DiffRhythm: Blazingly Fast and Embarrassingly Simple End-to-End Full-Length Song Generation with Latent Diffusion

Anonymous ACL submission

Abstract

Recent advancements in music generation have garnered significant attention, yet existing approaches face critical limitations. Some current generative models can only synthesize either the vocal track or the accompaniment track. While some models can generate combined vocal and accompaniment, they typically rely on meticulously designed multi-stage cascading architectures and intricate data pipelines, hindering scalability. Additionally, most systems are restricted to generating short musical segments rather than full-length songs. Furthermore, widely used language model-based methods suffer from slow inference speeds. To address these challenges, we propose DiffRhythm, the first latent diffusion-based song generation model capable of synthesizing complete songs with both vocal and accompaniment for durations of up to 4m45s in only ten seconds, maintaining high musicality and intelligibility. Despite its remarkable capabilities, DiffRhythm is designed to be simple and elegant: it eliminates the need for complex data preparation, employs a straightforward model structure, and requires only lyrics and a style prompt during inference. Additionally, its non-autoregressive structure ensures fast inference speeds. This simplicity guarantees the scalability of DiffRhythm. Moreover, we release the complete training code along with the pre-trained model on large-scale data to promote reproducibility and further research[1].

1 Introduction

Music, as a form of artistic expression, holds profound cultural importance and resonates deeply with human experiences (Briot et al., 2017). The field of music generation has witnessed remarkable advancements in recent years, driven by innovations in deep learning, particularly the deep generative models.

While these models have shown promise, they often exhibit critical limitations that restrict their practical applicability. Many existing approaches are designed to generate vocal tracks and accompaniment tracks independently, resulting in a disjointed musical experience. For instance, studies such as Melodist (Hong et al., 2024) and MelodyLM (Li et al., 2024a) demonstrate the effectiveness of isolated track generation, yet highlighting the need for more holistic solutions that capture the interplay between vocals and accompaniment.

Currently, there are relatively few studies on end-to-end song generation in the academic field. State-of-the-art platforms like Seed-Music (Bai et al., 2024) and Suno[2] are generally for commercial products and provides no open-source implementation or detailed technical documentation.

Recent academic work such as SongCreator (Lei et al., 2024) and SongEditor (Yang et al., 2024) endeavor to create combined vocal and accompaniment outputs; however, these typically rely on complex, multi-stage cascading architectures. This complexity not only complicates design and implementation but also limits scalability, particularly for longer audio synthesis where maintaining consistency is challenging. The ability to generate complete compositions is essential for practical applications in both artistic creation and commercial music production. Moreover, most existing music generation models follow a language model paradigm (Hong et al., 2024; Li et al., 2024a; Yang et al., 2024; Agostinelli et al., 2023), often struggling with slow inference speeds, which hinder real-time applications and user interactivity.

To address these challenges, we present DiffRhythm, the first full-diffusion-based song generation model that is capable of synthesizing full-length songs comprising both vocal and accompaniment for durations of up to four minutes.

[1] https://anonymous.4open.science/w/DiffRhythm-3EBC/

[2] https://suno.com/

DiffRhythm distinguishes itself not only through its ability to maintain high levels of musicality and intelligibility but also through its simple yet effective model architecture and data processing pipeline, designed specifically for scalability. Additionally, our non-autoregressive approach allows for fast generation speeds, significantly improving usability compared to current models. The main contributions of this paper are summarized as follows:

- We propose DiffRhythm, the first end-to-end diffusion-based song generation model capable of generating full song with both vocal and accompaniment.
- We propose a sentence-level lyrics alignment mechanism for better vocal intelligibility, which tackles ultra-sparse lyrics-vocal alignment with minimal supervision.
- We train a Variational Autoencoder (VAE) tailored for high-fidelity music reconstruction,while demonstrating exceptional robustness against MP3 compression artifacts. Moreover, our VAE shares the same latent space with the famous Stable Audio VAE[3], enabling seamless plug-and-play substitution in existing latent diffusion frameworks.
- Our experiments show that despite its simpleness, DiffRhythm achieves excellent performance in song generation. The data processing pipeline, pretrained models trained on large-scale datasets, and the complete training recipe are publicly available.

2 Related Work

2.1 Vocal Generation

Early models for vocal generation, or singing voice generation, focused on synthesizing natural singing voices based on lyrics, musical scores, and corresponding durations. VISinger 2 (Zhang et al., 2023) introduces an end-to-end system utilizing a digital signal processing (DSP) synthesizer to enhance sound quality. StyleSinger (Zhang et al., 2024) employ a reference voice clip for timbre and style extraction, enabling style transfer and zero-shot synthesis. PromptSinger (Wang et al., 2024a) was the first system to attempt guiding singing voice generation through text descriptions, placing greater emphasis on timbre control. DiffSinger (Liu et al., 2022) addresses the issue of excessive smoothness by implementing a shallow diffusion mechanism. To bridge the gap between realistic music scores and detailed MIDI annotations, RMSSinger (He et al., 2023) proposes a word-level modeling approach combined with diffusion-based pitch prediction. MIDI-Voice (Byun et al., 2024) incorporates MIDI-based priors for expressive zero-shot generation. VoiceTuner (Huang et al., 2024) advocates a self-supervised pre-training and fine-tuning strategy to mitigate data scarcity, applicable to low-resource SVS tasks. There are also recent models that do not rely on strict music score and duration annotations, such as Freestyler (Ning et al., 2024), which takes lyrics and accompaniment as inputs to generate rapping vocals with strong stylistic and rhythmic alignment with accompanying beats.

