# WanSong v1.0 Technical Report

Wan Team, Alibaba Group

Music generation foundation models have recently attracted significant industry attention. However, achieving efficient generation and high-fidelity long-form audio while supporting controllability remains challenging. To address these needs, we present **WanSong**, a simple yet powerful approach for long-form, commercial-grade song generation. Unlike autoregressive (AR) and cascaded multi-stage pipelines (*e.g.*, AR followed by diffusion), **WanSong** is a pure diffusion-based model that directly generates high-fidelity, multilingual songs up to 5 minutes and outputs dual stems (vocals and background music) in a single run. In addition, our diffusion framework enables faster inference through step-distillation, and offers an efficient pathway for fine-tuning and customization to support downstream editing tasks.

**Date:** July 17, 2026
**Model Card:** WanSong v1.0



## 1 Introduction

Music/song foundation models are undergoing a paradigm shift. Commercial-grade systems, *e.g.* Suno (2026) and Mureka (2026), have demonstrated impressive music fidelity and fluency. However, most existing methods remain fundamentally based on autoregressive (AR) modeling. While AR models offer a principled way to generate sequences, they can lead to challenges in generation efficiency and in maintaining consistent perceptual quality for long-form audio.

Some recent works incorporate diffusion approaches on top of AR pipelines, for instance by using diffusion as an additional refinement module to improve audio realism. Building on these ideas, we further investigate how to design music foundation models in a more end-to-end manner. In particular, we aim to directly model audio generation without relying on a multi-stage AR-dominant pipeline, enabling a unified framework for controllable long-form song generation.

Motivated by this goal, we abandon conventional AR-dominant approaches and instead treat audio as a sequence of continuous tokens. We then develop an end-to-end, single-stage music foundation model using a pure diffusion scheme.

In this work, we present **WanSong v1.0**, an end-to-end single-stage diffusion-based music foundation model that treats audio directly as continuous tokens. Specifically, we employ a refined hybrid-MMDit architecture as the model backbone conditioned on an LLM text captioner. All text and audio tokens are sequentially concatenated into one unified sequence, feeding into the hybrid-MMDit model. However, we find that jointly encoding both vocals and BGM[1] within a single token causes the diffusion model to over-focus on the vocal components, while the more complex BGM part is suppressed. In addition, Classifier-Free Guidance (CFG) cannot achieve a good balance between vocals and BGM: as CFG increases, the vocals become more accurate but the BGM weakens; conversely, when CFG is smaller, the BGM improves but the vocals deteriorate. To address these issues, we propose a novel dual-stem token scheme that models vocals and BGM separately, fundamentally eliminating the interference between them. In addition, we also employ reinforcement learning (RL) to align the model with human preferences.

Our contributions are summarized as follows:

- We present WanSong v1.0, a commercial-grade text-to-music model that supports multilingual, high-fidelity, and long-duration song generation.

