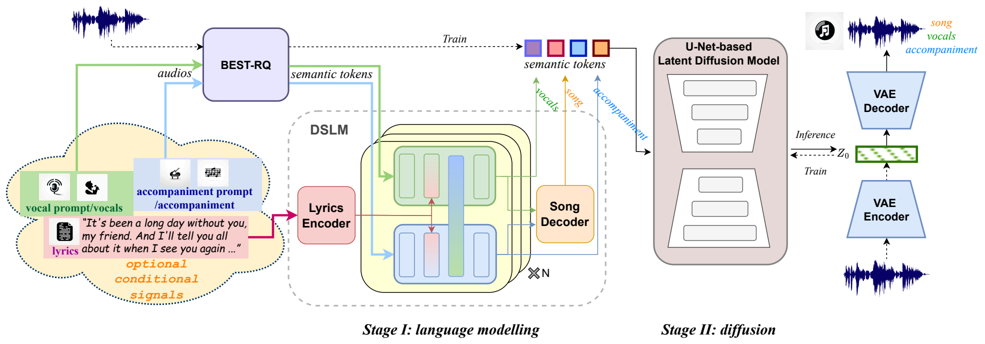
simultaneously generate vocals and accompaniment in a song from lyrics using a single model. However, it exhibits two major limitations. Firstly, this approach treats the combination of vocals and accompaniment as an entity. While the design facilitates the generation of songs, it ignores the mutual influence between vocals and accompaniment, resulting in vocals that sound unnatural and a lack of musicality in both the melody and accompaniment, and inhibiting the independent controllability of the generated vocals and accompaniment. Secondly, it is confined to performing specific tasks of lyrics-to-song generation, which restricts the broader application of song generation models in complex musical scenarios, including the generation of vocals or instrumental music, as well as universal song generation tasks such as song editing and accompaniment-to-vocal generation. Recently, while the industry has seen the emergence of song generation tools like Suno [13] and Udio [14], neither has disclosed their methodologies nor has expanded into universal song generation tasks.

In this work, we introduce SongCreator, a system designed to generate high-quality songs with harmoniously coordinated vocals and accompaniment based on lyrics. It is worth mentioning that by learning composition and arrangement abilities during training, SongCreator can also be applied to universal song generation tasks, as shown in Appendix B, including (but not limited to) lyrics-to-vocal, accompaniment-to-song and song editing. Formalized as a combination of a language model (LM) and a latent diffusion model (LDM) [31], SongCreator features a novel dual-sequence language model (DSLM), which utilizes two decoders to separately model vocals and accompaniment information, and employs a dynamic bidirectional cross-attention module to capture the influences between these two sequences. This approach treats vocals and accompaniment within a song as separate but interrelated sequences, effectively reducing their mutual influence during training. Additionally, inspired by UniLM [32] and GLM [33], we design a series of attention mask strategies for DSLM, which enables SongCreator to complete song generation tasks of various forms, such as editing, understanding and generation in a unified manner. Our contributions can be summarized as follows:

- We propose a novel dual-sequence language model for song generation. Compared to previous ones, it not only emphasizes the respective quality of vocals and accompaniment, but also learns their mutual influences to coordinate them into harmonious songs, greatly enhancing the quality of generations.
- We propose a series of attention mask strategies for song generation, which endows our model with the ability to unify song generation tasks of various forms, such as lyrics-to-song, accompaniment-to-song and song editing. It also makes multi-task training feasible for SongCreator, which underlies its versatile generation ability.
- On top of the mechanisms above, we propose a versatile system for song generation. It can be readily applied to lyrics-based vocals/song generation, or even editing. It also supports universal conditioning and generation: given any one of the vocals or accompaniment as a condition, SongCreator is able to generate the other. Moreover, SongCreator is able to generate songs with separate audio prompts for vocals and accompaniment.
- We conduct extensive experiments to demonstrate the abilities of our system in the eight tasks shown in Appendix B. Ablation experiments justify the effectiveness of our designs.

2 Related Work

Singing voice synthesis Singing Voice Synthesis (SVS) [15–20] aims at synthesize vocals given scores, has made great progress in recent years. Several works attempt to adopt transformer models



Train
audios
BEST-RQ
semantic tokens
semantic tokens
vocals
song
accompaniment
DSLM
vocal prompt/vocals
accompaniment prompt /accompaniment
lyrics
"It's been a long day without you, my friend. And I'll tell you all about it when I see you again ..."
optional conditional signals
Lyrics Encoder
Song Decoder
N
U-Net-based Latent Diffusion Model
Inference
Train
z_0
VAE Decoder
VAE Encoder
song
vocals
accompaniment
Stage I: language modelling
Stage II: diffusion

Figure 1: The overview of SongCreator. The BEST-RQ tokens is a proxy that bridges the DSLM and the latent diffusion model.

[15], generative adversarial networks [16] and conditional variational autoencoder [17, 18] for SVS. Recently, research [19, 20] focuses on enhancing the quality of synthesized vocals through diffusion models, demonstrating state-of-the-art (SOTA) performance. Similarly, SongCreator also employs a diffusion model to improve the quality of synthesized songs. However, compared to traditional SVS methods that require additional scores composed by humans, SongCreator facilitates composing and arranging songs directly from lyrics and generates complete songs with accompaniment.

Music generation Music generation has been long studied under various setups. Early efforts [34, 35] primarily focus on generating symbolic music, which is confined to fixed instrumental timbres and lacks expressiveness. Several works [12, 22, 25, 36] have achieved text-to-music generation by tokenizing music into discrete sequences that can be further processed by language models (LMs) [2, 3]. Singsong [29] and Melodist [30] follow a similar approach for accompaniment generation. Diffusion models [37–39], as another competitive class of generative models, have also delivered impressive results in music generation. Many emerging methods [24, 40–42] use latent diffusion model (LDM) to generate high-quality and high-fidelity music. Recently, MeLoDy [23] and AudioLDM 2 [43] introduce a novel solution by combining the advantages of LMs and LDM, demonstrating SOTA performances with high fidelity and musicality. However, these methods are designed for generating non-vocal music and limited to specific task such as text-to-music or vocal-to-accompaniment. By leveraging the DSLM, SongCreator can effectively model songs that include both vocals and accompaniment, and it can also extend to various song generation tasks.

Speech editing and synthesis Speech editing requires to alter a segment within a speech to match the target transcript. Early methods [44–48] utilized the surrounding speech context as a condition, enabling models to generate the masked segment. Subsequently, several works [11, 49–51] attempted to establish a unified model for both speech editing and text-to-speech (TTS). However, despite their impressive achievements, these efforts are restricted to handing clean signals only, and required duration information for each phoneme. It restricts the applicability of such methods in song or vocal editing. Recently, the advancement of LMs significantly promoted progress in speech generation, particularly in zero-shot TTS [9, 52, 53] and speech editing [54–56]. Different from these works that only focus on speech, to our knowledge, we are the first to implement song and vocal editing. Additionally, through a serious of attention mask strategies, our proposed SongCreator provides a general solution that enables a single system to handle multiple tasks in song generation.

3 Method

3.1 Overview

Let $\mathbf{x} \in \mathcal{X}$ represent a song audio. A song generation process can be defined as $f : \mathcal{C} \mapsto \mathcal{X}$, where $\mathcal{C}$ is the set of conditioning signals. In this work, we consider a flexibly conditioned song generation system f with $\mathbf{C} \in \mathcal{C}$, accepting a variety of optional inputs including lyrics, vocal prompt, accompaniment prompt, pre-determined vocal track and pre-determined accompaniment track. The

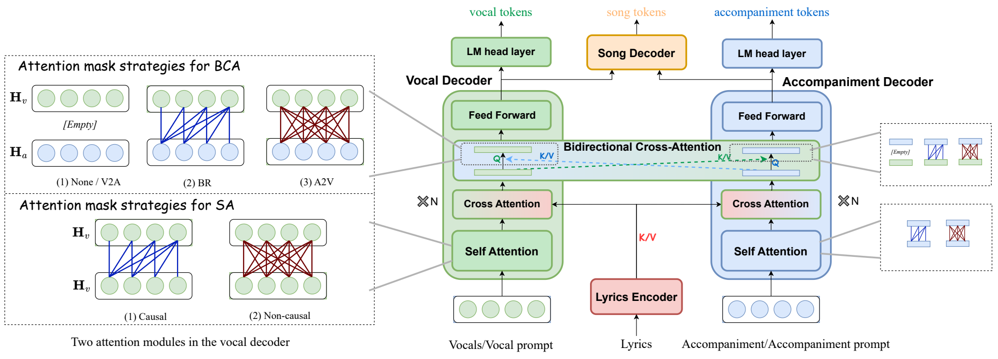


Figure 2: The overview of DSLM with the attention mask strategies. The DSLM can utilize specific attention mask strategy to achieve different song generation tasks. We illustrate multiple attention mask strategies of what each vocal token's representation attend to in both self-attention and bidirectional cross-attention. Attention mask strategies in the accompaniment decoder are similar.

high flexibility of conditions empowers the controllability of our model, so that different elements within the generated songs can be customized as needed.

However, end-to-end generating a high-fidelity song $\mathbf{x}$ from $\mathbf{C}$ with a neural network f remains challenging to date. In the same spirit as previous works [23, 25, 36], we introduce a language-alike discrete sequence (a.k.a., *semantic tokens*), denoted as $\mathbf{S} = (S_1, \ldots, S_N)$, to capture significant structural information in song and to embody LMs as the "brain" of our system for writing songs.

As illustrated in Figure 1, to obtain the semantic tokens, we train a BEST-RQ [57] on an unlabeled dataset containing songs, vocals and music, and conduct vector quantization over its intermediate hidden representations. These tokens encapsulate sufficient semantic and acoustic details that are necessary for reconstructing $\mathbf{x}$. With such a purpose, an LDM, consisting of a VAE and a diffusion model, is trained to decode the semantic tokens into high-quality song audio, in a way similar to [23]. Since both BEST-RQ and LDM were trained and re-produced with open-source implementations, their respective details are beyond the focus of this paper, and are described in Appendix A.2-A.3.

To predict the semantic tokens $\mathbf{S}$ given $\mathbf{C}$, we designed a novel form of LM – dual-sequence language model (DSLM) for multi-condition song generation, as illustrated in Figure 2. Specifically, DSLM includes three decoders, respectively adopting semantic tokens of vocals (i.e., $\mathbf{S}_v \in \mathcal{S}_v$), accompaniment (i.e., $\mathbf{S}_a \in \mathcal{S}_a$), and song (i.e., $\mathbf{S}_s \in \mathcal{S}_s$) as the prediction targets. Mathematically, we define DSLM : $\mathcal{C} \mapsto \mathcal{S}_s \times \mathcal{S}_v \times \mathcal{S}_a$. By applying an off-the-shelf source separation algorithm to a large corpus of songs with lyrics, a large volume of paired data can be manufactured for the multi-target generation task of interest.

The remainder of this section presents the main contribution of this paper – DSLM, and the attention mask strategies for DSLM.