2.2 Music Generation

Music generation encompasses various tasks, including symbolic music generation, lyrics generation, and accompaniment generation. MuseGAN (Dong et al., 2018) achieves symbolic music generation through a GAN-based approach. SongMASS (Sheng et al., 2021) designs a method for songwriting that generates lyrics or melodies conditioned on each other, while SongComposer (Ding et al., 2024) proposes a large language model (LLM) for song composition, capable of generating melodies and lyrics with symbolic song representations. DeepRapper (Xue et al., 2021) focuses on rap lyrics generation, which also leverages an LLM to generate lyrics from right to left with rhyme constraints. Inspired by two-stage modeling in audio generation (Borsos et al., 2023), MusicLM (Agostinelli et al., 2023) uses a cascade of transformer decoders to sequentially generate semantic and acoustic tokens, based on joint textual-music representations from MuLan (Huang et al., 2022). MusicGen (Copet et al., 2023) introduces a novel approach with codebook interleaving patterns to generate music codec tokens in a single transformer decoder, which is further combined with stack patterns in Le Lan et al., 2024 to improve generation quality. Additionally, MeLoDy (Lam et al., 2023) presents an LM-guided diffusion model that efficiently generates music audio, and MusicLDM (Chen et al., 2024a) incorporates beat-tracking information and latent mixup data augmentation to address potential plagiarism issues in music generation. Several works focus specifically on vocal-to-

[3]https://github.com/Stability-AI/stable-audio-tools

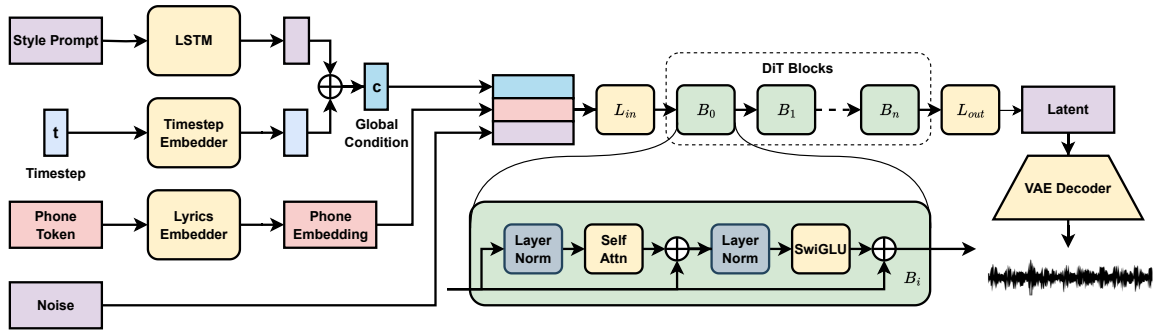


Figure 1: Architecture of DiffRhythm. The style and lyrics are used as external control signals, which are preprocessed to get the style embedding and lyrics token, input to DiT to generate latent, and subsequently go through the VAE decoder to generate the audio.

accompaniment generation, such as SingSong (Li et al., 2024b), which generates instrumental music to accompany input vocals, and Melodist (Hong et al., 2024), which utilizes a transformer decoder for controllable accompaniment generation.

2.3 Song Generation

Song generation models aim to produce natural singing voices accompanied by music. Song generation incorporates elements from both vocal and music generation. A common methodology in song generation employs a two-stage process: initially generating the vocal track from lyrical input, followed by the prediction of accompanying music. Melodist (Hong et al., 2024) utilizes two autoregressive transformers to sequentially produce vocal and accompaniment codec tokens, conditioned on lyrics, musical scores, and natural language prompts. MelodyLM (Li et al., 2024a) eliminates the need for music scores in Melodist and instead relies solely on textual descriptions and vocal references. However, given the intricate relationship between vocals and accompaniment, sequential generation may not be optimal. Different from Melodist and MelodyLM, SongCreator (Lei et al., 2024) simultaneously generates vocal and accompaniment, while SongEditor (Yang et al., 2024) also offering flexible song editing capabilities. It is noteworthy that these models predominantly utilize language model-based architectures. While effective, their autoregressive nature introduces significant computational overhead and challenges in maintaining consistent style and rhythm over long sequences.

3 DiffRhythm

To address the limitations of existing approaches and overcome the challenges in full-length song generation, we present DiffRhythm - the first full-diffusion-based model specifically designed for end-to-end song generation.