[1] or called accompaniment

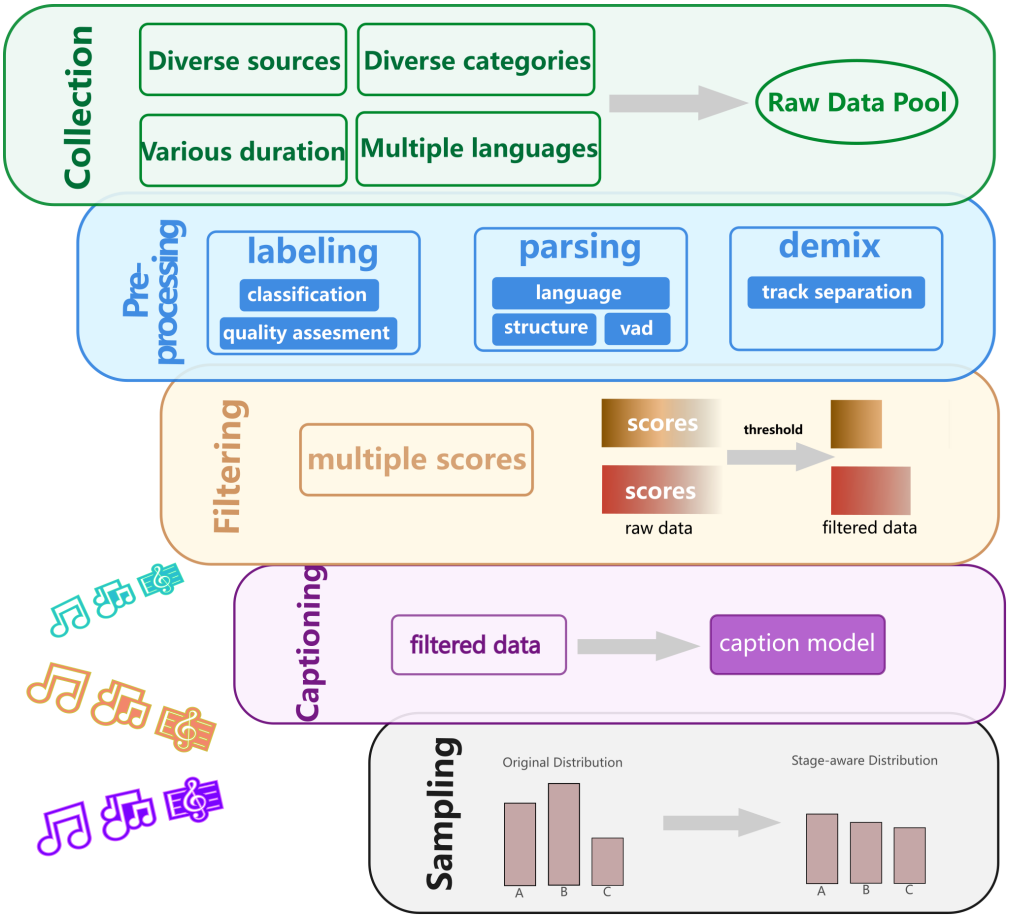


**Figure 1** Data pipeline overview.

- Pure diffusion-based framework: audios are treated as continuous tokens and then fed into an end-to-end hybrid-MMDit backbone for learning the relations between text captions and songs.
- Dual-stem modeling: We independently learn the vocal and BGM components, thereby preventing cross-interference between the two. As a result, WanSong can directly output separated stems for vocals and BGM, which makes subsequent post-production editing more convenient.

## 2 Data

As a key foundation for large-scale music generative models, the quality and diversity of training data greatly determine overall model performance. To build a high-quality, well-balanced, and diverse dataset, we develop a rigorous data curation pipeline with five stages: collection, pre-processing, filtering, captioning, and sampling, as shown in Figure 1.

**Data collection:** We collect a large and diverse corpus of audio data covering multiple languages, diverse musical styles, both vocal songs and instrumental music, and a wide range of song durations.

**Data pre-processing:** As the raw collected audio dataset is noisy, we implement a multi-stage pre-processing pipeline. Specifically, each audio file undergoes audio classification, quality assessment, structural parsing, VAD (voice activity detection), language classification, and track separation. This process yields a candidate pool of musics with diverse annotations.

**Data filtering:** By jointly considering the audio category and audio quality, we construct a high-quality, balanced sub-dataset for model training. In addition, the multi-dimensional labeling taxonomy we develop allows us to schedule training data in a flexible and adaptive manner across different training phases.

**Captioning:** Captioning is one of the most important components, and the effectiveness of captioning directly drives the model's generation performance. In order to enhance the captioning ability, We have meticulously annotated musical style, vocals, instruments, mode and key, language, reverb, emotions, lyrics, and more.

**Data sampling:** Since the collected data are imbalanced across different categories (*e.g.*, music styles, vocal gender, audio quality and so on), we adopt a fine-grained balancing strategy at each stage of training.

The resulted training set has more than 6 million hours of multilingual song data.

# 3 Method

## 3.1 Variational Autoencoder

To bridge raw waveforms and the downstream diffusion transformer, we train a continuous 1-D Variational Autoencoder (VAE) that compresses a stereo audio signal into a compact latent stream. Compared to Stable Audio 2 Evans et al. (2025), we halve the compression ratio to retain substantially more time-frequency detail, which is crucial for reproducing sharp transients and high-fidelity source separation.

**Overall Configuration.** The VAE maps a stereo waveform $x \in \mathbb{R}^{2\times T}$ at 44.1 kHz to a latent tensor $z \in \mathbb{R}^{C_z \times T'}$, where $C_z = 64$ and $T' = T/1024$. This corresponds to a downsampling factor of 1024 and a latent frame rate of $\approx 43.1$ Hz.

