Qwen-Music Technical Report

Qwen Team

Abstract

In this report, we introduce Qwen-Music, a powerful music generation model capable of producing highly musical and high-fidelity songs with complete vocal singing. Qwen-Music supports two core tasks: **Text to Music Generation**, which create entirely new songs from text descriptions, lyrics, and musical attributes, and **Cover Song Generation**, which reinterprets existing songs with different styles and vocal characteristics. Architecturally, Qwen-Music integrates three core components: **Qwen-Music-Tokenizer**, **Qwen-Music-LLM**, and **Qwen-Music-Render**. Qwen-Music-Tokenizer compresses audio into a 25 Hz single-codebook stream of Music Semantic Tokens that preserve semantic and melodic information for LLM prediction. Based on these tokens, Qwen-Music-LLM performs autoregressive music semantic modeling, with a key novelty being a melody-token-based chain-of-thought (Melody-CoT) mechanism that plans melodies before full-song generation, improving creativity, musicality, structural coherence, and reference-audio-based melody cloning. To overcome the fidelity limitations of discrete semantic tokens, Qwen-Music-Render performs generative stereo rendering, enriching acoustic details and producing high-fidelity stereo waveforms. Finally, we train Qwen-Music-LLM on more than 5 million hours of multilingual music data covering hundreds of languages. We first apply quality-aware pre-training curriculum, then use progressive post-training, comprising supervised initialization, offline DPO, and online GSPO, to further improve musicality and instruction-following ability. Across 600 Chinese and English prompts, Qwen-Music achieves state-of-the-art results in 13 of 16 objective musicality and audio-quality metrics. Professional evaluators also prefer Qwen-Music over leading proprietary systems. For cover song generation, Qwen-Music preserves reference melodies more accurately than Suno V5.5, Suno V5, and MiniMax Cover on the AI-generated reference set, and outperforms MiniMax Cover on most metrics in the real-world popular-song reference set.

Figure 1: Qwen-Music is a powerful and controllable music generation model capable of producing songs with complete vocal singing. It generates complete songs from text descriptions, lyrics, and musical attributes, and uses reference audio for cover song generation with different styles and vocal characteristics.

1 Introduction

Music generation has recently emerged as a key challenge in generative modeling, aiming to synthesize music that is musically coherent, acoustically natural, and semantically aligned with user intent (Agostinelli et al., 2023; Schneider et al., 2023; Copet et al., 2024; Lam et al., 2024; Majumder et al., 2024; Huang et al., 2023; Chen et al., 2024). Unlike general audio generation, song generation requires the joint modeling of lyrics, melody, rhythm, vocal performance, instrumentation, and musical structure over minutes (Yang et al., 2026; Lei et al., 2026b;a; Liu et al., 2025; Gong et al., 2025; 2026; Yuan et al., 2025; Ning et al., 2025; Jiang et al., 2025; Lei et al., 2024). A useful system should support open-ended song generation from text descriptions and lyrics while allowing users to control musical attributes such as genre, mood, instrumentation, and vocal timbre. It should also reinterpret existing songs by preserving a reference melody while changing style or vocal characteristics. Despite rapid progress in large-scale audio generation models, generating songs with stable melodic development, clear lyric articulation, realistic singing voices, and natural instrumental accompaniment remains challenging.

The central technical challenge is the mismatch between semantic composition and acoustic rendering. At the semantic level, a model must plan lyrics, vocal melody, section structure, repetition, and stylistic progression over long horizons. At the acoustic level, it must render timbre, phase, stereo placement, and high-frequency detail at waveform resolution. Discrete-token language models provide a scalable interface for global sequence generation, but compressed tokens lose acoustic detail (Xu et al., 2025; Défossez et al., 2022; Yang et al., 2026). Reference-guided cover song generation adds another constraint: the system must reuse the melody while allowing controllable changes to the arrangement, singing voice, and style.

In this report, we introduce **Qwen-Music**, a large-scale music generation system that separates song generation into semantic composition and acoustic rendering. Qwen-Music supports both text-to-music generation and reference-audio-based cover song generation within a single framework. Given text descriptions, lyrics, musical attributes, and optional reference audio, Qwen-Music can generate complete songs with controllable genre, mood, instrumentation, vocal timbre, and vocal gender. Trained on over 5 million hours of multilingual music data, Qwen-Music generalizes strongly across diverse languages, genres, and musical styles.