3.1 Overview

DiffRhythm produces full-length stereo musical compositions (up to 4m 45s) at 44.1kHz sampling rate, guided by lyrics and style prompts. The architecture consists of two consecutively trained models : 1) A variational autoencoder (VAE) that learns compact latent representations of waveforms while preserving perceptual audio details, effectively resolving the sequence length constraints in raw audio modeling; 2) A diffusion transformer (DiT) operating in the learned latent space that generates songs through iterative denoising. Compared with conventional discrete tokens in LM-based approaches, our continuous latent representation captures richer music details and vocal nuances, enabling high-fidelity audio reconstruction. Meanwhile, the DiT's strong modeling capabilities and the reduced sequence length of continuous VAE latents ensure superior long-term musical structure consistency and vocal intelligibility across full-length songs.

Furthermore, to tackle the critical challenge of lyric-vocal alignment in full-song generation, we propose a novel sentence-level alignment mechanism to establish semantic correspondence between dense lyrical content and sparse singing vocals.

3.2 Variational Autoencoder

To lower the computational demands of training the diffusion model towards long-form high-quality song generation, we first train an autoencoding model which learns a latent space that is perceptually equivalent to the audio space, but offers significantly reduced computational complexity.

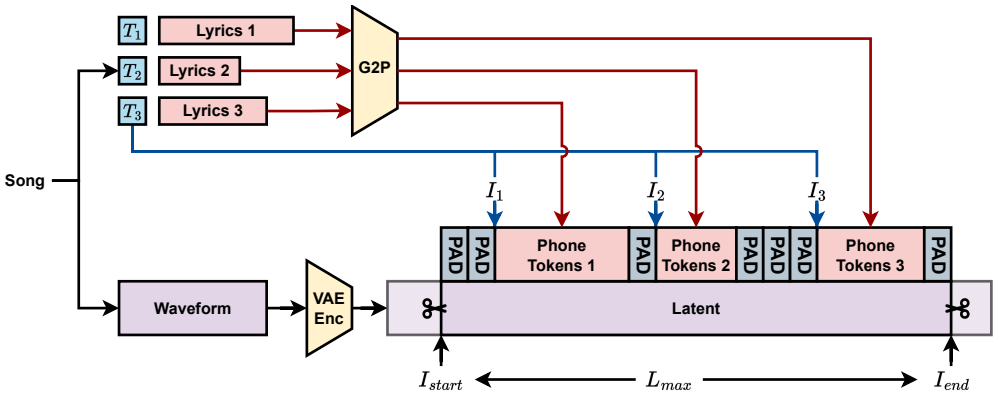


Figure 2: The data preprocessing pipeline of DiffRhythm. Lyrics go through G2P and are placed at the positions corresponding to their timestamps

Model Backbone The backbone of the autoencoder is fully-convolutional that allows the compression and reconstruction of full-songs with arbitrary-length. The encoder and decoder structures are taken from Stable Audio 2 (Evans et al., 2024b). Given a raw stereo waveform $y \in \mathbb{R}^{T\times 2}$, the encoder $\mathcal{E}$ encodes y into a latent representation $z = \mathcal{E}(y)$, the decoder $\mathcal{D}$ reconstructs the song from the latent, giving $\hat{y} = \mathcal{D}(z) = \mathcal{D}(\mathcal{E}(y))$, where $z \in R^{L\times c}$. The encoder downsamples the audio by a factor $f = T/L$.

Training Objectives The VAE is optimized through a composite loss function integrating spectral reconstruction and adversarial training components. The primary training objective combines a multi-resolution STFT loss (Steinmetz and Reiss, 2020) with perceptual weighting, specifically designed for stereo signal processing. To address potential ambiguities in spatial localization, we compute this loss in both mid-side (M/S) decomposition and individual left/right channel domains, with the latter contribution scaled by 0.5 relative to the M/S term.

Complementing this reconstruction objective, we implement an adversarial training scheme using a convolution-based discriminator (Défossez et al., 2023). While maintaining hyperparameters with Stable Audio (Evans et al., 2024a), the discriminator features substantially expanded channel dimensions, resulting in approximately quadrupled parameter count compared to the original implementation. This enhancement aims to improve the model's capacity for capturing high-frequency audio details through more discriminative feature learning.

Lossy-to-Lossless Reconstruction Considering that a large amount of song data exists in compressed MP3 format, where high-frequency components are compromised during compression, we employ data augmentation to equip the VAE with restoration capabilities. Specifically, the VAE is trained exclusively on lossless FLAC-format data, where the input undergoes MP3 compression while the reconstruction target remains the original lossless data. Through this lossy-to-lossless reconstruction process, the VAE learns to decode latent representations derived from lossy-compressed data back into lossless audio signals.

Latent Truncation for Training As illustrated in Figure 2, for diffusion training, we randomly sample a starting frame index I_{start} and truncate z from I_{start} to a feature length of L_{max} for batch consistency. Another small segment of latent is also randomly selected and used as the style prompt to provide style information. Specific length configurations are detailed in Section 4.

3.3 Diffusion Transformer

With compact latent features extracted by the VAE encoder as intermediate representations, we adopt the widely used diffusion transformer (DiT) for lyrics-to-latent generation. DiT has seen notable success in other modalities (Peebles and Xie, 2023; Esser et al., 2024), and has recently been applied to text-to-speech (Liu et al., 2024; Eskimez et al., 2024; Chen et al., 2024b) and music generation (Evans et al., 2024b; Fei et al., 2024; Hung et al., 2024).