**Training Objective.** The generator $G$ (encoder-decoder) and multi-scale discriminators $D$ are trained adversarially:

$$\begin{cases} \mathcal{L}_G = \lambda_{\text{mag}}\mathcal{L}_{\text{mag}} + \lambda_{\text{fmap}}\mathcal{L}_{\text{fmap}} + \lambda_{\text{adv}}\mathcal{L}_{\text{Adv}}(G;D) + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}, \\ \mathcal{L}_D = \mathcal{L}_{\text{Adv}}(D;G), \end{cases} \tag{1}$$

where:

- $\mathcal{L}_{\text{mag}}$: Multi-resolution STFT magnitude loss computed on Mid-Side (M/S) and Left-Right (L/R) channels.
- $\mathcal{L}_{\text{fmap}}$: $\ell_1$ discriminator feature-matching loss.
- $\mathcal{L}_{\text{Adv}}$: Hinge adversarial loss.
- $\mathcal{L}_{\text{KL}}$: KL divergence with the standard normal prior.

**Implementation Details.** We train the model on a curated corpus of $\sim 2.6 \times 10^8$ audio clips. We filter the dataset using quality classifiers to remove degraded, low-SNR, or low-bitrate samples, and apply silence detection to discard inactive segments. The retained corpus is explicitly balanced with a domain ratio of Music : Speech : Sound = 50% : 40% : 10%. All clips are resampled to 44.1 kHz stereo, and we employ random phase inversion for data augmentation.

We optimize the model using AdamW with a learning rate of $2 \times 10^{-4}$ for $G$ and $3 \times 10^{-4}$ for $D$. Training is conducted end-to-end on 32 NVIDIA A100 GPUs for 1.5M steps using 1.5 s stereo crops. Notably, during the latter stage of training, we apply random noise augmentation to the latent representation to robustify the decoder against out-of-distribution artifacts during inference.

## 3.2 Architecture

All tokens, *i.e.*, text tokens encoded by decoder-only LLM and dual-stem audio continuous tokens encoded VAE, are concatenated together as a sequence and then fed into our transformer-based diffusion backbone. It is worth noting that, in order to improve token throughput and token utilization within each batch, we pack tokens from different samples together by concatenating their sequences and feed the combined stream into the network.

**Hybrid transformer:** MMDit proposed by Esser et al. (2024) has shown promising results in image and video foundation model settings. We follow this basic idea to process tokens from different modalities. However, our experiments show that we do not need to learn separate parameters for different modalities in every layer. Inspired by the ideas in Flux (Labs et al. (2025)), we ultimately adopt a hybrid-transformer architecture.

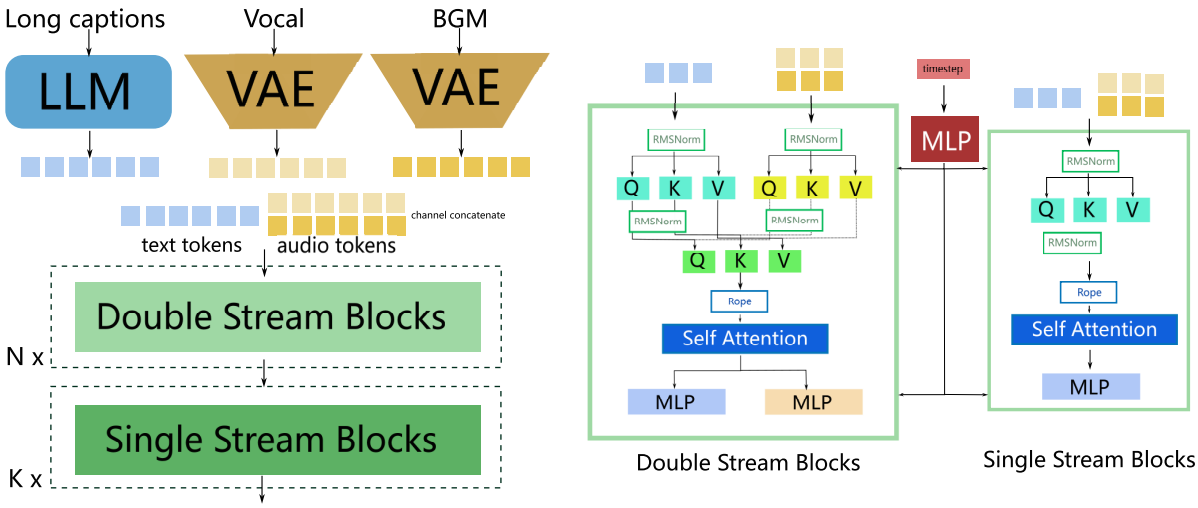


**Figure 2** Our hybrid-transformer backbone.

Moreover, following our previous settings in Wan2.1 (Wan et al. (2025)), we employ the fully-shared AdaLN in each block to effectively reduce parameters while preserving performance.