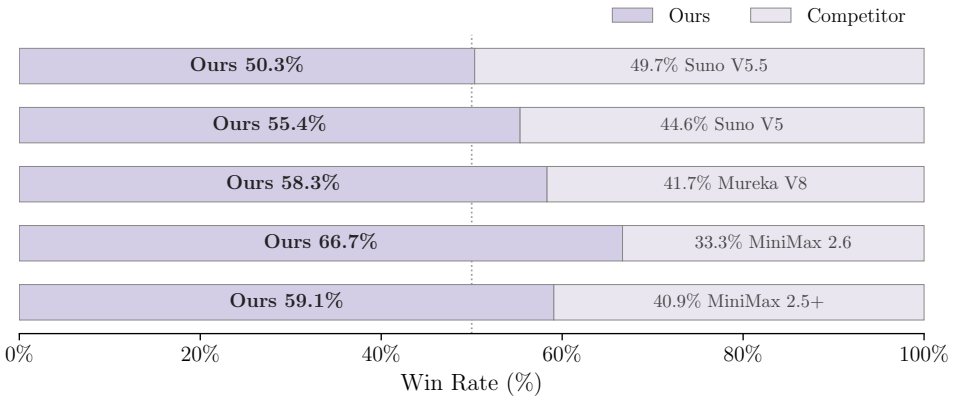
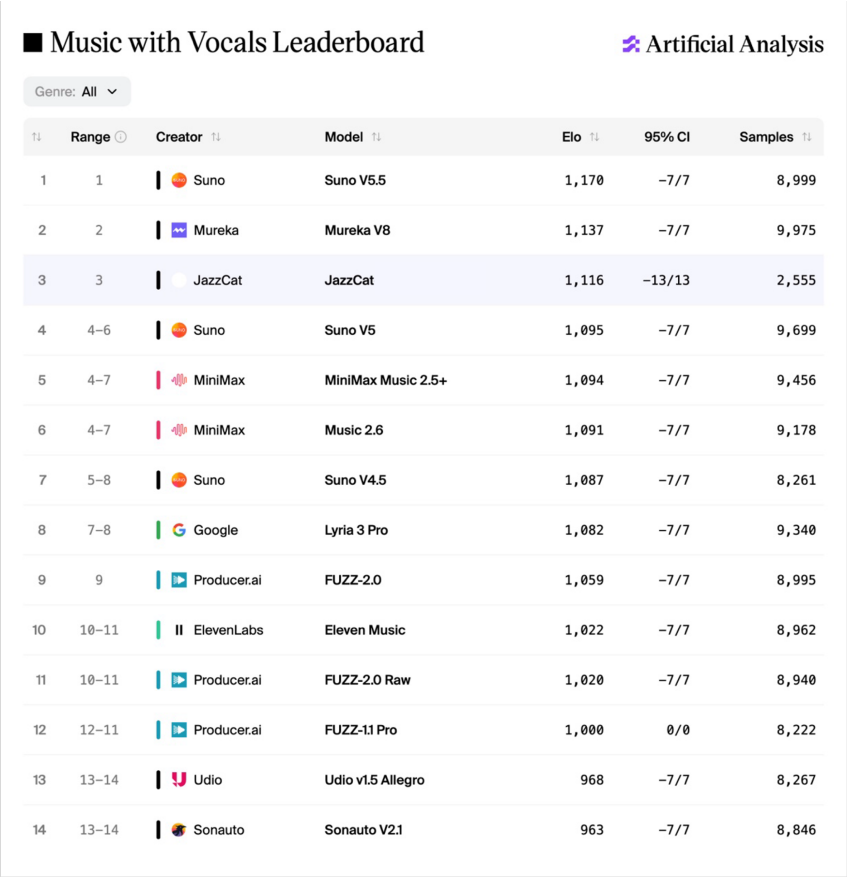


Figure 2: Blind A/B preference results between Qwen-Music and leading proprietary systems, including MiniMax Music 2.5+, MiniMax Music 2.6, Mureka V8, Suno V5, and Suno V5.5, judged by professional human raters. Each pair is evaluated under anonymized and randomized presentation, and values indicate the percentage of expert votes preferring each system.

Qwen-Music is designed around four key principles:

- **Composition in a compact semantic music space.** Instead of directly modeling waveforms, Qwen-Music represents music as 25 Hz single-codebook Music Semantic Tokens. This compact stream gives the autoregressive model a tractable sequence space for full-song composition while preserving the musical semantics needed by the renderer.
- **Explicit melody planning and control.** To bridge the gap between high-level textual intent and concrete musical realization, Qwen-Music-LLM makes melody an explicit intermediate representation. For text-to-music generation, Melody-CoT plans the vocal melody before full-token generation. For cover generation, reference melody tokens condition the sequence so the output follows the source melody under the new style and vocal attributes.



	Range	Creator	Model	Elo	95% CI	Samples
1	1	Suno	Suno V5.5	1,170	−7/7	8,999
2	2	Mureka	Mureka V8	1,137	−7/7	9,975
3	3	JazzCat	JazzCat	1,116	−13/13	2,555
4	4–6	Suno	Suno V5	1,095	−7/7	9,699
5	4–7	MiniMax	MiniMax Music 2.5+	1,094	−7/7	9,456
6	4–7	MiniMax	Music 2.6	1,091	−7/7	9,178
7	5–8	Suno	Suno V4.5	1,087	−7/7	8,261
8	7–8	Google	Lyria 3 Pro	1,082	−7/7	9,340
9	9	Producer.ai	FUZZ-2.0	1,059	−7/7	8,995
10	10–11	ElevenLabs	Eleven Music	1,022	−7/7	8,962
11	10–11	Producer.ai	FUZZ-2.0 Raw	1,020	−7/7	8,940
12	12–11	Producer.ai	FUZZ-1.1 Pro	1,000	0/0	8,222
13	13–14	Udio	Udio v1.5 Allegro	968	−7/7	8,267
14	13–14	Sonauto	Sonauto V2.1	963	−7/7	8,846

Figure 3: External leaderboard result on the Artificial Analysis Music with Vocals Leaderboard. Qwen-Music participated as `JazzCat` and ranked among the leading English vocal music generation systems.

- **Generative acoustic rendering.** Qwen-Music-Render treats Music Semantic Tokens as musical sketches for generative rendering. A semantic-conditioned DiT predicts acoustic latents, the Spec-VAE decoder reconstructs spectrograms, and the Band-Mode Refiner corrects frequency-dependent magnitude and phase detail before waveform synthesis.
- **Quality-graded scaling and multi-stage preference alignment.** To effectively exploit large-scale heterogeneous music corpora, Qwen-Music organizes pre-training around a quality-graded data curriculum, progressively learning from data at different quality levels. It further adopts a multi-stage post-training pipeline that combines supervised learning, offline preference optimization, and online policy optimization to improve musicality, instruction following, and controllability.