3.2 Dual-sequence language model

Formally speaking, the proposed dual-sequence language model (DSLM) is tasked with the generation of $(\mathbf{S}_s, \mathbf{S}_v, \mathbf{S}_a)$ given $\mathbf{C}$. An overview of the proposed architecture is presented in Figure 2. Concerning the quadratic complexity of Transformer with respect to sequence length, instead of processing the concatenated sequences of multiple target sequences token-by-token as in [29], in DSLM we utilize different decoders to model the semantic tokens of vocals $\mathbf{S}_v$ and accompaniment $\mathbf{S}_a$ and harmoniously combine them to generate the semantic tokens of song $\mathbf{S}_s$.

The proposed DSLM consists of a lyrics encoder, two decoders (one for vocals and one for accompaniment) inter-connected through a bidirectional cross-attention module, and a final song decoder. The lyrics encoder is built upon a stack of Transformer encoder layers, which, as a architecture widely adopted in speech synthesis [7, 10], extracts critical information related to the pronunciation of the lyrics $\mathbf{C}_{\text{lyrics}}$. On the other hand, the vocal decoder and accompaniment decoder are together composed of multiple DSLM blocks. Each DSLM block is composed of a self-attention (SA) layer, a cross-attention (CA) layer, a bidirectional cross-attention (BCA) layer and a feed-forward layer.

The cross-attention layer is utilized to attend the information from lyrics encoder, , which has been widely applied in previous works on speech synthesis [58, 59] and audio generation [60, 61]. For vocal decoder, it models the alignment between the lyrics and vocals. For accompaniment decoder, it extracts semantic information from the lyrics for generating accompaniment Moreover, in a complete song, the vocal and accompaniment parts have a complex interrelationship. The accompaniment must complement the vocal track without overshadowing them, ensuring that both parts work together to highlight the song's expressive and artistic intents. To understand and model this interrelationship, we introduce a bidirectional cross-attention (BCA) layer, which consists of two symmetrical cross-attention mechanisms. For example, in the vocal decoder, the BCA allows the model to attend to the generated parts of accompaniment while generating vocals, making arrangements accordingly. The BCA layer is then defined as follows:

$$\mathbf{Q}_v = \mathbf{H}_v \mathbf{W}_v^Q, \quad \mathbf{K}_v = \mathbf{H}_a \mathbf{W}_v^K, \quad \mathbf{V}_v = \mathbf{H}_a \mathbf{W}_v^V \tag{1}$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \tag{2}$$

$$\mathbf{A}_v = \text{softmax}\left(\frac{\mathbf{Q}_v \mathbf{K}_v^\top}{\sqrt{d_k}} + \mathbf{M}\right) \tag{3}$$

where $\mathbf{H}_v, \mathbf{H}_a \in \mathbb{R}^{T \times d_h}$ denote the previous layer's outputs from the vocal decoder and accompaniment decoder, respectively. These outputs are linearly projected to a triple of queries, keys and values with learnable weights $\mathbf{W}_v^Q, \mathbf{W}_v^K, \mathbf{W}_v^V \in \mathbb{R}^{d_h \times d_k}$, respectively, and the mask matrix $\mathbf{M} \in \mathbb{R}^{T \times T}$ is used to control whether a pair of tokens can be attended to each other. Here, we use T to denote the length of tokens in LM, and use d_h and d_k to denote the hidden size and attention layer size.

The vocal decoder and accompaniment decoder treats the generation of semantic tokens as conditional language modeling tasks, performing autoregressive predictions token by token. Leveraging the in-context learning capabilities of the language model, we can control various acoustic conditions of the generated audio with a prompting technique. Given a vocal prompt (represented by semantic tokens), denoted as $\hat{\mathbf{S}}_v$, it tends to control a mixture of speaker, vocal melody, and tempo. Similarly, given an accompaniment prompt (represented by semantic tokens), denoted as $\hat{\mathbf{S}}_a$, it tends to control instruments, musical melody, and rhythm. The semantic tokens of prompt audio are passed as a prefix to the DSLM and the model uses this prefix to sequentially predict the following token sequence. Taking the vocal decoder $\boldsymbol{\theta}_{\text{vocal}}$ as an example. The task of the vocal decoder can be formulated as:

$$p(\mathbf{S}_v | \mathbf{C}_{\text{lyrics}}, \hat{\mathbf{S}}_v; \boldsymbol{\theta}_{\text{vocal}}) = \prod_{t=0}^{T} p(\mathbf{S}_{v,t} | \mathbf{S}_{v,<t}, \mathbf{S}_{a,<t}, \mathbf{C}_{\text{lyrics}}, \hat{\mathbf{S}}_v; \boldsymbol{\theta}_{\text{vocal}}) \tag{4}$$

Then, we concatenate the embeddings $\mathbf{E}_v, \mathbf{E}_a \in \mathbb{R}^{T \times d_e}$ from outputs of these two decoders. The combined embeddings $\mathbf{E}_s \in \mathbb{R}^{T \times 2d_e}$ are fed into a song decoder composed of multiple Transformer blocks to non-autoregressively generate the semantic token sequence for the complete song, achieving a natural and seamless integration of vocals and instruments, which can be simply represented as:

$$p(\mathbf{S}_s | \mathbf{E}_v, \mathbf{E}_a; \boldsymbol{\theta}_{\text{song}}) = \prod_{t=0}^{T} p(\mathbf{S}_{s,t} | \mathbf{S}_v, \mathbf{S}_a; \boldsymbol{\theta}_{\text{song}}) \tag{5}$$

3.3 Attention mask strategies for universal song generation

In both self-attention (SA) layer and bidirectional cross-attention (BCA) layer, we employ the mask matrix $\mathbf{M}$ as shown in Equation 2 to control the access of the semantic tokens to be predicted. As shown in Figure 2, we implement multiple mask strategies for SA and BCA using different $\mathbf{M}$.

Specifically, we employ two different masking strategies for the SA to control each semantic token's access to the context within the same sequence. One strategy is the causal attention mask, where the representation of each token can only access the leftward context tokens and itself. This approach predicts a token conditioned on its historical (left) context, thereby learning generation and continuation capabilities, but it is difficult to fully capture the dependencies between the context. The other strategy is the non-causal attention mask, where all token can attend to each other within the same

Table 2: Specific attention mask strategy of all tasks supported by SongCreator. [·] indicates that the condition is optional. * indicates that our proposed model achieves significant improvements in this task.

Tasks	Conditions	Outputs	SA mask	BCA mask
Lyrics-to-song*	Lyrics, [Vocal prompt], [Accompaniment prompt]	Song, Vocals	Causal, Causal	BR
Lyrics-to-vocals*	Lyrics, [Vocal prompt]	Vocals	Causal, Causal	BR
Accompaniment-to-song	Lyrics, Accompaniment, [Vocal prompt]	Song, Vocals	Causal, Non-causal	A2V
Vocals-to-song	Vocals, [Lyrics], [Accompaniment prompt]	Song, Music	Non-causal, Causal	V2A
Music continuation	Accompaniment prompt	Music	None, Causal	None
Song editing*	Lyrics, Vocals, Accompaniment	Song, Vocals	Causal, Causal	BR
Vocals editing	Lyrics, Vocals	Vocals	Causal, None	None
Vocals editing in song*	Lyrics, Vocals, Accompaniment	Song, Vocals	Causal, Non-causal	A2V

sequence. It incorporates contextual information from the entire sequence, and can generate more comprehensive and enriched context representations than the causal approach.

For BCA, we design four masking strategies to control the mutual attention between the semantic token sequences representing vocals and accompaniment. The bidirectional mask (BR) allows representations in both the vocal sequence and accompaniment sequence to attend to representations in the other sequence. However, when predicting the token at time step t, it can only attend to the representation of tokens in the other sequence at time step less than or equal to t. For example, the representation $\mathbf{H}_{v,t}$ of semantic token $\mathbf{S}_{v,t}$ can only pay attention to $\mathbf{H}_{a,\leq t}$, but not to $\mathbf{H}_{a,>t}$. It attempts to capture the relationships between vocals and accompaniment, but does not consider the full context of the other sequence, leading to certain limitations when one sequence is pre-determined. As a supplement, the accompaniment-to-vocals (A2V) and vocals-to-accompaniment (V2A) strategies allow one sequence to attend to all tokens in the other sequence. Take the A2V as an example, the tokens in vocal sequence can attend to the full context of the accompaniment sequence, while tokens in the accompaniment sequence are not allowed to attend to the vocal sequence. In this way, the vocal decoder can generate vocals based on the complete accompaniment information. Similarly, the V2A strategy allows the model to predict accompaniment tokens conditioned on the entire vocals sequence. Additionally, the None strategy means neither sequence can attend to the other, supporting the independent generation of instrumental music or vocals.

By employing different mask strategies for SA and BCA, as well as the input format, a single SongCreator can achieve competitive performance on multiple song generation tasks, as shown in Table 2 and Appendix B. We also demonstrate in the ablation studies that the specific attention mask we employed for each task are effective. Furthermore, we support additional tasks shown on our demo page.

3.4 Training Setup

We investigate a multi-task training setup, in which the model is trained on several tasks to enhance its composition, arrangement, and comprehension abilities. We consider the following three tasks:

Song generation from lyrics In this task, the SA in both the vocals decoder and the accompaniment decoder employs the causal attention mask to simultaneously generate vocal and accompaniment semantic tokens. For BCA, 80% of the time we use the bidirectional attention mask to learn how to generate harmoniously coordinated vocals and accompaniment. In the remaining 20% of the time, we use the None strategy to allow the model to learn to generate accompaniment or vocal track independently. This probability setting was inspired by classifier-free guidance related work [54, 62] to ensure it does not disrupt the training of the BCA.

Song generation from pre-determined accompaniment or vocals Take the accompaniment is determined as an example, in this task, the SA in the vocals decoder maintains the causal mask to generate vocals, while the SA in the accompaniment decoder employs the non-causal mask, with the BCA using the A2V strategy. Note that for the non-causal mask, we randomly mask 20% of tokens in the input sequence, to encourage the model to learn the relationships between context tokens. Furthermore, for the above two training tasks, we provide the model with a vocal and accompaniment prompt to encourage the model to learn to control the acoustic conditions of the generated audio.

Song editing The song editing task combines the above two tasks. The difference is that we randomly select a span of tokens from the end of the target sequence to replace the audio prompt, using a special token <EDIT> in between to distinguish the editing task from the generation task.

In all training tasks, the vocal decoder and accompaniment decoder are trained using the next token prediction objective, and the song decoder predicts the semantic tokens of the complete song based on the embeddings extracted from the vocal decoder and accompaniment decoder. After that, we calculate the cross-entropy loss for vocals, accompaniment and song, and optimize the DSLM with the sum of these losses. Calculating the loss of the song also helps the model effectively reduce the impact of the source separation tool on the overall quality of the generated song. Note that we follow previous works [54, 63] and calculate the loss on all tokens, not just the masked tokens, for non-causal strategy. Moreover, we also mask the lyrics 20% of the time to encourage the model to attempt unconditional generation.