**Dual-stem modeling**: In diffusion models, CFG typically serves to adjust the output distribution. However, in our early experiments, we found that a larger CFG improves phoneme accuracy, but it also suppresses the BGM by vocals. Conversely, a smaller CFG better restores the BGM, but the vocal accuracy degrades. To address this issue, we propose a dual-stem output with joint-learning-in-layer strategy. It means that the vocal and BGM tokens are produced independently at the model's output. While, during the learning of each block, we treat them as different channels of the same token, such that the model maintains independent output streams while simultaneously learning their intrinsic dependencies.

The overall architecture is shown in Figure 2. $\frac{N}{N+K} = 0.3$ and the total model size is nearly 25B.

## 3.3 Training

To gradually learn the fine details of long-duration songs, the pre-training phase consists of 3 stages, including 90s-duration training, 300s-duration training and supervised-finetuning(SFT) stages. The flow matching framework (Lipman et al. (2022), Esser et al. (2024)) is used to model a denoising diffusion process within the audio domain. During training, given an audio latent $X_1$, a random noise $X_0 \sim \mathcal{N}(0, I)$, and a timestep $t \in [0, 1]$ sampled from a logit-normal distribution, an intermediate latent $X_t$ is obtained as the training input. Following Rectified Flows (RFs) (Esser et al. (2024)), $X_t$ is defined as a linear interpolation between $X_0$ and $X_1$:

$$X_t = tX_1 + (1 - t)X_0 \tag{2}$$

The ground truth velocity $V_t$ is:

$$V_t = \frac{dX_t}{dt} = X_1 - X_0 \tag{3}$$

Then the model can be trained to predict this velocity, the loss function can be formulated as the mean squared error (MSE) between the model output and $V_t$. In this work, our loss function is as:

$$L = \alpha_{vocal}\mathbb{E}_t[||\theta(X_{vocal,t}, C_{txt}, t) - V_{vocal,t}||^2] + \alpha_{bgm}\mathbb{E}_t[||\theta(X_{bgm,t}, C_{txt}, t) - V_{bgm,t}||^2], \tag{4}$$

where $\alpha_{vocal}$ and $\alpha_{bgm}$ are coefficients for balancing the learning strength between vocal and bgm. $C_{txt}$ represents the text condition.

## 3.4 Reinforcement Learning

To further enhance the quality of generated songs/musics, we employ Reinforcement Learning from Human Feedback (RLHF) to align model with human preferences. Specifically, we focus on three critical dimensions that significantly influence the overall listening experience: **musicality**, **lyric accuracy** and **prompt alignment**.

For each dimension, we collect dimension-specific prompts and infer out the corresponding audios, then human preference annotations over these audios are collected. We train reward models to provide fine-grained supervision signals. These reward models are subsequently used to guide the music generation model via reinforcement learning. Each reward model can be trained by:

$$L = -\mathbb{E}_{a^+,a^-}[\log \tau(R(a^+) - R(a^-))] \tag{5}$$

where $a^+, a^-$ represent the positive and negative sample annotated by human, respectively. With these paired samples, the reward model can learn what is good and what is bad in each dimension. After training these reward models, we employ both DPO(Wallace et al. (2024)) and ReFL(Xu et al. (2023)). [2]

Since the DPO scheme directly optimizes the relationships between positive and negative samples, it provides supervision for every timestep $t \in [0, 1]$. This makes it more beneficial for optimizing the global structure of the audio. But a weakness of DPO is that it can't precisely push both positive and negative samples toward more favorable directions. This can be seen directly from the loss function itself: the pairwise formulation inherently introduces a relatively large amount of noise. In contrast, ReFL-like methods might work well since it directly optimize the model by precise reward evaluation and information backward. Unfortunately, due to limitations in its formulation, ReFL only works during the low-noise stage. Therefore, it is inclined to optimize fine-grained details, and it is also more susceptible to being "hacked." To achieve complementary effects, we first perform DPO training and then follow with ReFL training.