Empirically, Qwen-Music demonstrates strong competitiveness against leading proprietary music generation systems. As shown in Figure 2, blind A/B preference tests judged by professional human raters show that Qwen-Music achieves win rates of 59.1% against MiniMax Music 2.5+, 66.7% against MiniMax Music 2.6, 58.3% against Mureka V8, and 55.4% against Suno V5, while remaining comparable to Suno V5.5 with a slight preference advantage of 50.3%. Beyond our internal evaluation, Qwen-Music also appears as `JazzCat` on the Artificial Analysis Music with Vocals Leaderboard[1], ranking third among leading English vocal music generation systems (Figure 3). Objective evaluation in Section 4 further shows that Qwen-Music obtains the best result on 13 out of 16 text-to-music metrics across SongBench (Wu et al., 2026), SongEval (Yao et al., 2025), and AudioBox-Aesthetic (Tjandra et al., 2025), while cover song generation results demonstrate strong reference-melody preservation, stylistic adaptation, and vocal controllability.

Overall, Qwen-Music provides a unified framework for controllable, high-fidelity, and structurally coherent music generation. It bridges semantic-level composition and waveform-level synthesis, enabling both open-ended song creation and reference-guided musical reinterpretation in a single, coherent system.

[1] https://artificialanalysis.ai/music/leaderboard/vocals

2 Architecture

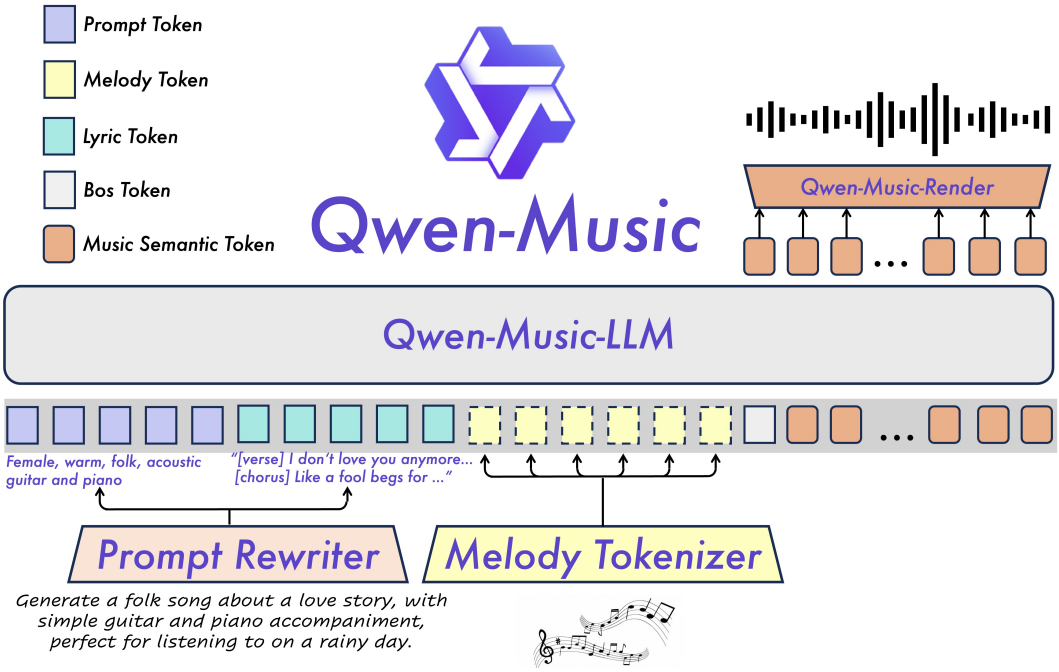


Figure 4: Overview of the Qwen-Music inference pipeline. Given a user request, Qwen-Music first rewrites the natural-language description into a structured textual condition, including musical tags (such as genre, singer characteristics, instrumental arrangement, etc.) and generated lyrics. Qwen-Music-LLM then generates Music Semantic Tokens, optionally using melody tokens extracted from a reference song for melody cloning. Finally, Qwen-Music-Render takes both the rewritten textual condition and the generated Music Semantic Tokens as input and performs generative rendering to produce high-fidelity 48 kHz stereo waveforms.

2.1 Overview

As shown in Figure 4, Qwen-Music follows an inference pipeline that transforms a user request into a complete high-fidelity stereo song. Given a natural-language description, the system first rewrites it into a structured textual condition containing musical tags (such as genre, singer characteristics, instrumental arrangement, etc.) and generated lyrics. Conditioned on this rewritten text, Qwen-Music-LLM autoregressively predicts Music Semantic Tokens, optionally incorporating reference melody tokens via Melody-CoT conditioning for cover song generation. Qwen-Music-Render then takes both the rewritten textual condition and the generated Music Semantic Tokens as input and performs generative rendering to produce high-fidelity 48 kHz stereo waveforms. This pipeline is supported by three core components: Qwen-Music-Tokenizer, Qwen-Music-LLM, and Qwen-Music-Render.

- **Qwen-Music-Tokenizer** maps raw music audio into 25 Hz Music Semantic Tokens using a Conformer-based encoder. It is trained with a four-stage recipe that combines self-supervised learning, causal adaptation, multi-task supervision, and vector quantization. The resulting token stream provides the discrete music representation used for language-model training and renderer conditioning.
- **Qwen-Music-LLM** generates Music Semantic Tokens conditioned on text descriptions, lyrics, musical attributes, and optional melody tokens. For original song generation, it can first produce a Melody-CoT plan before generating the full Music Semantic Token sequence. For cover song generation, it uses melody tokens extracted from a reference song as conditioning context while following the target style and vocal attributes.
- **Qwen-Music-Render** converts generated Music Semantic Tokens into 48 kHz stereo waveforms. It uses a semantic-conditioned Diffusion Transformer to predict continuous acoustic latents, a spectral-domain VAE decoder to reconstruct complex spectrograms, and a Band-Mode Refiner to correct frequency-dependent magnitude and phase details before inverse STFT synthesis.