4 Experiments

4.1 Experimental setup

Data and model DSLM is trained on 8,500 hours of song data with lyrics (approximately 270,000 songs). We employed an automatic speech recognition (ASR) model to provide timestamps for each sentence in the lyrics and a voice activity detection (VAD) model to detect silent segments. Then, we select appropriate silent segments to split the dataset into 1.7M clips, each no longer than 30 seconds and ensuring the completeness of the sentences. Each clip is input into the Demucs [64, 65] music source separation model to extract vocals and accompaniment. Our DSLM has approximately 0.6B parameters. Detailed configurations are shown in Appendix A.1.

Training and Inference During training, we train the DSLM for 500K steps using 8 NVIDIA A800 GPUs, with a batch size of 8 for each GPU. Adam optimizer is used with $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$ and follow the same learning rate schedule in [66]. Consistently, top-k sampling is adopted for inference, in which k and temperature are set to 50 and 0.9, respectively.

Evaluations Most tasks are evaluated using both objective and subjective metrics.[2] For objective evaluations, Fréchet Audio Distance (FAD) [67] is used to evaluate the generation fidelity; Mel Cepstral Distortion (MCD) is used to measure the spectral distance between the synthesized and Ground Truth; Speaker Embedding Cosine Similarity (SECS) is used for the similarity of speaker identity. For subjective evaluations, we utilize the commonly used mean opinion score (MOS) tests. In various tasks, we assess multiple aspects: musicality, quality (focusing on clarity and intelligibility), style similarity (including speaker, melody and instruments), harmony between vocals and accompaniment, and naturalness. Moreover, AB preference tests are also conducted. The appendix G shows details of the evaluations.

Baselines We conducted comprehensive comparisons between SongCreator and multiple baselines on each task. First, we establish two baseline models for each task. One is SongCreator (Single) trained on a specific sequence generation task, and the other replaces DSLM with GPT [68] in SongCreator to predict the target sequence, named GPT. For lyrics-to-song, we directly conditioned SOTA music generation models, MusicLM [25] and MusicGen [22], on lyrics to predict songs. Furthermore, we add another baseline where GPT is used to first predict vocals and then predict the song, named GPT (Vocals & Song). For lyrics-to-vocals, in addition to MusicLM, we also introduce the SOTA text-to-speech method VALL-E [9]. For vocals-to-song and accompaniment-to-song, we utilize the structure proposed in SingSong [29] to perform these two tasks, respectively. To ensure a fair comparison, we replace the semantic and acoustic tokens with BEST-RQ [57] tokens and use our latent diffusion model to convert them into the waveform, establishing another baseline, SingSong (Diffusion). For music continuation, we employ AudioLM [36] as a baseline. The detailed implementations of each baseline are shown in Appendix C.

4.2 The results of tasks

[2]Following the setting of DiffSinger [19], vocals generation tasks don't report the objective results.

Table 3: Lyrics-to-song evaluation without audio prompt.

Model	FAD ↓	Musicality ↑	Quality ↑
Ground Truth	-	4.3 ± 0.04	4.09 ± 0.05
MusicLM	6.47	3.21 ± 0.09	3.25 ± 0.07
MusicGen	2.31	3.08 ± 0.06	2.99 ± 0.06
GPT	8.18	3.32 ± 0.10	3.26 ± 0.08
GPT (Vocals & Song)	11.23	3.55 ± 0.09	3.64 ± 0.07
SongCreator	**2.14**	$\mathbf{4.25 \pm 0.05}$	$\mathbf{4.08 \pm 0.06}$
SongCreator (Single)	3.04	3.85 ± 0.06	3.75 ± 0.05

Table 4: Lyrics-to-vocals evaluation without audio prompt.

Model	Musicality ↑	Quality ↑
Ground Truth	3.89 ± 0.09	3.91 ± 0.07
MusicLM	3.31 ± 0.06	3.35 ± 0.06
VALL - E	3.15 ± 0.08	3.23 ± 0.06
GPT	3.64 ± 0.07	3.58 ± 0.07
SongCreator	$\mathbf{3.98 \pm 0.04}$	$\mathbf{3.79 \pm 0.05}$
SongCreator (Vocal Only)	3.68 ± 0.06	3.63 ± 0.05
SongCreator (Single)	3.53 ± 0.06	3.64 ± 0.05

Table 5: Prompt-based lyrics-to-song. We sample the prompt at random from a held-out set.

Model	FAD ↓	MCD ↓	Musicality ↑	Similarity ↑
Ground Truth	-	-	4.04 ± 0.06	3.79 ± 0.09
MusicGen	**1.90**	9.78	3.46 ± 0.11	3.27 ± 0.11
SongCreator	2.06	**8.44**	$\mathbf{4.01 \pm 0.07}$	$\mathbf{3.82 \pm 0.08}$

Table 6: Prompt-based lyrics-to-vocals. We sample the prompt at random from a held-out set.

Model	SECS ↑	Musicality ↑	Similarity ↑
Ground Truth	0.62	3.63 ± 0.08	3.57 ± 0.08
VALL - E	0.66	3.34 ± 0.07	3.30 ± 0.08
SongCreator	**0.68**	$\mathbf{3.57 \pm 0.06}$	$\mathbf{3.55 \pm 0.07}$

Lyrics-to-song As shown in Table 3, our proposed SongCreator significantly outperforms the baselines across all three metrics, confirming the effectiveness of SongCreator. The difference between SongCreator and Ground Truth is merely 0.05 and 0.01 for musicality and quality, respectively. SongCreator (Single) and GPT (Vocals & Song) perform better than other baselines, demonstrating the difficulty of directly modeling the complete song. Additionally, we use the same lyrics from the demos of the previous SOTA model Jukebox [12] and conduct the AB preference test. As shown in Table 15, SongCreator is preferred over Jukebox 60% of the time.

To investigate the ability of SongCreator to maintain acoustic conditions from prompts, we compared it with MusicGen. The results are shown in Table 5. SongCreator achieved scores of 4.01 in musicality and 3.82 in similarity, considerably improving upon MusicGen's scores of 3.46 and 3.27, with only a slightly lower score of 0.16 in FAD. In addition, SongCreator can independently control the acoustic conditions of the vocals and accompaniment in the generated song. This capability is lacking in previous methods and results can be found on the demo page.

Lyrics-to-vocals SongCreator provides two inference methods for lyrics-to-vocals. One is similar to lyrics-to-song, where the model considers the relationship between the vocals and accompaniment to generate both vocal and accompaniment tokens, but we only use the generated vocals. The other doesn't use BCA and the accompaniment decoder, relying solely on the vocal decoder to generate the vocals, named SongCreator (Vocal Only). As shown in Table 4, SongCreator (Vocal Only) achieves scores of 3.68 in musicality and 3.63 in quality, comparable to the performance of SongCreator (Single) and GPT. However, after considering the relationship between vocals and accompaniment, SongCreator surpasses these models with a substantially higher score of 3.98 in musically. In this study, we also conduct a zero-shot evaluation of the vocals between our proposed model and VALL-E. Table 6 presents the results. From the performance evaluated by MOS and SECS, our proposed model outperforms VALL-E, especially in terms of similarity, demonstrating SongCreator's robust zero-shot clone ability for generating vocals.

Vocals-to-song and accompaniment-to-song As shown in Table 7 and Table 8, our proposed SongCreator gets comparable results with recent SOTA models in terms of musicality and harmony.

Table 7: Vocals-to-song evaluation.

Model	FAD ↓	Musicality ↑	Harmony ↑
Ground Truth	-	4.12 ± 0.05	3.91 ± 0.08
SingSong	3.37	3.67 ± 0.10	3.63 ± 0.08
SingSong (Diffusion)	4.13	3.71 ± 0.08	3.67 ± 0.06
GPT	3.07	3.73 ± 0.07	3.69 ± 0.07
SongCreator	1.88	$\mathbf{3.77 \pm 0.08}$	$\mathbf{3.77 \pm 0.07}$
SongCreator (Single)	**1.46**	3.58 ± 0.08	3.65 ± 0.06

Table 8: Accompaniment-to-song evaluation.

Model	FAD ↓	Musicality ↑	Harmony ↑
Ground Truth	-	4.15 ± 0.07	4.11 ± 0.07
SingSong	1.82	3.36 ± 0.06	3.42 ± 0.07
SingSong (Diffusion)	2.98	3.66 ± 0.06	3.65 ± 0.05
GPT	1.64	3.53 ± 0.08	3.53 ± 0.09
SongCreator	1.24	$\mathbf{3.67 \pm 0.05}$	$\mathbf{3.78 \pm 0.06}$
SongCreator (Single)	**1.23**	3.60 ± 0.07	3.62 ± 0.06

For the FAD score, our model reaches 1.88 and 1.24 on the two tasks, respectively, outperforming SingSong. A possible reason is that our model considers the complete song, rather than just the partially separated vocals considered in SingSong. In addition, we used the same vocals (6 samples) in SingSong's demos to generate songs with our model, and asked subjects to choose their preferred songs. As shown in Table 16, SingSong gets an extra preference (54.1%) over SongCreator (30%). We speculate one of the reasons is that SingSong uses a large-scale high-quality dataset (46k hours).

Music Continuation For the music continuation task, we compare different models by generating 10s music based on a 5s instrumental music prompt. As illustrated in Table 9, we can see that SongCreator achieves comparable results with AudioLM and GPT. This indicates that SongCreator can effectively continue the musical elements in the prompt, providing the capability to control the accompaniment in song generation.

Table 9: Music continuation evaluation.

Model	**FAD** ↓	**Musicality** ↑	**Similarity** ↑
Ground Truth	-	3.9 ± 0.11	3.70 ± 0.10
AudioLM	1.33	3.95 ± 0.10	3.78 ± 0.08
GPT	**1.28**	3.90 ± 0.10	3.73 ± 0.11
SongCreator	1.54	$\mathbf{3.97 \pm 0.08}$	$\mathbf{3.83 \pm 0.08}$

Table 10: Song editing evaluation.

Model	**FAD** ↓	**MCD** ↓	**Musicality** ↑	**Naturalness** ↑
Ground Truth	-	-	4.08 ± 0.07	3.99 ± 0.06
GPT	2.29	8.30	3.84 ± 0.07	3.72 ± 0.06
SongCreator	**1.81**	7.90	$\mathbf{4.01 \pm 0.06}$	$\mathbf{3.78 \pm 0.07}$
SongCreator (Single)	1.87	**7.85**	3.93 ± 0.08	3.75 ± 0.08

Table 11: Vocals editing evaluation.