Feature Conditioning As shown in Figure 1, DiT is conditioned by three features: A style prompt for controlling song style, a timestep indicating the current diffusion step, and lyrics for vocal content control. The style prompt goes through a Long Short-Term Memory (LSTM) net-

work, where the final hidden state is extracted as the global style information. This information is then summed with the time-step embedding to form a global condition feature. The phone tokens of the lyrics undergo processing through an embedding layer to produce continuous phoneme embeddings. Following this, latent representations undergo noise addition to get noised latent. These three features are concatenated along the channel dimension to serve as inputs to DiT. The feature extraction process will be detailed in Sec. 3.4.

Model Backbone Different from the original DiT implementation (Peebles and Xie, 2023), DiT in DiffRhythm incorporates stacks of LLaMA decoder layers. Given that LLaMA is widely used in natural language processing (NLP), several readily available acceleration libraries, such as Unsloth[4] and Liger-Kernel[5], that can easily achieve more than 25% training and inference speed-ups relative to the original DiT without any performance degradation through kernel fusion. We employ efficient FlashAttention2 (Dao, 2024) and gradient checkpointing (Chen et al., 2016) to reduce the computational and memory impact of applying a transformer architecture over longer sequences. These techniques are essential for the effective training of models with extensive context lengths.

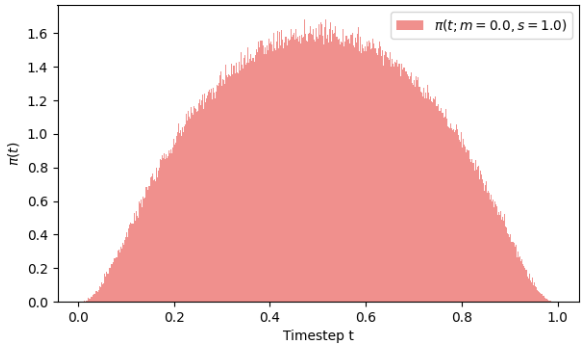


Figure 3: Logit-normal timestep distribution.

Training Objectives Following the conditional flow matching paradigm (Lipman et al., 2023), our model learns a velocity field $v_\theta(z_t, t)$ that transports the noise distribution $p_0(z)$ to the data distribution $p_1(z)$ through the ODE:

$$\frac{dz_t}{dt} = v_\theta(z_t, t) \quad \text{with} \quad \begin{cases} z_0 \sim p_0(z) \\ z_1 \sim p_1(z) \end{cases} \tag{1}$$

[4]https://github.com/unslothai/unsloth
[5]https://github.com/linkedin/Liger-Kernel

The training objective minimizes the expected squared error between predicted and target velocity fields:

$$\mathcal{L} = \mathbb{E}_{t\sim\pi_{\ln}, z_t\sim p_t(z_t)} \left[\| v_\theta(z_t, t, c) - (z_1 - z_0) \|_2^2 \right], \tag{2}$$

where c is the condition, and the timestep sampling distribution $\pi_{\ln}(t; m, s)$ follows the logit-normal density:

$$\pi_{\ln}(t; m, s) = \frac{1}{s\sqrt{2\pi}} \frac{1}{t(1-t)} \exp\left(-\frac{(\text{logit}(t) - m)^2}{2s^2} \right), \tag{3}$$

with $\text{logit}(t) = \log \frac{t}{1-t}$. As discussed in Stable Diffusion 3 (Esser et al., 2024), logit-normal sampling provides adaptive weighting where the scale parameter s controls concentration around midpoint timesteps (challenging prediction regions), while the location parameter m enables bias toward either data ($m < 0$) or noise ($m > 0$) domains. This allows training to focus more effectively on complex intermediate regions. In practice, we sample $u \sim \mathcal{N}(m, s)$ and map it through the logistic function $t = \sigma(u) = 1/(1 + e^{-u})$. Figure 3 illustrates the timestep distribution when $m = 0$ and $s = 1$.

3.4 Lyrics-to-Latent Alignment

Song generation, which necessitates the creation of intelligible vocal content, presents unique alignment challenges beyond conventional text-to-speech (TTS) task. While TTS models typically handle shorter speech segments (usually less than 30 seconds) with continuous articulation, vocal generation must address two critical alignment problems:

(1) *Discontinuous temporal correspondence*: Vocal segments are often separated by prolonged instrumental intervals, creating phonetic discontinuity that disrupt conventional temporal alignment mechanisms.

(2) *Accompaniment interference*: As we target to simultaneously model voice and accompaniment, the same words, although corresponding to the same pronunciation, have different accompaniment in different songs, which brings more difficulty in aligning.

With conventional text conditioning approaches in diffusion-based TTS models like cross-attention mechanisms or direct feature concatenation (Eskimez et al., 2024; Chen et al., 2024b), we failed to achieve intelligibility in song generation.

Table 1: Comparative evaluation of waveform reconstruction performance using objective metrics. STOI, PESQ, and MCD scores are reported for both lossless-to-lossless and lossy-to-lossless reconstruction.