In the following sections, we first introduce Qwen-Music-Tokenizer and its multi-stage training recipe. We then describe Qwen-Music-LLM, including melody tokenization and melody-aware generation. Finally, we present Qwen-Music-Render, focusing on diffusion-based latent generation, spectral-domain decoding, and band-wise refinement.

2.2 Qwen-Music-Tokenizer

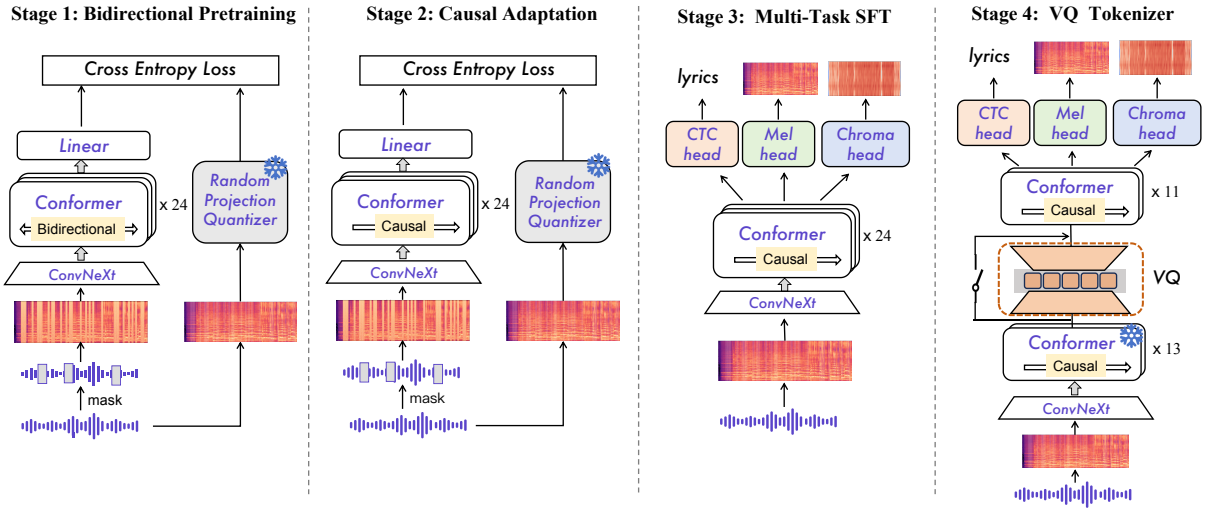


Figure 5: Overview of Qwen-Music-Tokenizer. The tokenizer maps music waveforms to 25 Hz Music Semantic Tokens through BestRQ pretraining, causal adaptation, multi-task SFT, and VQ tokenizer training.

Qwen-Music-Tokenizer is a low-bitrate discrete tokenizer that maps a raw music waveform into a single stream of 25 Hz Music Semantic Tokens. It uses a single backbone consisting of a convolutional frontend followed by a Conformer encoder (Gulati et al., 2020). As illustrated in Figure 5, we train the backbone in *four* stages: (i) bidirectional self-supervised pretraining with BestRQ (Chiu et al., 2022), (ii) causal adaptation of the pretrained encoder, (iii) multi-task supervised fine-tuning (SFT) with lyric and spectral targets, and (iv) VQ tokenizer training. The final stage inserts the VQ bottleneck and trains the final tokenizer. Stages 2–4 are initialize from the previous checkpoint, while Stage 1 starts from random initialization. This recipe first learns continuous music representations and then introduces causal attention, supervision, and vector quantization. The deployed tokenizer keeps the causal encoder up to the VQ insertion point and produces the 25 Hz token stream consumed by downstream autoregressive models. Table 1 summarizes the recipe.

2.2.1 Backbone Architecture

The tokenizer takes raw music audio as input and encodes it into a compact low-frame-rate token sequence. Specifically, each audio clip is converted to a mono 24 kHz waveform and represented using log-Mel spectrogram features. A lightweight convolutional frontend reduces the frame rate to 25 Hz, corresponding to one frame every 40 ms. The tokenizer backbone is a 24-layer, 0.6B-parameter Conformer (Gulati et al., 2020), which is shared across all four training stages. Starting from Stage 2, the backbone is converted to causal self-attention to match the downstream autoregressive modeling setting. When encoding audio, the tokenizer is applied offline to the full clip and produces a compact 25 Hz token stream.

2.2.2 Stage 1: Bidirectional BestRQ Pretraining

We begin with a self-supervised pretraining stage to build a general-purpose music representation from unlabeled audio. The encoder is trained *bidirectionally* with BestRQ (Chiu et al., 2022), a masked-prediction objective whose targets are produced by a *frozen* random quantizer, removing the need for a separately learned target network.

Random-projection targets Target codes are formed from the clean log-Mel spectrogram: a local context window is pooled with temporal stride 4 so that targets align with the 25 Hz encoder frames, projected by a fixed random projection, and mapped to the nearest entry of a fixed, ℓ_2-normalized random codebook.

The projection and codebook are randomly initialized and kept frozen throughout training, so no target network is learned.

Masking and objective We apply time-domain span masking: the waveform is partitioned into 0.4 s spans, each masked independently with probability 0.3 (guaranteeing at least one span per sample), and masked spans are replaced with low-variance Gaussian noise. The encoder consumes the masked audio, and a lightweight prediction head is trained to recover the frozen random-projection targets at masked frames under a cross-entropy loss,

$$\mathcal{L}_{\text{BestRQ}} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \text{CE}\big(p(h_t), q(x_t)\big). \tag{1}$$

where $\mathcal{M}$ is the set of masked frames, h_t is the encoder output at frame t, $q(\cdot)$ the frozen random-projection quantizer, and $p(\cdot)$ the prediction head.