Model	**SECS** ↑	**Musicality** ↑	**Naturalness** ↑
Ground Truth	-	3.65 ± 0.08	3.45 ± 0.07
GPT	0.87	3.64 ± 0.07	$\mathbf{3.43 \pm 0.07}$
SongCreator	0.87	$\mathbf{3.68 \pm 0.06}$	3.31 ± 0.06
SongCreator (Single)	0.87	3.63 ± 0.06	3.41 ± 0.06

Editing tasks To evaluate the performance on editing tasks, we manually constructed a dataset of 30 song editing examples, as shown in Appendix D. Table 10 presents the results of song editing. We can see that SongCreator gets comparable performance in terms of naturalness to the baselines. However, benefiting from its strong ability to generate song, SongCreator surpasses these baselines in musicality, achieving a score of 4.01. In the vocal editing, as shown in Table 11, all three models achieve relatively close performance in both subjective and objective evaluations. To demonstrate the editing ability of SongCreator, we further conduct the AB preference test on three tasks: song editing, vocals editing, and vocals editing in song. In each task, SongCreator restores the masked song using its original lyrics and compares it with the audio samples reconstructed using BEST-RQ encoding and LDM decoding to eliminate the potential impact from the encoding and decoding processes during our experiments. The results are shown in Table 17. In all tasks, there is no significant difference between the generated song and the Ground Truth ($p > 0.01$), where the p-values are calculated using the Wilcoxon signed-rank test. This means that humans judge the edited song produced by SongCreator to be as natural as the original unedited song.

4.3 Ablation Studies

The influence of multi-task training Through previous experiments, we can find that multi-task training significant improves most tasks, especially in lyrics-to-song. This indicates that the DSLM effectively capture the shared information between different tasks, such as composition, arrangement and the relationship between vocals and accompaniment.

The influence of bidirectional cross-attention layer We evaluate the SongCreator and the model without using BCA on lyrics-to-song and lyrics-to-vocals. Figure 3 shows the results. When the BCA is removed from the DSLM, the performance on lyrics-to-song exhibit a marked deterioration, suggesting utilizing BCA is helpful for the model generate harmonious vocals and accompaniment. Interestingly, the performance also declined on the lyrics-to-vocals task, demonstrating that learning the relationships between vocals and accompaniment is also beneficial for generating vocals.

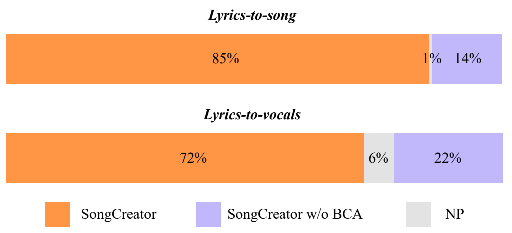


Figure 3: Results of the AB preference test between SongCreator and the model without using BCA.

The influence of attention mask strategies in self-attention layer To validate our designed SA mask strategies, we disable the non-causal mask of SA during training and conduct an AB preference test to compare this version with SongCreator on three tasks: lyrics-to-song, vocals-to-song, and accompaniment-to-song. As shown in Figure 4, the performance on all three tasks showed significant degradation, especially for vocals-to-song. These results indicate that incorporating the non-causal attention mask assists the learning of the relationships within the context and provides additional contextual information for generation.

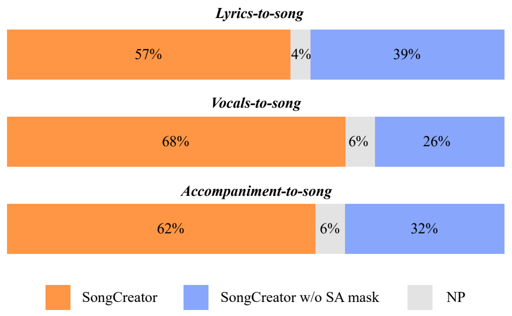


Figure 4: Results of the AB preference test between SongCreator and the model without using non-causal mask in SA.

The influence of attention mask strategies in bidirectional cross-attention layer To validate our designed BCA mask strategies, we conduct AB preference tests for the lyrics-to-song and accompaniment-to-song tasks. For lyrics-to-song, we compared BR strategy with A2V, V2A and None strategy. As shown in Table 18, replacing the BR strategy with other strategies leads to a significant performance deterioration, demonstrating that the BR strategy is helpful for the model generate harmonious vocals and accompaniment. The None strategy, which disregards the relationship between vocals and accompaniment, performed the worst. In accompaniment-to-song, we compared A2V strategy with BR strategy. Table 19 shows the results, We find that participants preferred the song generated with the A2V strategy. We believe that this is because the A2V strategy provides more context about the accompaniment sequence when generating vocals.

5 Conclusion and Discussion

Conclusion In this paper, we propose SongCreator, a system designed for lyrics-based song generation. We introduce a dual-sequence language model (DSLM) to separately model vocals and accompaniment information, and employs a dynamic bidirectional cross-attention module to capture the influences between these two sequences, with designing a serious of attention mask strategies for DSLM. In experiments, the proposed SongCreator provides competitive performance on all eight tasks.

Limitations We acknowledge the limitations of our proposed SongCreator. Due to the challenges in collecting data, SongCreator currently cannot control the genre and style of the output songs through text descriptions. Besides, the interference from accompaniment in the song makes it difficult for BEST-RQ to fully encode the vocal information, imposing a limited clarity of the synthesized vocals – in further work, we hope to extract better semantic representations for songs. Another issue is that the proposed model can only generate songs up to 30s, which is insufficient for supporting the generation of songs with complete structures.

Broader Impact We believe that our work has huge potential to develop into a song creation tool for content creators or novices to seamlessly express their creative pursuits with a low entry barrier, while also streamline and improve the workflow of experienced music producers. However, the potential negative impacts of SongCreator can't be overlooked. One of the primary concerns is the ability to replicate someone's voice with the vocal prompt, which could be exploited in the generation of misinformation, deepfake audio, or any harmful content. We are committed to advancing the field responsibly, and therefore, the checkpoints trained on the full dataset will not be released.

Acknowledgements

This work is supported by National Natural Science Foundation of China (62076144) and Shenzhen Science and Technology Program (WDZC20220816140515001, JCYJ20220818101014030).

References

[1] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[3] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[4] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[5] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.

[6] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022.

[7] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*, 2021.

[8] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR, 2021.

[9] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

[10] Kai Shen, Zeqian Ju, Xu Tan, Eric Liu, Yichong Leng, Lei He, Tao Qin, sheng zhao, and Jiang Bian. Naturalspeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Ziyue Jiang, Jinglin Liu, Yi Ren, Jinzheng He, Zhenhui Ye, Shengpeng Ji, Qian Yang, Chen Zhang, Pengfei Wei, Chunfeng Wang, et al. Boosting prompting mechanisms for zero-shot speech synthesis. In *The Twelfth International Conference on Learning Representations*, 2023.

[12] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[13] Suno. [Online]. `https://suno.com/`.

[14] Udio. [Online]. `https://www.udio.com/`.

[15] Jiawei Chen, Xu Tan, Jian Luan, Tao Qin, and Tie-Yan Liu. Hifisinger: Towards high-fidelity neural singing voice synthesis. *arXiv preprint arXiv:2009.01776*, 2020.

[16] Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2525–2535, 2022.

[17] Yongmao Zhang, Jian Cong, Heyang Xue, Lei Xie, Pengcheng Zhu, and Mengxiao Bi. Visinger: Variational inference with adversarial learning for end-to-end singing voice synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7237–7241, 2022.

[18] Zhiqing Hong, Chenye Cui, Rongjie Huang, Lichao Zhang, Jinglin Liu, Jinzheng He, and Zhou Zhao. Unisinger: Unified end-to-end singing voice synthesis with cross-modality information matching. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7569–7579, 2023.

[19] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11020–11028, 2022.

[20] Ji-Sang Hwang, Sang-Hoon Lee, and Seong-Whan Lee. Hiddensinger: High-quality singing voice synthesis via neural audio codec and latent diffusion models. *arXiv preprint arXiv:2306.06814*, 2023.

[21] Shuangrui Ding, Zihan Liu, Xiaoyi Dong, Pan Zhang, Rui Qian, Conghui He, Dahua Lin, and Jiaqi Wang. Songcomposer: A large language model for lyric and melody composition in song generation. *arXiv preprint arXiv:2402.17645*, 2024.

[22] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Max WY Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, et al. Efficient neural music generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[24] Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion. *arXiv e-prints*, pages arXiv–2301, 2023.

[25] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

[26] Maarten Grachten, Stefan Lattner, and Emmanuel Deruty. Bassnet: A variational gated autoencoder for conditional generation of bass guitar tracks with learned interactive control. *Applied Sciences*, 10(18):6627, 2020.

[27] Yueh-Kao Wu, Ching-Yu Chiu, and Yi-Hsuan Yang. Jukedrummer: Conditional beat-aware audio-domain drum accompaniment generation via transformer vq-vae. In *Ismir 2022 Hybrid Conference*, 2022.

[28] Yin-Cheng Yeh, Wen-Yi Hsiao, Satoru Fukayama, Tetsuro Kitahara, Benjamin Genchel, Hao-Min Liu, Hao-Wen Dong, Yian Chen, Terence Leong, and Yi-Hsuan Yang. Automatic melody harmonization with triad chords: A comparative study. *Journal of New Music Research*, 50(1): 37–51, 2021.

[29] Chris Donahue, Antoine Caillon, Adam Roberts, Ethan Manilow, Philippe Esling, Andrea Agostinelli, Mauro Verzetti, Ian Simon, Olivier Pietquin, Neil Zeghidour, et al. Singsong: Generating musical accompaniments from singing. *arXiv preprint arXiv:2301.12662*, 2023.

[30] Hong Zhiqing, Huang Rongjie, Cheng Xize, Wang Yongqi, Li Ruiqi, You Fuming, Zhao Zhou, and Zhang Zhimeng. Text-to-song: Towards controllable music generation incorporating vocals and accompaniment. *arXiv preprint arXiv:2404.09313*, 2024.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[32] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32, 2019.

[33] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022.

[34] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[35] Xipin Wei, Junhui Chen, Zirui Zheng, Li Guo, Lantian Li, and Dong Wang. A Multi-Scale Attentive Transformer for Multi-Instrument Symbolic Music Generation. In *Proc. INTERSPEECH 2023*, pages 5391–5395, 2023.

[36] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[37] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265, 2015.

[38] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[39] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

[40] Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1206–1210. IEEE, 2024.

[41] Seth* Forsgren and Hayk* Martiros. Riffusion - Stable diffusion for real-time music generation. 2022. URL `https://riffusion.com/about`.

[42] Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023.

[43] Haohe Liu, Qiao Tian, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D Plumbley. Audioldm 2: Learning holistic audio generation with self-supervised pretraining. *arXiv preprint arXiv:2308.05734*, 2023.

[44] Daxin Tan, Liqun Deng, Yu Ting Yeung, Xin Jiang, Xiao Chen, and Tan Lee. Editspeech: A text based speech editing system using partial inference and bidirectional fusion. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 626–633. IEEE, 2021.

[45] Tao Wang, Jiangyan Yi, Ruibo Fu, Jianhua Tao, and Zhengqi Wen. Campnet: Context-aware mask prediction for end-to-end text-based speech editing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:2241–2254, 2022.