	Lossless → Lossless			Lossy → Lossless					
	STOI↑	PESQ↑	MCD↓	STOI↑	PESQ↑	MCD↓	Sampling Rate	Frame Rate	Latent Channels
Music2Latent	0.584	1.448	8.796	-	-	-		10 Hz	
Stable Audio 2 VAE	0.621	1.96	8.033	-	-	-	44.1 kHz	21.5 Hz	64
DiffRhythm VAE	**0.646**	**2.235**	**8.024**	**0.639**	**2.191**	**9.319**		21.5 Hz	

It is relatively challenging for the model to tackle both tasks simultaneously. Therefore, we aim to reduce the difficulty of alignment, allowing the model to focus more on the second challenge. To achieve this, we propose a sentence-level alignment paradigm that requires only sentence-start annotations. Given lyric sentences with timestamp annotations $(t_i^{start}, s_i)_{i=1}^{N}$, we first convert each lyric sentence s_i into a phoneme sequence $\mathbf{p}_i \in \mathcal{V}^{L_i}$ through grapheme-to-phoneme (G2P) conversion, where $\mathcal{V}$ denotes the phoneme vocabulary and L_i denotes the sequence length of s_i. Next, we initialize a latent-aligned sequence $\mathbf{P}_i = [\langle\text{pad}\rangle]^{L_{max}}$ with the same length as the latent representation. Then, for each phoneme sequence $\mathbf{p}_i = [p_1, \ldots, p_{L_i}]$, we overwrite the corresponding section of $\mathbf{P}_i$ as follows: $\mathbf{P}_i[f_i^{start} : f_i^{start} + L_i] = \mathbf{p}_i$, $f_i^{start} = \lfloor t_i^{start} \cdot F_s \rfloor$, where F_s denotes the latent frame rate. The whole process is detailed in Figure 2. The proposed approach achieves high intelligibility while minimizing the reliance on supervision, effectively reducing the cost of data labeling processing.

4 Experimental Setup

4.1 Dataset

DiffRhythm was trained on a comprehensive music dataset comprising approximately 1 million songs (totaling 60,000 hours of audio content) with an average duration of 3.8 minutes per track. The dataset features a multilingual composition ratio of 3:6:1 for Chinese songs, English songs, and instrumental music respectively. To ensure lyrical quality, we implemented a simple rule-based lyrics cleaning pipeline that systematically filters out low-quality lyrics. Subsequently we pre-extract the phoneme tokens from lyrics using MaskGCT (Wang et al., 2024b) G2P and song latent using the pre-trained VAE for faster training.

For autoencoder evaluation, we selected 10 representative music genres, sampling three tracks per genre to form a 30-song test set. Five non-overlapping 10-second clips were randomly extracted from each track for analysis. To assess the song generation quality, we reserved 30 songs from the training dataset and generated samples using ground-truth lyrics and style prompts as input conditions.

4.2 Model Configuration

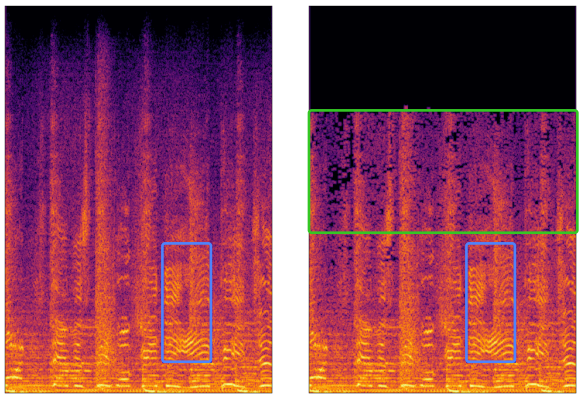
VAE Our implementation adapts the pre-trained weights from Stable Audio 2's VAE with 157M parameters, freezing the encoder while training the decoder for 2.5M iterations on a curated dataset of 250k lossless audio samples. The architecture processes 44.1 kHz stereo audio inputs through 5× downsampling blocks achieving a compression factor of $f = 2048$, yielding 64-dimensional latent representations at 21.5 Hz frame rate. During training, there was a 1/3 probability of keeping the inputs unchanged and a 2/3 probability of applying MP3 compression, with uniformly randomized VBR[6] quality value from 0 to 7. MP3 compression is achieved using pedalboard[7].

DiT Our DiT implementation comprises 16 LLaMA decoder layers[8] with 2048-dimensional hidden size and 32-head self-attention mechanisms (64 dimensions per head), totaling 1.1B parameters. We apply independent 20% dropout to lyrics and style prompts to facilitate classifier-free guidance (CFG) (Ho and Salimans, 2022). The diffusion process employs an Euler ODE solver with 32 steps and CFG scale of 4 during inference. Training occurs in two phases: initial base model training with $L_{max} = 2048$ ($\approx$ 95s), followed by fine-tuning to $L_{max} = 6144$ ($\approx$ 4m45s).