2.2.3 Stage 2: Causal Adaptation

To prevent the encoder from accessing future frames and to align its representations with downstream autoregressive modeling, we restrict self-attention to be causal. Rather than training a causal encoder from scratch, we resume BestRQ pretraining from the Stage 1 checkpoint after switching self-attention from full bidirectional masking to causal (left-context-only) masking, while leaving the pretraining objective otherwise unchanged. This short adaptation stage converts the encoder to causal self-attention at a small fraction of the Stage 1 cost, while preserving much of the representation learned under the less constrained bidirectional regime. All subsequent stages inherit causal attention.

2.2.4 Stage 3: Multi-Task Supervised Fine-Tuning

Before quantization, we use multi-task supervision to make the encoder representations more informative for both language and music. In particular, this stage encourages the future discrete codes to preserve lyrical content together with musical semantic attributes such as melody and harmony.

We attach three lightweight heads to the 25 Hz encoder outputs: (i) a linear *CTC head* (Graves et al., 2006) that transcribes lyrics over a multilingual subword vocabulary; (ii) a ConvNeXt *Mel head* that upsamples the latents back to the 100 Hz, 128-bin Mel spectrogram; and (iii) a ConvNeXt *chroma head* predicting the 12-bin chroma feature. The objective combines the CTC loss with spectral-reconstruction losses on Mel and chroma, each a sum of a spectral-convergence term and a magnitude term,

$$\mathcal{L}_{\text{SFT}} = \lambda_{\text{ctc}}\,\mathcal{L}_{\text{CTC}} + \lambda_{\text{mel}}\,\mathcal{L}_{\text{spec}}^{\text{mel}} + \lambda_{\text{chr}}\,\mathcal{L}_{\text{spec}}^{\text{chroma}}, \qquad \lambda_{\text{ctc}}{=}\lambda_{\text{mel}}{=}\lambda_{\text{chr}}{=}1. \tag{2}$$

Input masking is effectively disabled ($p_{\text{mask}}{=}0.01$) so the model sees near-clean audio, and training operates on full songs (up to 300 s, i.e. $\sim$7.5k latent frames at 25 Hz) rather than fixed crops. We initialize from the Stage 2 checkpoint.

2.2.5 Stage 4: Vector-Quantized Tokenizer

The final stage discretizes the encoder by inserting a single VQ bottleneck at an intermediate layer of the 24-layer Conformer stack. The activation at that layer is quantized, and the remaining upper layers and the SFT heads then operate on the quantized stream, forcing the codes to retain the information required for lyric transcription and spectral reconstruction.

Quantizer We insert a single VQ codebook of 32768 entries under a cosine metric, trained with a straight-through estimator and a commitment loss. The quantizer incorporates dedicated mechanisms that keep essentially the entire codebook active throughout training, so that codebook utilization reaches 99%+ and dead codes are virtually eliminated even at 2^{15} entries; the full code space therefore contributes to the bitrate budget. A single codebook of size 2^{15} at 25 Hz corresponds to 25 tokens/s and a bitrate of 375 bit/s.

Smooth insertion and staged unfreezing Injecting a hard quantizer into a converged encoder is unstable, so we blend the quantized and continuous paths through a scalar gate, $h_{\text{out}} = h + \alpha\,(h_q - h)$, where α ramps linearly from 0 to 1 over an initial warmup. Concurrently, we first freeze the frontend and all layers up to and including the insertion point and train only the quantizer, the upper layers, and the heads, then unfreeze the whole model. The loss is the SFT objective plus the VQ loss, with the CTC weight lowered to 0.5 (Mel, chroma, and VQ weights 1.0). We initialize from the Stage 3 checkpoint.

Table 1: Overview of the four-stage training recipe for Qwen-Music-Tokenizer. All stages share the 0.6B-parameter Conformer backbone (24 layers, width 1024, 16 heads, 25 Hz latents) and are initialized from the previous checkpoint.

Stage	Objective	Attention	Audio length	Init. from
1	BestRQ masked prediction	bidirectional	30 s crop	random
2	BestRQ (causal adaptation)	causal	30 s crop	Stage 1
3	CTC + Mel + chroma SFT	causal	full song ($\leq$ 300 s)	Stage 2
4	SFT + VQ	causal	full song ($\leq$ 300 s)	Stage 3

Tokenization. At inference, extracting codes requires only the frontend, the Conformer layers up to the insertion point, and the quantizer; the post-VQ layers and all heads are discarded. Each 40 ms frame maps to a single integer in $[0, 32768)$, giving the 25 Hz token stream consumed by downstream models.

2.3 Qwen-Music-LLM

Qwen-Music-LLM adopts an autoregressive backbone initialized from a 3B dense variant of Qwen3.5-Omni (Team, 2026). It models music in the discrete semantic-token space produced by Qwen-Music-Tokenizer. Given the output of the prompt rewriter, including global music tags and structured lyrics, the model predicts a sequence of Music Semantic Tokens that captures the long-range composition, vocal phrasing, accompaniment, and section-level musical development of the target song. These generated tokens are then passed to Qwen-Music-Render for high-fidelity waveform synthesis.

A central challenge in text-to-music generation is that text prompts typically specify style and lyrics only at a high level, without explicitly determining the melodic contour or compositional structure of the song. Directly predicting full-mixture Music Semantic Tokens from text therefore requires the model to learn composition and arrangement simultaneously. To address this, we introduce **Melody-CoT**, a melody-token-based chain-of-thought for explicit melody planning: before generating full-mixture Music Semantic Tokens, the model is trained to produce an intermediate sequence of melody tokens that describes a coarse-grained vocal melody contour. This intermediate melody plan encourages the LLM to first resolve the compositional structure of the song and then render that plan into the full Music Semantic Token sequence.