# 4 Evaluation

## 4.1 VAE

We evaluate reconstruction fidelity on two held-out benchmarks that jointly stress the two dominant modalities the VAE has to serve: a music benchmark of 200 randomly held-out clips from the internal Wan-song corpus, and the standard SeedTTS speech benchmark Anastassiou et al. (2024), from which we randomly sampled 2000 clips. For every clip we run encode → decode and report **(i)** STFT distance ($\ell_1$ on log-magnitude spectra, lower is better), **(ii)** mel-spectrogram distance (lower is better), and **(iii)** scale-invariant SDR (dB, higher is better).

We compare against two baselines: the official Stable Audio 2 VAE (compression 2048), and an ablation of our own model at compression 2048 ("Ours 2048") that shares our training data and network architecture — so the gap between "Ours 2048" and Stable Audio 2 isolates the effect of everything *except* the compression-rate change, while the gap between "Ours 1024" and "Ours 2048" isolates the effect of the compression rate itself. Results are summarized in Table 1 and Table 2.

**Table 1** VAE reconstruction quality on the wan-song music benchmark (200 clips).

| Method | STFT ↓ | MEL ↓ | SI-SDR (dB) ↑ |
|---|---|---|---|
| Stable Audio 2 | 1.430 | 0.856 | 4.386 |
| Ours (2048) | 1.163 | 0.772 | 4.661 |
| **Ours (1024)** | **1.029** | **0.629** | **7.246** |

**Table 2** VAE reconstruction quality on the SeedTTS speech benchmark (2000 random samples).

| Method | STFT ↓ | MEL ↓ | SI-SDR (dB) ↑ |
|---|---|---|---|
| Stable Audio 2 | 0.999 | 0.754 | 8.653 |
| Ours (2048) | 0.783 | 0.593 | 10.371 |
| **Ours (1024)** | **0.698** | **0.458** | **12.922** |

We present two key observations from these results:

**First, our architectural optimizations and domain balancing yield superior reconstruction over Stable Audio 2 at matched compression ratios.** Specifically, at a compression factor of 2048, our model outperforms Stable

[2]The loss function details of DPO and ReFL are not the core of this paper, so please refer to the original paper for more details.

Audio 2 across all metrics on both datasets (e.g., −19% STFT and +0.28 dB SI-SDR on music; −22% STFT and +1.72 dB SI-SDR on speech). We attribute these gains to our domain-balanced 260M corpus and wider encoder/decoder capacities. This validates our 50/40/10 domain rebalancing decision: a purely music-centric VAE systematically struggles with transient fricative and plosive speech details, whereas our joint music/speech/sound training resolves this weakness without compromising music-specific fidelity.

**Second, halving the compression ratio from 2048 to 1024 provides the most substantial boost in overall fidelity.** This change is most prominent in SI-SDR performance, triggering a +2.86 dB improvement on the wan-song benchmark and a +4.27 dB jump on SeedTTS. This demonstrates that a 2048× latent grid is inherently too coarse to capture sharp, fine-grained transient details regardless of how heavily optimized the underlying neural backbone is, confirming the necessity of our 1024× design choice.

## 4.2 WanSong Bench

**Objective quantitative results:** In order to evaluate the effectiveness of our WanSong, we collect a testing benchmark, including 200-samples covering four languages (Chinese, English, Japanese, Korean) and over ten Level-1 musical genres (*e.g.*, Pop, Funk, EDM, Rock, Country, Jazz, *etc.*). All samples' durations are around 4 minutes. We evaluate the performances by Pronunciation Error Rate (PER), Songeval(Yao et al. (2025)) and Muq text alignment(Zhu et al. (2025)) as shown in Table 3.

However, since the evalscore is trained on near 2,000 songs, its generalization ability is difficult to guarantee. As shown in Table 3, the score differences are very small. To address this, we use our own musicality evaluation model to perform the assessment. Specifically, this model employs a comprehensive scoring scheme that combines multiple aspects—melody, structure, and overall listening experience ("harmony," "texture," "arrangement," "emotion," "vocals," and "reverb"). These components are weighted and aggregated with ratios of 0.4, 0.25, and 0.35, respectively, to produce the final single score. The score range is $[0-10]$. Our model are trained on 80k songs annotated by human. The performance comparisons are shown in Table 4.