Both models were trained using AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.95$. The learning rate was set to 1×10^{-4} with exponential ramp-up and decay. To ensure model stability and performance, we maintain a secondary copy of the

[6]Variable bit rate, lower values represent higher quality
[7]https://github.com/spotify/pedalboard
[8]https://github.com/huggingface/transformers

model weights, updated every 100 training batches through an exponential moving average (EMA) with a decay rate of 0.99, following the approach outlined by (Peebles and Xie, 2023). All models were trained on 8x Huawei Ascend 910B with fp16 mixed-precision.



(a) GT (Lossless) (b) GT (MP3 Compressed)

(c) Proposed VAE (d) Stable Audio 2 VAE

Figure 4: Visualization of Spectrograms from (a) lossless ground-truth, (b) ground-truth after MP3 compression, (c) MP3 reconstructed by proposed VAE, (d) MP3 reconstructed by Stable Audio VAE. Boxed regions indicate areas to be analyzed in the main text.

4.3 Evaluation Metrics

Objective Evaluation To evaluate the quality of waveform reconstruction, we calculate STOI (Taal et al., 2010), PESQ (Rix et al., 2001) and Mel cepstral distortion (MCD) (Kubichek, 1993). For evaluating the quality song generation, we utilize the Phoneme Error Rate (PER) and Fréchet Audio Distance (FAD) (Kilgour et al., 2019). We employ FireRedASR (Xu et al., 2025), which is currently the state-of-the-art Automatic Speech Recognition (ASR) model, to recognize the vocal content of the generated songs. FireRedASR not only achieves remarkably high performance for vocals but is also robust in recognizing singing vocals. Given that ASR may perceive vocal content as different words with consistent pronunciation, such errors do not accurately reflect actual vocal intelligibility; therefore, we calculate the PER instead of the Word Error Rate (WER) or Character Error Rate (CER). Realtime factor (RTF) is also calculated using Nvidia RTX 4090 to demostrate the computational efficiency of the comparison models.

Subjective Evaluation We conducted mean opinion score (MOS) listening tests for subjectively evaluation. Specifically, 30 listeners participated in rating each generated song sample on a scale from 1 to 5 across three aspects: musicality, quality and intelligibility.

5 Evaluation Results

5.1 Waveform Reconstruction

We conduct a comprehensive evaluation of waveform reconstruction performance comparing our VAE with two popular open-sourced baselines: Music2Latent (Pasini et al., 2024) and Stable Audio 2 (Evans et al., 2024b). The evaluation protocol consists of two experimental settings: (1) lossless-to-lossless reconstruction using lossless audio inputs, and (2) lossy-to-lossless reconstruction using MP3-compressed inputs while maintaining lossless reference targets. As shown in Table 1, the proposed method achieves superior performance across all metrics in both experimental conditions. Specifically, under lossless input conditions, our model demonstrates 3.8% and 12.3% relative improvements in STOI and PESQ respectively over the best baseline, while maintaining comparable MCD scores. More importantly, when processing lossy MP3 inputs - a scenario where baseline models completely fail due to their lack of restoration capability - our method maintains robust performance with only minimal degradation on all three metrics compared to the lossless condition.

To further validate the reconstruction quality, we perform spectral visualization comparing the proposed VAE with baseline models. Figure 4 reveals three key observations: First, MP3 compression artifacts manifest as both high-frequency attenuation (above 32 kHz) and mid-frequency hollowing effects (16 kHz - 32 kHz). Second, our VAE successfully addresses both artifact types - it not only generates missing high-frequency components but also restores the spectral continuity in mid-

Table 2: Objective and subjective evaluation results of comparison and ablation systems for song generation. DiffRhythm-base and DiffRhythm-full represent DiffRhythm with generation length of 1m35s and 4m45s respectively, and w/o align stands for the ablation system without sentence-level alignment.

	PER↓	FAD↓	Musicality↑	Quality↑	Intelligibility↑	Generation Length	RTF↓
GT (VAE-reconstructed)	16.14%	0.88	4.68±0.06	4.43±0.06	4.17±0.03	-	-
SongLM	21.35%	1.92	4.27±0.04	4.06±0.03	3.44±0.03	120 s	1.717
DiffRhythm-base	17.47%	2.11	4.14±0.07	4.19±0.05	3.80±0.04	95 s	0.037
DiffRhythm-full	18.02%	2.25	4.02±0.02	4.21±0.04	3.68±0.07	285 s	0.034
w/o align	-	3.16	4.07±0.05	3.04±0.02	-	95 s	0.037

frequency regions (green box). Third, the proposed model demonstrates superior harmonic reconstruction capability for vocal components, particularly in preserving formant structures, resulting in significantly clearer vocal components compared to the open-source baseline that produces vague harmonics (blue box).

5.2 Song Generation

For the evaluation of song generation, we compare DiffRhythm with SongLM (Yang et al., 2024), the samples of SongLM were kindly provided by the authors. As shown in Table 2, the GT songs reconstructed via VAE naturally achieves the best performance across all metrics, serving as an upper bound for synthesized song quality. Compared to the SongLM baseline, DiffRhythm models achieve superior quality and intelligibility while maintaining comparable musicality. The significant 18.2% relative reduction in PER further confirms our model's improved vocal content clarity. However, SongLM shows slightly better FAD and musicality scores, suggesting room for improvement in long-term acoustic consistency and melodic expression.