[46] He Bai, Renjie Zheng, Junkun Chen, Mingbo Ma, Xintong Li, and Liang Huang. A3t: Alignment-aware acoustic and text pretraining for speech synthesis and editing. In *International Conference on Machine Learning*, pages 1399–1411, 2022.

[47] Zalan Borsos, Matthew Sharifi, and Marco Tagliasacchi. SpeechPainter: Text-conditioned Speech Inpainting. In *Proc. Interspeech 2022*, pages 431–435, 2022.

[48] Ziyue Jiang, Qian Yang, Jialong Zuo, Zhenhui Ye, Rongjie Huang, Yi Ren, and Zhou Zhao. Fluentspeech: Stutter-oriented automatic speech editing with context-aware diffusion models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11655–11671, 2023.

[49] Chenpeng Du, Yiwei Guo, Feiyu Shen, Zhijun Liu, Zheng Liang, Xie Chen, Shuai Wang, Hui Zhang, and Kai Yu. Unicats: A unified context-aware text-to-speech framework with contextual vq-diffusion and vocoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17924–17932, 2024.

[50] Dacheng Yin, Chuanxin Tang, Yanqing Liu, Xiaoqiang Wang, Zhiyuan Zhao, Yucheng Zhao, Zhiwei Xiong, Sheng Zhao, and Chong Luo. RetrieverTTS: Modeling Decomposed Factors for Text-Based Speech Insertion. In *Proc. Interspeech 2022*, pages 1571–1575, 2022.

[51] Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023.

[52] Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision. *Transactions of the Association for Computational Linguistics*, 11:1703–1718, 2023.

[53] Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv preprint arXiv:2303.03926*, 2023.

[54] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36, 2024.

[55] Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. Speechx: Neural codec language model as a versatile speech transformer. *arXiv preprint arXiv:2308.06873*, 2023.

[56] Puyuan Peng, Po-Yao Huang, Daniel Li, Abdelrahman Mohamed, and David Harwath. Voicecraft: Zero-shot speech editing and text-to-speech in the wild. *arXiv preprint arXiv:2403.16973*, 2024.

[57] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2022.

[58] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6706–6713, 2019.

[59] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[60] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, pages 13916–13932. PMLR, 2023.

[61] Stable audio: Fast timing-conditioned latent audio diffusion. [Online]. `https://stability.ai/research/stable-audio-efficient-timing-latent-diffusion`.

[62] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*, 2024.

[63] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, et al. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*, 2022.

[64] Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid transformers for music source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[65] Alexandre Défossez. Hybrid spectrogram and waveform source separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[67] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet Audio Distance: A Reference-Free Metric for Evaluating Music Enhancement Algorithms. In *Proc. Interspeech 2019*, pages 2350–2354, 2019.

[68] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners.

[69] Luca Lanzendörfer, Florian Grötschla, Emil Funke, and Roger Wattenhofer. Disco-10m: A large-scale music dataset. *Advances in Neural Information Processing Systems*, 36, 2024.

[70] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

[71] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, et al. Mert: Acoustic music understanding model with large-scale self-supervised training. *arXiv preprint arXiv:2306.00107*, 2023.

[72] Minz Won, Yun-Ning Hung, and Duc Le. A foundation model for music informatics. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1226–1230. IEEE, 2024.

[73] Andrew Hines, Jan Skoglund, Anil C Kokaram, and Naomi Harte. Visqol: an objective speech quality model. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015:1–18, 2015.

[74] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[75] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023.

[76] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135, 2017.

A Training and Implementation Details

A.1 Dual-sequence language model

Our DSLM consists of a lyrics encoder and three decoders. The lyrics encoder is a 4-layer Transformer [66] with 1024 hidden size. The vocal decoder and accompaniment decoder have a similar architecture that contains 8 DSLM layers with 1024 hidden size. The song decoder also consists of 4 feed-forward Transformer layers with 1024 hidden size. We provide detailed hyper-parameter settings about this model configuration in Table 12. We collected approximately 8500 hours of songs with lyrics from the internet for model training, comprising part of the DISCO-10M [69] dataset and some in-house datasets.

Table 12: Hyper-parameters of DSLM model

Hyper-parameter		Value
Lyrics Encoder	Encoder Layers	4
	Hidden Size	1024
	Attention Head	16
	Feed-Forward Dim	4096
	Max Context Length (in #tokens)	256
Vocal Decoder & Accompaniment Decoder	Decoder Layers	8
	Hidden Size	1024
	Attention Head	16
	Feed-Forward Dim	4096
	Max Context Length (in #tokens)	1500
Song Decoder	Decoder Layers	4
	Hidden Size	1024
	Attention Head	16
	Feed-Forward Dim	4096
	Max Context Length (in #tokens)	1500
Total Number of Parameters		631M

A.2 BEST-RQ with vector quantization

BERT-based **S**peech pre-**T**raining with **R**andom-projection **Q**uantizer (BEST-RQ) [57] is a simple and effective self-supervised learning model that learns representations from audio data without manually labeled annotations. This self-supervised algorithm helps alleviate the scarcity of song data with lyrics and provides a robust foundation for the entire generation system.

Our implementation of BEST-RQ was based on an open-source library.[3] In particular, our implementation follows the same architecture as of BEST-RQ [57], but with a codebook's vocabulary size of 1024. For feature extraction, 80-dimensional log Mel-spectrograms are extracted with 24kHz sampling rate with a hop size of 480 and fed into the model to obtain a 50Hz sequence of 1024-dimensional latent representations. We train this model, which has approximately 0.6 billion parameters, using our prepared 100k hours of audio data in the self-supervised learning (SSL) manner described in [57]. Furthermore, as we aim to achieve universal song generation, our training dataset includes not only complete songs with vocals and accompaniment but also separate instrumental music and vocals. This diverse dataset ensures that our model gains a comprehensive understanding of different music elements and their interactions, enhancing its ability to generate a wide array of musical and vocal outputs.

Next, we train a Vector Quantization (VQ) module to quantize the 1024-dimensional latent representations extracted from the 14th layer of the Conformer within the BEST-RQ model. Our implementation of the VQ module was based on an open-source library,[4] with codebook size of 16384 and codebook dimensional with 32. By combining BEST-RQ and the VQ module, we can extract 50Hz semantic token sequences from the audio.

[3]Implemented based on: https://github.com/lucasnewman/best-rq-pytorch.

[4]Implemented based on: https://github.com/lucidrains/vector-quantize-pytorch.

Table 13: Reconstructed music performance results for different semantic tokenizers.

Model	ViSQOL
HuBERT	2.47
MERT	2.90
MusicFM	2.94
BEST-RQ	**3.05**

Moreover, we offer a comparison of various prevalent semantic tokenizers, such as HuBERT [70], MERT [71] and MusicFM [72]. In this experiment, we train a latent diffusion model (LDM) for each tokenizer to convert sequences of quantized representations into audio, and randomly selected 50 song segments for fair comparison. To better evaluate the performance of reconstructing music with different semantic tokenizers, we use ViSQOL [73] as an audio quality assessment metric. As shown in Table 13, BEST-RQ providing a greater advantage in music reconstruction.

A.3 Latent diffusion model

As shown in Figure 1, we train a latent diffusion model (LDM) as a renderer, which converts a 50Hz semantic token sequence into a 44.1kHz audio, such as songs, vocals, and instrumental music. In contrast to DDPM [38], which directly models the raw training data, LDM operates on a low-dimensional latent space to significantly reduce the computational cost and improve the generation stability. Our implementation of the latent diffusion model was based on the open-source Stable Audio.[5] The reproduced latent diffusion model is composed of a VAE and a U-NET-based conditioned diffusion model.

In particular, for the VAE, we use the same encoder-decoder network architecture as in DAC. [6] To train the VAE, we first adopted the pre-trained model provided in DAC, then fine-tuned the encoder and decoder components (i.e., replacing the vector quantizers with a diagonal Gaussian re-sampler as in LDM). We retained the frequency-domain reconstruction loss, discriminators and adversarial loss from DAC and added a KL loss typically used for training VAEs. The VAE was trained on our prepared dataset of 100k hours of songs data, which is the same as the one used for training BEST-RQ. For the network configurations, the encoder (downsampler) uses strides of [4, 4, 8, 8], d_model of 128 and latent_dim of 64, where the 64-dim matrix is employed as the mean and variance of VAE latents (in 32-dim). Besides, the decoder (upsampler) uses strides of [8, 8, 4, 4] and hidden channels of 1536 to transform the 64-dim latents back to 44.1kHz audio. Based on this pre-trained VAE, we subsequently train a diffusion model in a way similar to Stable Audio 1.0, except having 32-dim latents as targets and semantic tokens as conditions.

B The details of all tasks supported by SongCreator

Benefiting from our specially designed attention mask strategies and multi-task training approach, SongCreator can effectively support the following eight tasks:

Lyrics-to-song This task aims to generate a complete song that includes harmoniously integrated vocal and accompaniment from lyrics. Therefore, we use the causal mask for the SA in both vocal and accompaniment decoders to support autoregressive generation. Regarding the BCA mask strategies, since vocals and accompaniment need to be generated simultaneously, we use the BR strategy to consider the interrelationship between vocals and accompaniment so as to ensure the harmony of vocals and accompaniment.

SongCreator supports to control various acoustic conditions in the generated song by providing optional prompts. The vocal prompt can control speaker, vocal melody, and tempo, while the accompaniment prompt can control instruments, musical melody, and rhythm. The vocal prompt and accompaniment prompt can either be present simultaneously, exist individually, or be absent altogether.

[5]Implemented based on: https://github.com/Stability-AI/stable-audio-tools.

[6]Implemented based on: https://github.com/descriptinc/descript-audio-codec.

Lyrics-to-vocals This task aims to generate the vocals without accompaniment based on the given lyrics. As mentioned before, we use the same attention mask strategy as lyrics-to-song task to support better vocal generation. In this case, the vocal prompt can be provided to control the speaker, melody, and tempo of the generated vocals.

Accompaniment-to-song The purpose of this task is to supplement the vocal track of a song based on the given lyrics for a pre-determined accompaniment track. The vocal track in generated song should complement the input accompaniment track to create a coherent song. To better encode the contextual representation of the input accompaniment track, we use a non-causal mask strategy for the SA in accompaniment decoder. For the SA in vocal decoder, we use a causal mask strategy for autoregressive sequence generation. And to ensure that the generated vocal sequence can consider the full context of the input accompaniment track, we use the A2V strategy in BCA. Similar to lyrics-to-vocals, the generated vocals can also be controlled using the vocal prompt.

Vocals-to-song Contrary to the accompaniment-to-song task, the purpose of this task is to generate harmonious accompaniment for the input vocal track and combine them to create a coherent song. Thus, in this task, the attention mask strategy is set up in contrast to the Accompaniment-to-song task. Similarly, the generated accompaniment can be controlled by the accompaniment prompt.

Music continuation This task is expected to generate instrumental music, which is coherent with the accompaniment prompt in terms of instruments, melody, harmony and rhythm. In this task, we only utilize the accompaniment decoder and use a causal mask in SA for sequence generation. And to support independent accompaniment sequence generation, we use the None strategy in BCA.