**Table 3** Objective quantitative evaluations.

| | PER (%) ↓ | SongEval ↑ | | | | | Muq ↑ |
|---|---|---|---|---|---|---|---|
| | | Cla | Coh | Mem | Mus | Nat | |
| LeVo(Lei et al. (2026)) | 27.11 | 3.42 | 3.59 | 3.47 | 3.45 | 3.28 | 0.34 |
| MurekaV7.6(Mureka (2026)) | 12.7 | 4.38 | 4.51 | 4.47 | 4.35 | 4.32 | 0.43 |
| SunoV5(Suno (2026)) | 22.80 | 4.44 | **4.56** | 4.53 | 4.41 | 4.34 | **0.44** |
| WanSong(ours) | **7.43** | **4.47** | 4.55 | **4.57** | **4.46** | **4.4** | **0.44** |

**Table 4** Objective quantitative evaluations.

| | levo(Lei et al. (2026)) | MurekaV7.6(Mureka (2026)) | SunoV5(Suno (2026)) | WanSong(ours) |
|---|---|---|---|---|
| Musicality ↑ | 1.69 | 3.83 | 4.18 | **5.49** |

**Ablation study on compression ratio of token:** For convenience, we only conducted comparative experiments in the PT-90s stage. We trained our model based on these three different setups: (1) a VAE based on a 2048 compression rate; (2) a VAE based on a 1024 compression rate, with an additional patch size of 2 (*i.e.*, concatenating adjacent tokens into a single token); and (3) a VAE based on a 1024 compression rate. We tested the above three models on our 90s-bench. The final results are shown in the table 5.

One can observe from Table 5 that increasing the token compression ratio generally leads to worse perceptual quality. When using a VAE with compression ratio 2048 (patch size = 1), the model achieves a PER of 19.2% and a Quality score [3] of 2.1. For the same effective compression ratio (Actual compression ratio = 2048), adopting a lower base compression ratio (1024) together with a larger patch size (patch size = 2, i.e., concatenating neighboring tokens) results in a slightly higher PER of 20.6% and a similar Quality score of 2.4, suggesting that a larger patch size introduces an inherent compression-related degradation that cannot be

[3]Audio quality evaluation was conducted by our expert assessors, as both EvalScore Yao et al. (2025) and AudioBox Tjandra et al. (2025) cannot precisely measure the song audio quality.

fully recovered by a better VAE configuration. In contrast, further reducing the actual compression ratio to 1024 (patch size = 1) substantially improves both metrics, yielding the best PER of 15.0% and the highest Quality score of 3.2. Overall, these results suggest that the effective token compression ratio is the dominant factor in balancing efficiency and audio quality. However, we did not further reduce the compression ratio because a too-low compression rate would cause the number of tokens to increase dramatically, leading to higher resource consumption and slower generation speed.

**Table 5** Ablation study on token compression ratio.

| PT-90s model | | | | |
|---|---|---|---|---|
| VAE compression ratio | patch size | Actual compression ratio | PER(%) ↓ | Quality (1-5) ↑ |
| 2048 | 1 | 2048 | 19.2 | 2.1 |
| 1024 | 2 | 2048 | 20.6 | 2.4 |
| 1024 | 1 | 1024 | **15.0** | **3.2** |

## 5 Related Work

Song/Music generation aims to produce coherent vocals and BGM given lyrics and style prompts. And there are many types of works implementing this basic target. For example, Musiclm(Agostinelli et al. (2023)) generate music by multi-stage sequence-to-sequence pipeline. SongGen(Liu et al. (2025)) uses discrete tokens to train an autoregressive (AR) model. Yue(Yuan et al. (2025)) and LeVo(Lei et al. (2026)) also employ the discrete tokens to train multi-stage AR models, which are sometimes complex. SongCreator(Lei et al. (2024)) employs U-Net based diffusion model, but fundamentally it still requires discrete tokens and a multi-stage generation pipeline. MusicCot(Lam et al. (2025)) tries to use chain-of-thought to enhance the reasoning ability of songs. Recently, some works start to employ diffusion strategy to enhance the audio quality. DiffRhythm2(Jiang et al. (2025)) proposes semi-autoregressive architecture based on block flow matching. ACE-Step1.5(Gong et al. (2026)) proposes LM planner based diffusion model. They all require more elaborate training pipeline/component design. To this end, this paper proposes a simple yet effective pure diffusion framework that directly outputs demixed vocal and BGM stems for high-fidelity song generation. No gimmicky tricks are needed.