Melody-CoT also provides a unified interface for text-to-music and cover song generation. For text-to-music generation, Qwen-Music-LLM conditions only on tags and lyrics and directly generates Music Semantic Tokens, optionally producing the melody plan as an intermediate step. For cover song generation, melody tokens extracted from a reference song are inserted into the prefix together with the target tags and lyrics. The model then generates new full-mixture Music Semantic Tokens that follow the reference melody contour while allowing timbre, arrangement, and genre to be controlled by the textual conditions.

2.3.1 Melody Tokenizer

The Melody Tokenizer converts a reference vocal pitch contour into a compact sequence of discrete melody tokens. We first extract a 50 Hz vocal pitch curve using RMVPE (Wei et al., 2023). Since the goal is to preserve the compositional melody rather than fine-grained singing expression, the pitch curve is downsampled by a factor of 8 with median pooling, yielding a 6.25 Hz melody sequence. This low frame rate suppresses local ornaments, vibrato, and other performance-level details while retaining the coarse contour needed for melody conditioning.

To prevent the melody condition from leaking absolute pitch range, key, or singer-dependent timbre information, we use a **relative** MIDI representation. Specifically, we convert the downsampled pitch values from Hertz to MIDI pitch values and then compute the median MIDI value over all voiced frames in the sequence and subtract it from every voiced frame. The resulting relative semitone offsets are clipped to a fixed range and mapped to a 256-entry vocabulary, with one dedicated token reserved for unvoiced frames. As a result, the melody tokens encode the shape of the vocal melody but discard most absolute-register information, encouraging the generated song to obey the target tags and lyrics for style, timbre, and arrangement.

We also explored chromagram-based representations (Copet et al., 2024; Zhang et al., 2026) for melody tokenization. However, our preliminary experiments indicate that chromagrams often retain substantial information beyond the leading melody, including residual background and acoustic details of the arrangement. Such extra information can make the conditioning signal too strong and reduce the model's ability to follow the global music tags. We therefore adopt pitch-contour-based melody tokens as a

cleaner representation of the reference melody.

Formally, let $\mathbf{f} = (f_1, \ldots, f_T)$ denote the RMVPE pitch curve at 50 Hz. We first apply $8\times$ median-pooling downsampling,

$$\tilde{\mathbf{f}} = \mathrm{MedianPool}_8(\mathbf{f}), \tag{3}$$

where unvoiced frames are ignored when computing the median and a pooled frame is set to 0 if the corresponding window is mostly unvoiced. The melody token z_i is then computed as

$$\begin{aligned} \bar{m} &= \mathrm{median}\{\mathrm{round}(\mathrm{hz2midi}(\tilde{f}_i)) \mid \tilde{f}_i > 0\}, \\ z_i &= \begin{cases} \mathrm{clip}(\mathrm{round}(\mathrm{hz2midi}(\tilde{f}_i)) - \bar{m},\, -127,\, 127) + 127, & \tilde{f}_i > 0, \\ 255, & \tilde{f}_i = 0, \end{cases} \end{aligned} \tag{4}$$

where $\tilde{f}_i$ is the downsampled pitch value, $\bar{m}$ is the global voiced MIDI median, and $z_i \in [0, 255]$ is the final melody token.

2.3.2 Melody-CoT Training

Qwen-Music-LLM is trained with a next-token prediction objective over sequences that may contain text tokens, melody tokens, and Music Semantic Tokens. The text portion, consisting of tags and lyrics, is used as conditioning context. Loss is applied to both the Melody-CoT region and the final Music Semantic Token region, so the model learns not only to generate realistic Music Semantic Tokens, but also to construct an explicit melody plan from textual conditions.

We mix multiple sequence patterns during training to support different inference requirements within a single model. The plain text-to-music pattern is

$$[\text{text},\ \text{music semantic tokens}],$$

where the model directly generates Music Semantic Tokens from tags and lyrics. The melody-planning patterns insert Melody-CoT before the Music Semantic Token sequence:

$$[\text{text},\ \text{section-level melody},\ \text{music semantic tokens}]$$

and

$$[\text{text},\ \text{unique-section-level melody},\ \text{music semantic tokens}].$$

Balancing these patterns allows the model to retain strong text-to-music capability while also learning to use explicit melody conditions for cover song generation.

The **section-level** Melody-CoT pattern writes melody tokens for vocal sections that contain lyrics. Each melody segment is preceded by its section label, such as `[verse]` or `[chorus]`, and non-vocal sections such as `intros`, `instrumental breaks`, `outros`, and `silence` are omitted from the Melody-CoT sequence. This design weakens the melody condition compared with a full-song pitch trace and avoids exposing the exact full-song duration through the length of the melody sequence.

The **unique-section-level** pattern further reduces over-conditioning. Instead of including every occurrence of a repeated section label, it groups candidate segments by section type and samples one representative melody segment for each unique label during training. For example, multiple chorus occurrences are represented by one sampled chorus melody. This shortens the Melody-CoT sequence, improves data diversity across epochs, and encourages the model to use melody as a reusable compositional guide rather than copying the reference song frame by frame.

2.4 Qwen-Music-Render

The language model captures long-range musical structure in a compact discrete token space. We introduce *Qwen-Music-Render*, a three-stage neural render that turns this discrete output into a high-fidelity waveform. The first stage is a *Diffusion Transformer* (DiT) (Peebles & Xie, 2023; Liu et al., 2023; Lipman et al., 2023) that, conditioned on the discrete tokens together with the textual musical tags and lyrics, produces a continuous latent through conditional flow matching. The second stage is *Spec-VAE*, whose Spec Decoder inverts the latent sequence back to a coarse complex STFT. The third stage is a *Band-Mode Refiner* that predicts frequency-adaptive magnitude and phase residuals on top of the decoder output; a final inverse STFT produces the 48 kHz stereo waveform.