Song editing This task requires a model to alter a segment within a song to match a target lyrics. The modified segment must be coherent with the unedited parts of the original song, i.e., maintaining the speaker, instruments, melody and rhythm. Considering that it has similar requirements to the lyrics-to-song task, we use the same attention mask strategy.

Vocals editing This task is similar to song editing, but the modification target is changed from the complete song to the vocals without accompaniment. Thus, we only utilize the vocal decoder and use a causal mask in SA for vocal sequence generation. And to support independent vocal sequence generation, we use the None strategy in BCA.

Vocals editing in song This is a unique capability of SongCreator, which modifies the content of the vocal track in a song while keeping the original accompaniment track unchanged. It means that the modified vocal segment not only maintains coherence with the unedited vocal track of the original song but also harmonizes with the accompaniment in the original song. Considering that it has similar requirements to the accompaniment-to-song task, we use the same attention mask strategy.

C Detailed baseline settings

All baselines are trained using similar strategies to those used for DSLM, includes the same dataset, training resources, optimizer settings, and similar parameter scales. Each model was trained for 500K steps. Additionally, for a fair comparison, baselines with semantic tokens as the prediction target (e.g., GPT, SingSong (Diffusion)) shared the same BEST-RQ and LDM modules as DSLM. Here are the implementation details for each baselines:

SongCreator (Single) Our proposed SongCreator is trained on multiple tasks. For comparison, we keep the model's structure and hyperparameters and train it on different specific tasks, resulting in SongCreator (Single) for each task.

GPT Inspired by UniAudio [51], we set up this baseline model, treating each task as a conditional language modeling task. For each task, we first tokenize both the conditional and target audio using BEST-RQ. Then, we concatenate the source-target pair as a single sequence and perform the next-token prediction task using GPT [68]. Our implementation of GPT was based on an open-source library,[7] that contains 24 Transformer layers with 1024 hidden size and 4096 feed-forward

[7]Implemented based on: https://github.com/karpathy/nanoGPT.

dimensional. Finally, we convert the predicted semantic token sequence into audio by the pre-trained latent diffusion model.

MusicLM MusicLM [25] has demonstrated excellent performance in text-to-music generation. Inspired by this, we attempted to employ its methods to lyrics-to-song and lyrics-to-vocals. Specifically, to achieve this, we make some modifications to the open-source library.[8] First, we replace the MuLan in MusicLM with a lyrics encoder to better encode phoneme information, and replace w2v-BERT with BEST-RQ to more effectively extract semantic tokens from songs. Additionally, since SoundStream [74] is not open-source, we used the widely adopted Encodec [75] as a substitute. Our reproduced MusicLM follows the same hyperparameters as [25], using 24 layers of decoder-only Transformers for both the semantic stage and acoustic stage.

MusicGen In addition to MusicLM, MusicGen [22] is another SOTA model in text-to-music generation. Our implementation of MusicGen for lyrics-to-song is based on the official open-source library.[9] It directly predicts the acoustic tokens extracted by Encodec from the lyrics, without additional semantic tokens. Similar to other baselines, we also use 24 Transformer layers to ensure this model has approximately 0.6B parameters. Moreover, considering that MusicGen allows control the generated output through prompts, we also compared it with our proposed SongCreator for the prompt-based lyrics-to-song evaluation.

VALL-E Recently, language model-based text-to-speech models (e.g., VALL-E) have shown the capability of generating high-quality personalized speech with a 3s acoustic prompt. Considering the similarity between text-to-speech and lyrics-to-vocals tasks, we attempted to directly apply VALL-E to the lyrics-to-vocals. Our implementation is based on the open-source library.[10] To ensure a fair comparison, both the autoregressive transformer decoder and the non-autoregressive transformer decoder in VALL-E are composed of 24 layers, 16 attention heads, an embedding dimension of 1024, and feed-forward layers of dimensionality 4096. And we also compared the zero-shot voice cloning abilities of SongCreator and VALL-E.

AudioLM To validate the performance of SongCreator in music continuation, we implement AudioLM [36] based on the open-source code.[11] Similar to our settings with MusicLM, we replace w2v-BERT with BEST-RQ and Soundstream with Encodec in AudioLM. Additionally, we also used a 24-layer decoder-only transformer structure for both the semantic and acoustic stages.

SingSong SingSong [29] has demonstrated excellent performance in vocals-to-accompaniment generation. In this work, we reproduce SingSong based on our previous implementation of AudioLM and utilize it as a baseline for the vocals-to-song task. We follow the same setup as SingSong [29], which generates the accompaniment based on the vocals first and then mixes the vocals and accompaniment to produce the complete song. To eliminate the influence of the pre-trained latent diffusion model, we also directly use it to convert the semantic tokens predicted by SingSong into audio without requiring an additional acoustic modeling stage. This new baseline is named SingSong (BEST-RQ).

Additionally, we established two baselines for the accompaniment-to-song task by concatenating the lyrics as another condition before the semantic token sequence of accompaniment. In this setup, the prediction target of this model is the semantic tokens of vocals in the song.

D The editing dataset

We performed insertion, deletion or substitution operations on the original lyrics, with editing spans ranging from 1 to 15 words. Examples of the song editing dataset are shown in table 14.

[8]Implemented based on: https://github.com/lucidrains/musiclm-pytorch.

[9]Implemented based on: https://github.com/facebookresearch/audiocraft.

[10]Implemented based on: https://github.com/lifeiteng/vall-e.

[11]Implemented based on: https://github.com/lucidrains/audiolm-pytorch.

Table 14: Examples of the song editing dataset.

Edit Types	Original Lyrics	Edited Lyrics
insertion	Will you let me know. If you have the key.	Will you let me know. If you **truly hold the answer to my heart,** have the key.
deletion	Was it something that i said? I tried my best, yeah, all for you. **I can see it in your eyes,** the way you lie, I'm such a fool;	Was it something that I said? I tried my best, yeah, all for you. The way you lie, I'm such a fool.
substitution	Cause you had your chance, but **he chose her first. It must be time, it must be time** that heals. You must need time.	Cause you had your chance, but **she caught his eye before you. Seems like fate, seems like fate** that heals. you must need time.
substitution	**I can remember the feeling of being small. Praying** to a god I don't believe in.	**Feeling so tiny, whispering** to a god I don't believe in.
substitution	Love you forever, that's for sure. **And promise, I'll keep you warm.** Our love, shines bright like the sun.	Love you forever, that's for sure. **Vow to hold you close, provide comfort through every storm, warm.** Our love, shines bright like the sun.
deletion	I don't wanna be adored I wanna adore you. **You're in a car driving home.** Leaving oceans and mountains between us, oh no.	I don't wanna be adored I wanna adore you. Leaving oceans and mountains between us, oh no.
insertion	Aa-aa-ah. Losing all emotions, I didn't feel nothing. In need of attention but nobody's looking.	Aa-aa-ah. Losing all emotions, I didn't feel nothing. in need of **a sign, a glance, just something to show me I'm seen** of attention but nobody's looking.
substitution	I'm take a sad girl, turn into a **bad girl**. Even at your worst, you're better than the rest.	I'm take a sad girl, turn into a **queen**. Even at your worst, you're better than the rest.

E Results of the AB preference test

Table 15: Results of the AB preference test between SongCreator and Jukebox in lyrics-to-song. N/P denotes "no preference".

SongCreator	**Jukebox**	**N/P**
60%	38.5%	1.5%

Table 16: Results of the AB preference test between SongCreator and Singsong in vocals-to-song. N/P denotes "no preference".

SongCreator	**SingSong**	**N/P**
30%	54.1%	15.9%

Table 17: Results of the AB preference test between SongCreator and the Ground Truth in different edit tasks. N/P denotes “no preference".

Tasks	SongCreator	Ground Truth	N/P
Song Editing	48%	40%	12%
Vocal Editing	53%	33%	14%
Vocal Editing in Song	32%	52%	16%

Table 18: Results of the AB preference test for using different attention mask strategies in BAC on the lyrics-to-song task.

BR	A2V	V2A	None	N/P
76%	20%	-	-	4%
71%	-	25%	-	4%
85%	-	-	14%	1%

Table 19: Results of the AB preference test for using different attention mask strategies in BAC on the Accompaniment-to-song task.

BR	A2V	N/P
27%	59%	14%

F Results of the inference speed

Table 20: The real-time factor (RTF) for different models on a single NVIDIA V100 GPU with a batch size of 1 during inference.

Model	RTF
MusicLM	14.545
MusicGen	2.104
GPT	1.525
GPT (Vocals & Song)	3.059
SongCreator	2.793

To evaluate the inference speed, we supplement the evaluation by comparing the real-time factor (RTF) for SongCreator and other baselines. RTF represents the time (in seconds) required for the system to synthesize one second of waveform. The evaluation was performed on a single NVIDIA V100 GPU with a batch size of 1. We randomly selected 20 generated audio samples, each longer than 20 seconds, to conduct the evaluation.

As shown in Table 20, the results indicate that methods utilizing a single LM module are significantly faster than MusicLM, which employs multiple LMs in cascading manner. Taking into account the experiments corresponding to Table 3, we observe that although GPT and MusicGen, which only model the song token sequence, are faster than GPT (Vocals & Song) and SongCreator, which predict multiple sequences, this gain in speed comes at the cost of reduced performance. In comparison to GPT (Vocals & Song), our proposed SongCreator, which leverages DSLM to simultaneously model both vocals and accompaniment, achieves not only faster speeds but also better results.

G Detailed experimental settings

G.1 Details in objective evaluations

Here, we provide details of the objective evaluations.

FAD Fréchet Audio Distance (FAD) [67] is used to evaluate the generation fidelity of music. We calculate FAD based on the distribution distance between the feature of the target an generated audios, extracted from VGGish [76] model.

MCD Mel-cepstral distortion (MCD) is a signal-level quality metrics derived from human auditory research, which measures the spectral distance between the synthesized and reference mel-spectrum features. In our research, we attempt to use it to indicate the distance between the generated song and the Ground Truth.

SECS Speaker Embedding Cosine Similarity (SECS) is a widely used metrics in the speech generation, employed to evaluate the similarity of speaker identify. We use the speaker encoder of the Resemblyzer package[12] to compute the SECS between the prompt vocals and synthesized vocals.

G.2 Details in subjective evaluations

For lyrics-to-song and lyrics-to-vocals, we focus on the musicality and quality of the generated songs. We conducted MOS (Mean Opinion Score) tests for both aspects, providing subjects with detailed descriptions, and report both mean and CI95 scores of our MOS tests. In these tests, subjects are specifically asked to focus on the musicality and quality of the song in each respective test. The subjects present and rate the samples, and each subject is asked to evaluate the subjective musicality and quality on a 1-5 scale.