## 6 Conclusion

This paper presents WanSong v1.0, a simple yet effective pure diffusion-based framework for long-form, commercial-grade music generation. Unlike autoregressive systems and cascaded AR+diffusion designs, WanSong adopts an end-to-end diffusion pipeline where audio is modeled as continuous tokens and generated in a single stage, enabling both strong fidelity and efficient inference through step distillation.

A key insight of our work is that vocal and BGM should not be jointly encoded as entangled targets. To address the resulting imbalance and the failure of CFG to simultaneously preserve vocal accuracy and BGM strength, we introduce a dual-stem token scheme with joint learning within each block. This design fundamentally reduces mutual interference, allowing WanSong to directly output demixed vocals and BGM stems, which greatly simplifies downstream post-production and editing workflows.

In addition, we further improve listening quality by aligning the generation process with human preferences using RLHF, targeting multiple dimensions including musicality, lyric accuracy, and prompt alignment. Extensive evaluations on our WanSong benchmark demonstrate that WanSong achieves strong performance across objective metrics, validating the effectiveness of the proposed diffusion framework and dual-stem modeling strategy.

Overall, WanSong v1.0 establishes a practical direction for diffusion-based music foundation models—delivering high fidelity, multilingual capability, long-duration generation, and editable stem outputs—without relying on complicated, gimmicky components.

## 7 Authors

Binghui Chen, Pandeng Li, Yu Liu, Jingren Zhou

## References


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.

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *ICASSP*, pages 1–5. IEEE, 2025.

Junmin Gong, Yulin Song, Wenxiao Zhao, Sen Wang, Shengyuan Xu, Jing Guo, and Xuerui Yang. Ace-step 1.5: Pushing the boundaries of open-source music generation. *arXiv preprint arXiv:2602.00744*, 2026.

Yuepeng Jiang, Huakang Chen, Ziqian Ning, Jixun Yao, Zerui Han, Di Wu, Meng Meng, Jian Luan, Zhonghua Fu, and Lei Xie. Diffrhythm 2: Efficient and high fidelity song generation via block flow matching. *arXiv preprint arXiv:2510.22950*, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. https://arxiv.org/abs/2506.15742.

Max WY Lam, Yijin Xing, Weiya You, Jingcheng Wu, Zongyu Yin, Fuqiang Jiang, Hangyu Liu, Feng Liu, Xingda Li, Wei-Tsung Lu, et al. Analyzable chain-of-musical-thought prompting for high-fidelity music generation. *arXiv preprint arXiv:2503.19611*, 2025.

Shun Lei, Yixuan Zhou, Boshi Tang, Max W Lam, Feng Liu, Hangyu Liu, Jingcheng Wu, Shiyin Kang, Zhiyong Wu, and Helen Meng. Songcreator: Lyrics-based universal song generation. *Advances in Neural Information Processing Systems*, 37:80107–80140, 2024.

Shun Lei, Yaoxun Xu, Huaicheng Zhang, Hangting Chen, Yixuan Zhang, Chenyu Yang, Haina Zhu, Shuai Wang, Zhiyong Wu, Dong Yu, et al. Levo: High-quality song generation with multi-preference alignment. *Advances in Neural Information Processing Systems*, 38:102448–102479, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Zihan Liu, Shuangrui Ding, Zhixiong Zhang, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Songgen: A single stage auto-regressive transformer for text-to-song generation. *arXiv preprint arXiv:2502.13128*, 2025.

Mureka, 2026. https://www.mureka.cn/.

Suno, 2026. https://suno.com/discover.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, Carleigh Wood, Ann Lee, and Wei-Ning Hsu. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. 2025. https://arxiv.org/abs/2502.05139.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu,

Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Jixun Yao, Guobin Ma, Huixin Xue, Huakang Chen, Chunbo Hao, Yuepeng Jiang, Haohe Liu, Ruibin Yuan, Jin Xu, Wei Xue, et al. Songeval: A benchmark dataset for song aesthetics evaluation. *arXiv preprint arXiv:2505.10793*, 2025.

Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, et al. Yue: Scaling open foundation models for long-form music generation. *arXiv preprint arXiv:2503.08638*, 2025.

Haina Zhu, Yizhi Zhou, Hangting Chen, Jianwei Yu, Ziyang Ma, Rongzhi Gu, Yi Luo, Wei Tan, and Xie Chen. Muq: Self-supervised music representation learning with mel residual vector quantization. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.