2.4.1 Diffusion Transformer

The Music Semantic Tokens generated by the language model provide a discrete representation of the song's global structure, section ordering, and vocal/instrumental arrangement. Conditioned on these tokens, the Diffusion Transformer (DiT) predicts the latent representation, which is subsequently decoded by the downstream Spec Decoder into a listenable spectrogram.

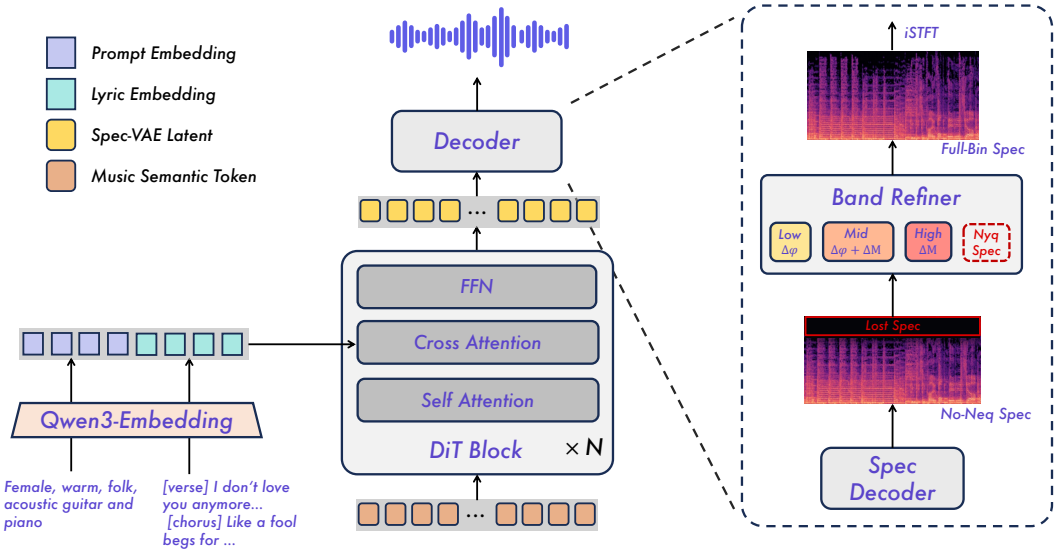


Figure 6: Overview of Qwen-Music-Render. Music Semantic Tokens and text conditions guide a semantic-conditioned DiT that predicts acoustic latents. Spec-VAE decodes the latents into complex spectrograms, and the Band-Mode Refiner corrects band-dependent magnitude and phase details before inverse STFT synthesis to 48 kHz stereo waveforms.

Architecture Our backbone is a 1.3 B DiT operating on continuous latents. It comprises 32 Transformer blocks with hidden dimension $d = 1024$, 24 attention heads, and a feed-forward expansion ratio of 8. Rotary positional embeddings (RoPE) (Su et al., 2024) are applied to the queries and keys in self-attention. Each block consists of a self-attention layer, a cross-attention layer that attends to the auxiliary conditioning context described below, and a feed-forward layer, all modulated by AdaLN parameters derived from the diffusion timestep. At each forward pass, the network takes as input a noisy latent $\phi_t \in \mathbb{R}^{T\times m}$ with $m = 128$, together with a frame-aligned token sequence $\mathbf{c} \in \{1, \ldots, V\}^T$ generated by the language model at 25 Hz.

Conditioning In Qwen-Music-Render, we further provide the DiT with the rewritten textual condition produced by the prompt rewriter, in addition to the Music Semantic Tokens generated by Qwen-Music-LLM. We find that this additional textual conditioning improves audio quality and helps the renderer better follow the intended song specification. Specifically, we use a frozen Qwen3-Embedding-0.6B (Zhang et al., 2025) text encoder to extract features from the textual condition, and inject them into the DiT through cross-attention. The song-description tokens are projected to the DiT hidden width with a bias-free linear layer, while the lyric tokens are further processed by a 6-layer RoPE Transformer encoder trained jointly with the DiT. The resulting description and lyric representations are concatenated into a single cross-attention context $\mathcal{C}$ attended by each DiT block. To enable classifier-free guidance (CFG) at inference time, we introduce a learnable null context $\mathcal{C}_\varnothing$ for the text-conditioning branch. During CFG inference, the conditional pass uses both the Music Semantic Tokens and the full text context $\mathcal{C}$, while the unconditional pass keeps the Music Semantic Tokens unchanged and replaces $\mathcal{C}$ with $\mathcal{C}_\varnothing$.

Training details We train the model in two stages. First, the DiT is pretrained on a large-scale music corpus. It is then supervised fine-tuned (SFT) on a smaller set of curated, lossless-quality recordings, which further improves output audio quality. Training samples are limited to tracks of up to 6 minutes, corresponding to at most 9,000 frames at 25 Hz. The cross-attention context is truncated to 256 tokens for the song description and 1536 tokens for the lyrics. We use AdamW for optimization, together with a linear warm-up schedule and gradient clipping at 1.0.