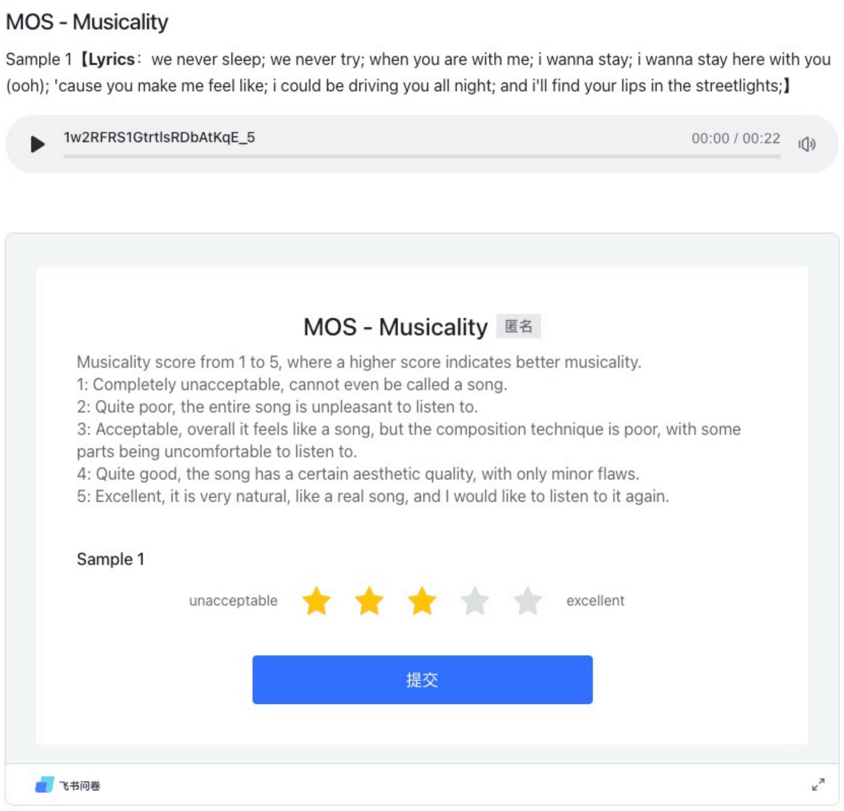
For the prompt-based lyrics-to-song, prompt-based lyrics-to-vocals and music continuation, in addition to musicality, we also asked subjects to focus on the similarity between the generated vocals and accompaniment (if present) to the provided reference audio. In this evaluation, subjects are instructed to ignored the differences in content and audio quality, and to evaluate how well the synthesized results matched the reference audio.

For the prompt-based vocals-to-song and accompaniment-to-song, in addition to musicality, we follow SingSong [29] to ask subjects to focus on the harmony between the vocals and accompaniment. We write explicit instructions to ask the subjects to assess the generated song.

For song editing and vocals editing, in addition to musicality, we also conduct naturalness MOS. This test is aimed to make subjects judge whether the audio appears to have been edited based on its naturalness. In addition, AB preference test is also conducted to ask subjects to give their preferences between a pair of songs.

Our MOS tests are crowd-sourced and conducted by 25 listening subjects, while the AB preference tests are conducted by 20 listening subjects. All the screenshots of instruction for subjects have been shown in Figure 5-11. We paid \$10 to subjects hourly and totally spent about \$600 on participant compensation. We tell the subjects that the data will be used in scientific research.

[12]Implemented based on: https://github.com/resemble-ai/Resemblyzer


MOS - Musicality

Sample 1【**Lyrics**：we never sleep; we never try; when you are with me; i wanna stay; i wanna stay here with you (ooh); 'cause you make me feel like; i could be driving you all night; and i'll find your lips in the streetlights;】

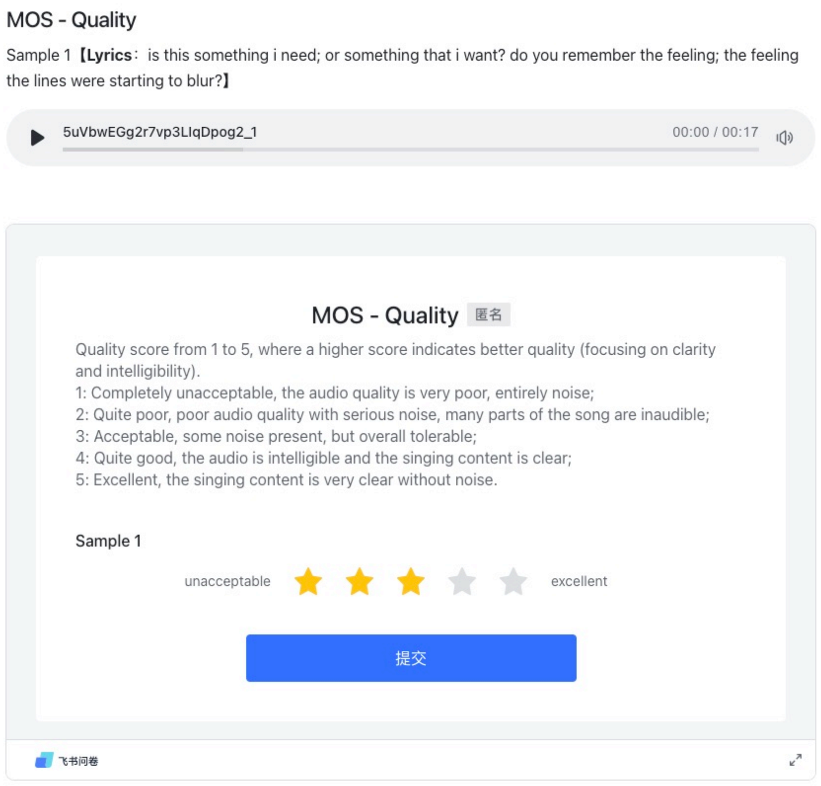
MOS - Musicality 匿名

Musicality score from 1 to 5, where a higher score indicates better musicality.
1: Completely unacceptable, cannot even be called a song.
2: Quite poor, the entire song is unpleasant to listen to.
3: Acceptable, overall it feels like a song, but the composition technique is poor, with some parts being uncomfortable to listen to.
4: Quite good, the song has a certain aesthetic quality, with only minor flaws.
5: Excellent, it is very natural, like a real song, and I would like to listen to it again.

Sample 1

unacceptable ★★★☆☆ excellent

Figure 5: The screenshot of MOS test in musicality evaluation.


MOS - Quality

Sample 1【**Lyrics**：is this something i need; or something that i want? do you remember the feeling; the feeling the lines were starting to blur?】

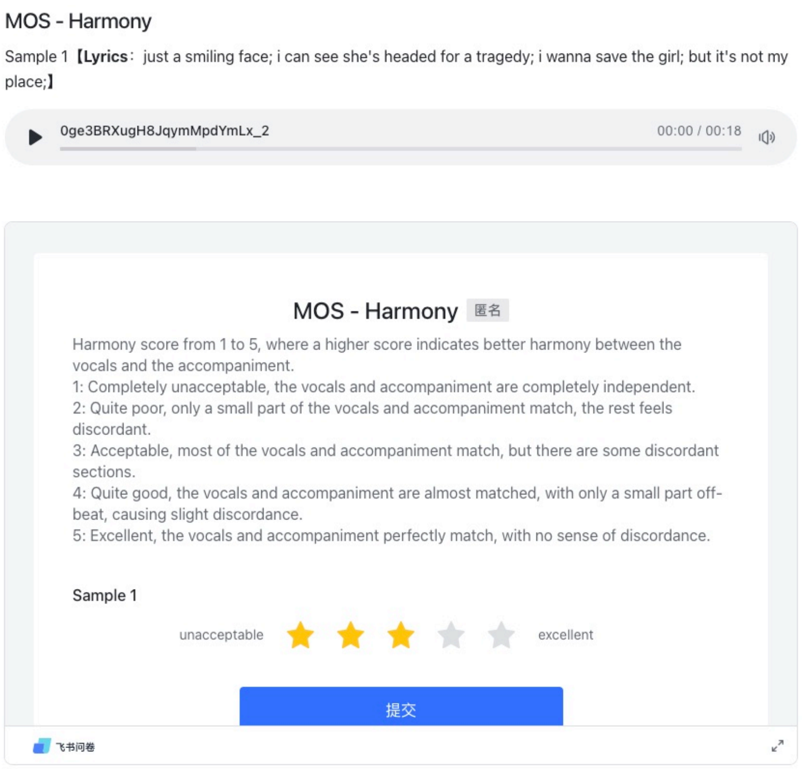
MOS - Quality 匿名

Quality score from 1 to 5, where a higher score indicates better quality (focusing on clarity and intelligibility).
1: Completely unacceptable, the audio quality is very poor, entirely noise;
2: Quite poor, poor audio quality with serious noise, many parts of the song are inaudible;
3: Acceptable, some noise present, but overall tolerable;
4: Quite good, the audio is intelligible and the singing content is clear;
5: Excellent, the singing content is very clear without noise.

Sample 1

unacceptable ★★★☆☆ excellent

Figure 6: The screenshot of MOS test in sound quality evaluation.



MOS - Harmony

Sample 1【**Lyrics**：just a smiling face; i can see she's headed for a tragedy; i wanna save the girl; but it's not my place;】

0ge3BRXugH8JqymMpdYmLx_2 00:00 / 00:18

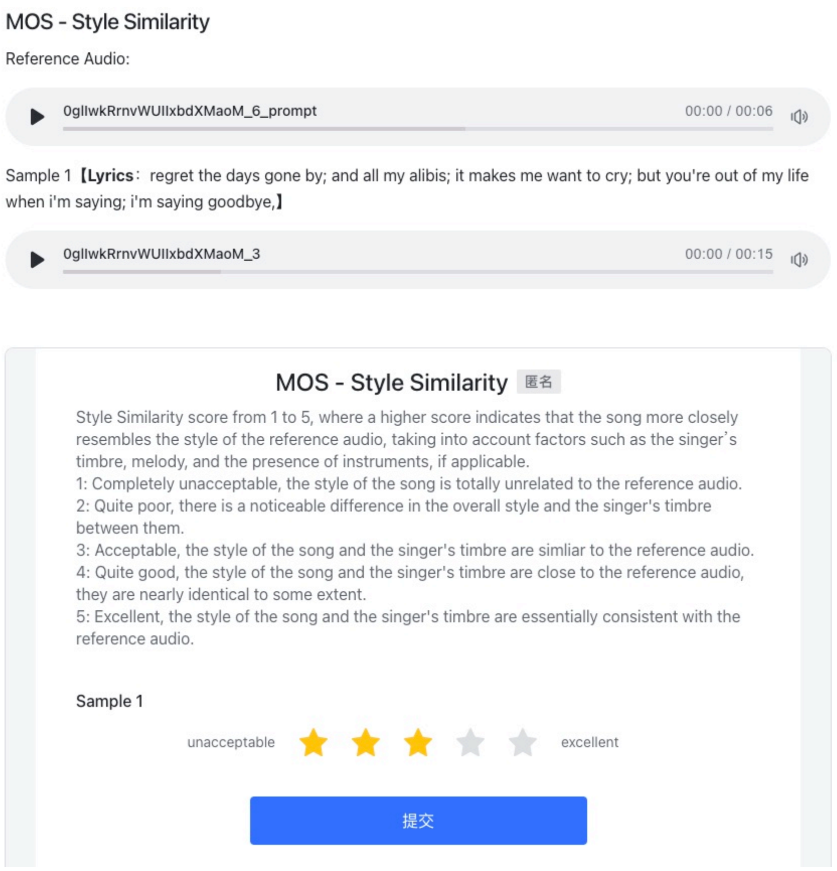
MOS - Harmony 匿名

Harmony score from 1 to 5, where a higher score indicates better harmony between the vocals and the accompaniment.
1: Completely unacceptable, the vocals and accompaniment are completely independent.
2: Quite poor, only a small part of the vocals and accompaniment match, the rest feels discordant.
3: Acceptable, most of the vocals and accompaniment match, but there are some discordant sections.
4: Quite good, the vocals and accompaniment are almost matched, with only a small part off-beat, causing slight discordance.
5: Excellent, the vocals and accompaniment perfectly match, with no sense of discordance.