The full-length DiffRhythm variant exhibits marginally degraded PER and FAD than its base version, likely due to increased modeling complexity for longer sequences. Notably, both variants maintain RTF below 0.04, achieving a $\sim 50\times$ speedup over SongLM, highlighting the computational efficiency of our diffusion-based approach compared to autoregressive language models.

Our ablation study reveals the critical role of sentence-level alignment. As shown in Table 2, removing this approach catastrophically degrades intelligibility (unmeasurable PER and intelligibility MOS) and audio quality, though interestingly preserves basic musical structure. This validates our hypothesis that sentence-level alignment is essential for establishing semantic correspondence between tight lyrics and vocals.

The relatively high PER across all systems may stem from using mixed audio containing both vocal and accompaniment without source separation for ASR evaluation, as accompaniment likely interferes with ASR recognition.

6 Conclusion

In this paper, we propose DiffRhythm, the first full-diffusion-based model capable of generating complete stereo songs of 4m45s in just 10 seconds, featuring both vocals and accompaniment. The model's elegant design eliminates the need for complex multi-stage cascading modeling and laborious data preprocessing, facilitating scalability. DiffRhythm's non-autoregressive structure ensures rapid inference speeds while preserving high musical quality and lyrical intelligibility. Extensive experimental results demonstrate the effectiveness of our approach and underscore the robust song generation capabilities of DiffRhythm. Furthermore, the system's simplicity and open accessibility—through our release of code and pre-trained models—establish a new foundation for scalable, end-to-end research in song generation.

7 Limitations

While DiffRhythm demonstrates good capability to generate high-quality full-length songs, two important aspects remain unexplored in our current framework. First, the functionality for editing specific segments within generated compositions has not been investigated. Incorporating random masking of latent representations during training could enable song editing (inpainting) and continuation (outpainting). Second, the model employs short audio clips as style references, integrating natural language conditioning mechanisms would enable finer-grained stylistic control through textual descriptions. This improves the flexibility of the model by eliminating the need for audio references.

References

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse H. Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Havnø Frank. 2023. Musiclm: Generating music from text. *CoRR*, abs/2301.11325.

Ye Bai, Haonan Chen, Jitong Chen, Zhuo Chen, Yi Deng, Xiaohong Dong, Lamtharn Hantrakul, Weituo Hao, Qingqing Huang, Zhongyi Huang, Dongya Jia, Feihu La, Duc Le, Bochen Li, Chumin Li, Hui Li, Xingxing Li, Shouda Liu, Wei-Tsung Lu, Yiqing Lu, Andrew Shaw, Janne Spijkervet, Yakun Sun, Bo Wang, Ju-Chiang Wang, Yuping Wang, Yuxuan Wang, Ling Xu, Yifeng Yang, Chao Yao, Shuo Zhang, Yang Zhang, Yilin Zhang, Hang Zhao, Ziyi Zhao, Dejian Zhong, Shicen Zhou, and Pei Zou. 2024. Seed-music: A unified framework for high quality and controlled music generation. *CoRR*, abs/2409.09214.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matthew Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. 2023. Audiolm: A language modeling approach to audio generation. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:2523–2533.

Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. 2017. Deep learning techniques for music generation - A survey. *CoRR*, abs/1709.01620.

Dong-Min Byun, Sang-Hoon Lee, Ji-Sang Hwang, and Seong-Whan Lee. 2024. Midi-voice: Expressive zero-shot singing voice synthesis via midi-driven priors. In *Proc. ICASSP*, pages 12622–12626. IEEE.

Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2024a. Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *Proc. ICASSP*, pages 1206–1210.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. 2016. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2024b. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. *CoRR*, abs/2410.06885.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2023. Simple and controllable music generation. In *Proc. NeurIPS*.

Tri Dao. 2024. Flashattention-2: Faster attention with better parallelism and work partitioning. In *Proc. ICLR*.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2023. High fidelity neural audio compression. *Trans. Mach. Learn. Res.*

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

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. 2018. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proc. AAAI*, pages 34–41.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, Yanqing Liu, Sheng Zhao, and Naoyuki Kanda. 2024. E2 TTS: embarrassingly easy fully non-autoregressive zero-shot TTS. *CoRR*, abs/2406.18009.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Proc. ICML*.

Zach Evans, CJ Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. 2024a. Fast timing-conditioned latent audio diffusion. In *Proc. ICML*.

Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. 2024b. Long-form music generation with latent diffusion. *CoRR*, abs/2404.10301.

Zhengcong Fei, Mingyuan Fan, Changqian Yu, and Junshi Huang. 2024. Flux that plays music. *arXiv preprint arXiv:2409.00587*.

Jinzheng He, Jinglin Liu, Zhenhui Ye, Rongjie Huang, Chenye Cui, Huadai Liu, and Zhou Zhao. 2023. Rmssinger: Realistic-music-score based singing voice synthesis. In *Proc. ACL*, pages 236–248.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598.

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

Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. 2022. Mulan: A joint embedding of music audio and natural language. In *Proc. ISMIR*, pages 559–566.