Sample 1

unacceptable ★★★☆☆ excellent

Figure 7: The screenshot of MOS test in harmony evaluation.



MOS - Style Similarity

Reference Audio:

0gllwkRrnvWUIlxbdXMaoM_6_prompt 00:00 / 00:06

Sample 1【**Lyrics**：regret the days gone by; and all my alibis; it makes me want to cry; but you're out of my life when i'm saying; i'm saying goodbye,】

0gllwkRrnvWUIlxbdXMaoM_3 00:00 / 00:15

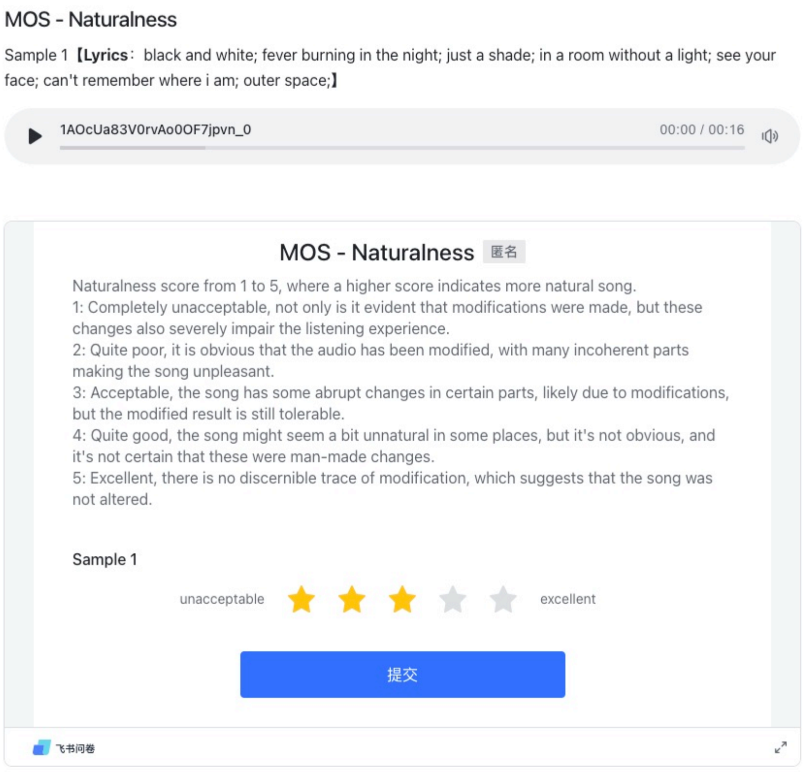
MOS - Style Similarity 匿名

Style Similarity score from 1 to 5, where a higher score indicates that the song more closely resembles the style of the reference audio, taking into account factors such as the singer's timbre, melody, and the presence of instruments, if applicable.
1: Completely unacceptable, the style of the song is totally unrelated to the reference audio.
2: Quite poor, there is a noticeable difference in the overall style and the singer's timbre between them.
3: Acceptable, the style of the song and the singer's timbre are simliar to the reference audio.
4: Quite good, the style of the song and the singer's timbre are close to the reference audio, they are nearly identical to some extent.
5: Excellent, the style of the song and the singer's timbre are essentially consistent with the reference audio.

Sample 1

unacceptable ★★★☆☆ excellent

Figure 8: The screenshot of MOS test in similarity evaluation.


MOS - Naturalness

Sample 1【**Lyrics**：black and white; fever burning in the night; just a shade; in a room without a light; see your face; can't remember where i am; outer space;】

1AOcUa83V0rvAo0OF7jpvn_0 00:00 / 00:16

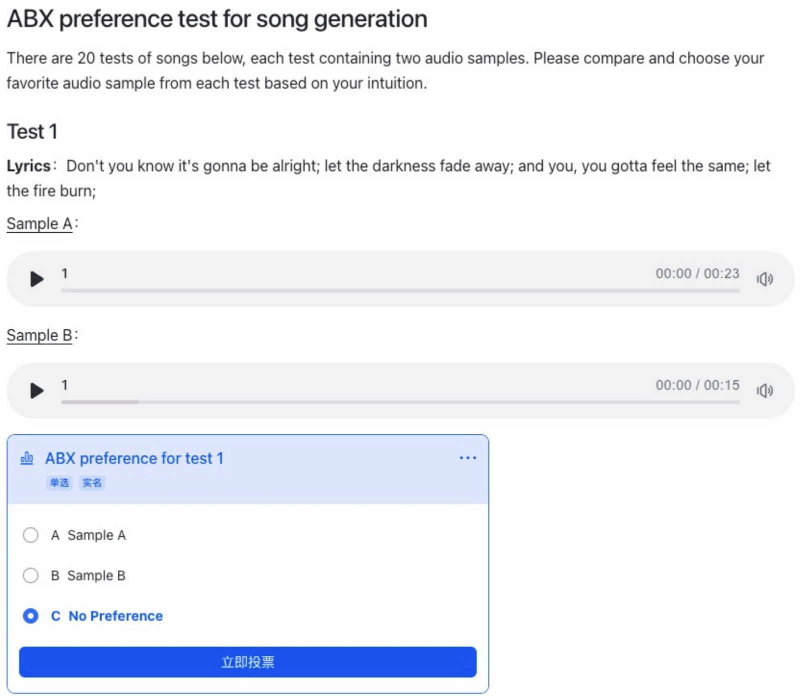
MOS - Naturalness 匿名

Naturalness score from 1 to 5, where a higher score indicates more natural song.
1: Completely unacceptable, not only is it evident that modifications were made, but these changes also severely impair the listening experience.
2: Quite poor, it is obvious that the audio has been modified, with many incoherent parts making the song unpleasant.
3: Acceptable, the song has some abrupt changes in certain parts, likely due to modifications, but the modified result is still tolerable.
4: Quite good, the song might seem a bit unnatural in some places, but it's not obvious, and it's not certain that these were man-made changes.
5: Excellent, there is no discernible trace of modification, which suggests that the song was not altered.

Sample 1

unacceptable ★★★☆☆ excellent

Figure 9: The screenshot of MOS test in naturalness evaluation.


ABX preference test for song generation

There are 20 tests of songs below, each test containing two audio samples. Please compare and choose your favorite audio sample from each test based on your intuition.

Test 1

Lyrics：Don't you know it's gonna be alright; let the darkness fade away; and you, you gotta feel the same; let the fire burn;

Sample A：

1 00:00 / 00:23

Sample B：

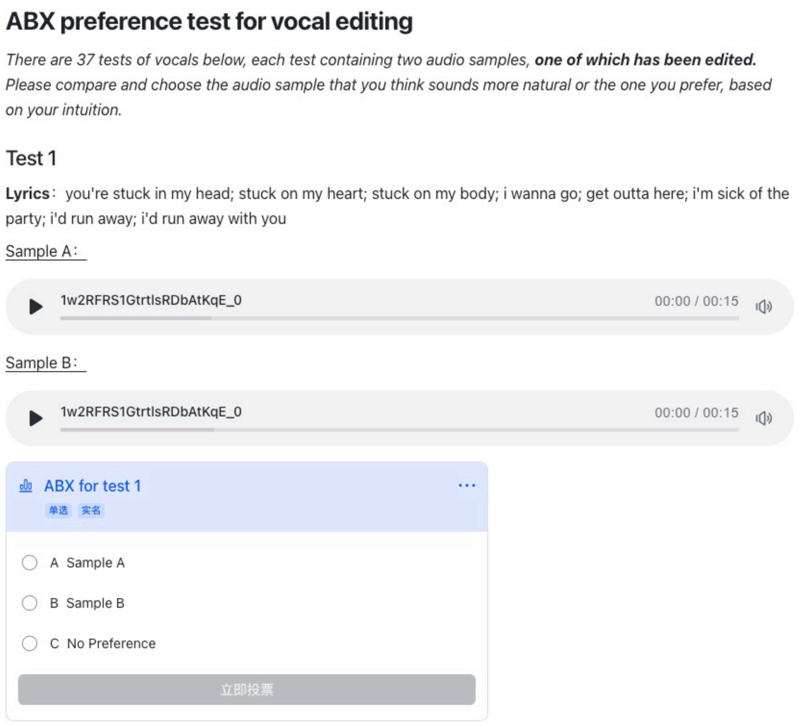
1 00:00 / 00:15

ABX preference for test 1

单选 实名

A Sample A
B Sample B
[✓ C No Preference]

Figure 10: The screenshot of AB preference test.



ABX preference test for vocal editing

There are 37 tests of vocals below, each test containing two audio samples, ***one of which has been edited.*** *Please compare and choose the audio sample that you think sounds more natural or the one you prefer, based on your intuition.*

Test 1

Lyrics: you're stuck in my head; stuck on my heart; stuck on my body; i wanna go; get outta here; i'm sick of the party; i'd run away; i'd run away with you

Sample A:

1w2RFRS1GtrtlsRDbAtKqE_0 00:00 / 00:15

Sample B:

1w2RFRS1GtrtlsRDbAtKqE_0 00:00 / 00:15

ABX for test 1

单选 实名

A Sample A

B Sample B

C No Preference

立即投票

Figure 11: The screenshot of AB preference test.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have ensured that the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have thoroughly discussed the limitations of our work in Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have clearly and comprehensively described the architecture and training strategy of our model in Section 3, and provided detailed hyperparameters in Appendix A to facilitate replication of the model.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We have provided instructions on data access and preparation in Section 4.1 and Appendix A.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have provided all the training and test details in Section 4.1 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have reported 95% confidence intervals (CI95) and the results of statistical significance tests for the experiments that support the main claims of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our experiments were conducted using GPUs, and detailed information about the computational resources is provided in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that our research conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed both potential positive societal impacts and negative societal impacts of the work performed in Section 5

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets used in the paper have been properly credited, and the original papers have been cited or URLs have been provided.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We conducted crowdsourcing experiments, and in Appendix G.2, we have provided the full text of the instructions given to subjects along with screenshots. Details about compensation are also included.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: We disclosed all potential risks to the subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.