Rongjie Huang, Yongqi Wang, Ruofan Hu, Xiaoshan Xu, Zhiqing Hong, Dongchao Yang, Xize Cheng, Zehan Wang, Ziyue Jiang, Zhenhui Ye, et al. 2024. Voicetuner: Self-supervised pre-training and efficient

fine-tuning for voice generation. In *ACM Multimedia 2024*.

Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. 2024. Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization. *arXiv preprint arXiv:2412.21037*.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. 2019. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Proc. Interspeech*, pages 2350–2354.

Robert Kubichek. 1993. Mel-cepstral distance measure for objective speech quality assessment. In *Proc. PACRIM*, pages 125–128.

Max W. Y. Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, Jitong Chen, Yuping Wang, and Yuxuan Wang. 2023. Efficient neural music generation. In *Proc. NeurIPS*.

Gael Le Lan, Varun Nagaraja, Ernie Chang, David Kant, Zhaoheng Ni, Yangyang Shi, Forrest Iandola, and Vikas Chandra. 2024. Stack-and-delay: a new codebook pattern for music generation. In *Proc. ICASSP*, pages 796–800.

Shun Lei, Yixuan Zhou, Boshi Tang, Max W. Y. Lam, Feng Liu, Hangyu Liu, Jingcheng Wu, Shiyin Kang, Zhiyong Wu, and Helen Meng. 2024. Songcreator: Lyrics-based universal song generation. In *Proc. NeurIPS*.

Ruiqi Li, Zhiqing Hong, Yongqi Wang, Lichao Zhang, Rongjie Huang, Siqi Zheng, and Zhou Zhao. 2024a. Accompanied singing voice synthesis with fully text-controlled melody. *CoRR*, abs/2407.02049.

Singsong Li, Shu Liu, Liming Ma, and Chaoping Xing. 2024b. Asymptotic construction of locally repairable codes with multiple recovering sets. *CoRR*, abs/2402.09898.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow matching for generative modeling. In *Proc. ICLR*.

Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proc. AAAI*, pages 11020–11028.

Zhijun Liu, Shuai Wang, Sho Inoue, Qibing Bai, and Haizhou Li. 2024. Autoregressive diffusion transformer for text-to-speech synthesis. *arXiv preprint arXiv:2406.05551*.

Ziqian Ning, Shuai Wang, Yuepeng Jiang, Jixun Yao, Lei He, Shifeng Pan, Jie Ding, and Lei Xie. 2024. Drop the beat! freestyler for accompaniment conditioned rapping voice generation. *CoRR*, abs/2408.15474.

Marco Pasini, Stefan Lattner, and George Fazekas. 2024. Music2latent: Consistency autoencoders for latent audio compression. *CoRR*, abs/2408.06500.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proc. ICCV*, pages 4172–4182.

Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. 2001. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *Proc. ICASSP*, pages 749–752.

Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. 2021. Songmass: Automatic song writing with pre-training and alignment constraint. In *Proc. AAAI*, pages 13798–13805.

Christian J Steinmetz and Joshua D Reiss. 2020. auraloss: Audio focused loss functions in pytorch. In *Proc. DMRN*.

Cees H. Taal, Richard C. Hendriks, Richard Heusdens, and Jesper Jensen. 2010. A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *Proc. ICASSP*, pages 4214–4217.

Yongqi Wang, Ruofan Hu, Rongjie Huang, Zhiqing Hong, Ruiqi Li, Wenrui Liu, Fuming You, Tao Jin, and Zhou Zhao. 2024a. Prompt-singer: Controllable singing-voice-synthesis with natural language prompt. *CoRR*, abs/2403.11780.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Shunsi Zhang, and Zhizheng Wu. 2024b. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *CoRR*, abs/2409.00750.

Kai-Tuo Xu, Feng-Long Xie, Xu Tang, and Yao Hu. 2025. Fireredasr: Open-source industrial-grade mandarin speech recognition models from encoder-decoder to llm integration. *CoRR*, abs/2501.14350.

Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang, and Tie-Yan Liu. 2021. Deeprapper: Neural rap generation with rhyme and rhythm modeling. In *Proc. ACL/IJCNLP*, pages 69–81.

Chenyu Yang, Shuai Wang, Hangting Chen, Jianwei Yu, Wei Tan, Rongzhi Gu, Yaoxun Xu, Yizhi Zhou, Haina Zhu, and Haizhou Li. 2024. Songeditor: Adapting zero-shot song generation language model as a multi-task editor. *CoRR*, abs/2412.13786.

Yongmao Zhang, Heyang Xue, Hanzhao Li, Lei Xie, Tingwei Guo, Ruixiong Zhang, and Caixia Gong. 2023. Visinger2: High-fidelity end-to-end singing voice synthesis enhanced by digital signal processing synthesizer. In *Proc. Interspeech*, pages 4444–4448.

Yu Zhang, Rongjie Huang, Ruiqi Li, Jinzheng He, Yan Xia, Feiyang Chen, Xinyu Duan, Baoxing Huai, and Zhou Zhao. 2024. Stylesinger: Style transfer for out-of-domain singing voice synthesis. In *Proc. AAAI*, pages 19597–19605.