2.4.2 Spec-VAE & Refiner

Architecture Spec-VAE follows the 2D-conv architecture of SpectroStream (Li et al., 2025), with discrete residual vector-quantized tokens replaced by a continuous spectral latent to match the latent diffusion framework in Section 2.4.1. Stereo audio at 48 kHz is transformed by STFT into a complex spectrogram $\mathbf{X} \in \mathbb{C}^{2\times 480\times T}$. A 7-block Spec Encoder compresses the spectrogram into 128-dimensional latents at 25 Hz, achieving a total compression ratio of 192×, while a mirrored Spec Decoder reconstructs a coarse

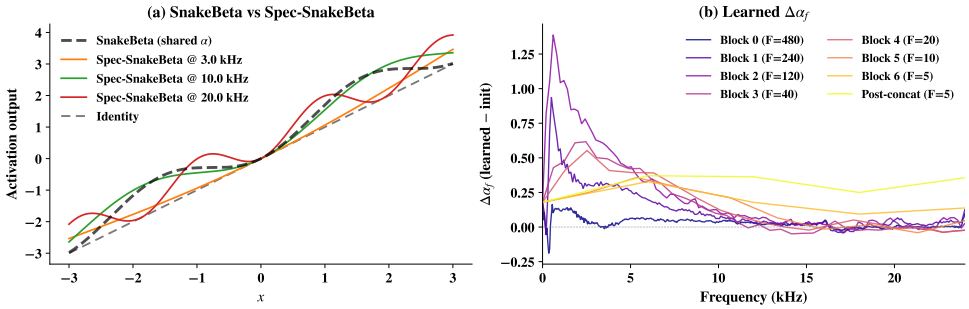


Figure 7: Spec-SnakeBeta analysis. **(a)** Activation curves: the standard SnakeBeta (black dashed) applies a single shared α to all frequency bins; Spec-SnakeBeta adapts the periodic modulation per frequency, producing distinct nonlinear shapes at 0.5, 3, 10, and 20 kHz. **(b)** Deviation of learned α_f from log-frequency initialization ($\Delta\alpha_f = \alpha_{\text{learned}} - \alpha_{\text{init}}$) across encoder layers—positive values indicate the network increases periodic modulation beyond the physical prior, with the strongest deviations concentrated in 0–5 kHz.

complex spectrogram for iSTFT synthesis. Stereo channels are handled with delayed fusion in the encoder and early splitting in the decoder. To further improve reconstruction quality, we attach a lightweight Band-Mode Refiner after the decoder to correct residual spectral artifacts.

Spec-SnakeBeta We introduce *Spec-SnakeBeta*, a frequency-aware extension of the SnakeBeta activation used in BigVGAN (Lee et al., 2023). While the original SnakeBeta uses channel-wise learnable parameters, Spec-SnakeBeta parameterizes α and β along the frequency axis, exploiting the fact that each STFT bin corresponds to a fixed physical frequency:

$$\text{Spec-SnakeBeta}(\mathbf{x})_{b,c,t,f} = x_{b,c,t,f} + \frac{1}{\exp(\beta_f) + \epsilon} \sin^2\left(x_{b,c,t,f} \cdot \exp(\alpha_f)\right). \tag{5}$$

We initialize α_f with a log-frequency prior,

$$\alpha_f = \log\left(\frac{\text{freq}_f}{\bar{f}} + \epsilon\right), \qquad \bar{f} = \frac{1}{F}\sum_f \text{freq}_f, \tag{6}$$

which encourages stronger periodic modulation at higher frequencies. As shown in Figure 7, the learned parameters preserve this frequency-dependent structure while adapting most strongly in the low-to-mid frequency range. This frequency-dependent parameterization provides the activation prior used by the Spec-VAE decoder.

Band-Mode Refiner The Band-Mode Refiner is a lightweight ConvNeXt-1D module (Liu et al., 2022) that refines the decoder's complex spectrogram output. Motivated by the complementary error patterns of magnitude and phase across frequency bands, it applies band-specific corrections: phase-only correction in the low-frequency band, joint magnitude and phase correction in the mid-frequency band, and magnitude-only correction in the high-frequency band. This design improves spectral fidelity while preserving stable optimization: at initialization, all refiner weights are zero, so the refined output is exactly equal to the decoder output.

Training Objective and Discriminators The decoder system is trained with a combination of multi-resolution STFT loss (Yamamoto et al., 2020), IF/GD phase loss (Wang et al., 2025), LSGAN adversarial loss with feature matching (Mao et al., 2017; Larsen et al., 2016), and KL regularization. We further adopt K-weighting perceptual filtering (ITU-R, 2015), MSLR stereo decomposition (Wang et al., 2025), adaptive log-magnitude normalization from SAME (Liao et al., 2025), and mixed-scale spectral losses from SpectroStream (Li et al., 2025). For adversarial training, we use complementary STFT, CQT, and spectral-domain discriminators to provide both waveform-level and spectral-domain feedback.

Three-Stage Training Pipeline As summarized in Table 2, training proceeds in three stages. Stage 1 performs reconstruction-only Spec-VAE pretraining. Stage 2 introduces waveform-domain adversarial training. Stage 3 freezes the Spec-VAE encoder–decoder and trains the Band-Mode Refiner with both waveform and spectral discriminators. This staged recipe stabilizes training under the 192× compression setting and progressively improves fine-grained acoustic detail.

Table 2: Spec-VAE and refiner training configuration across three stages. Stage 1 is reconstruction-only pretraining. Stage 2 introduces waveform-domain adversarial training. Stage 3 freezes the encoder–decoder and trains the refiner with waveform (**W**) and spectral (**S**) discriminators.

Category	Param	Stage 1	Stage 2	Stage 3
Optimizer	Generator optim.	Muon	Muon	AdamW
	Generator LR	10^{-4}	1.5×10^{-4}	1.5×10^{-4}
	Disc optim	—	Muon	AdamW
	Disc LR	—	3×10^{-5}	10^{-6} (W) / 10^{-5} (S)
Schedule	Batch size / GPU	4	1	1
	Segment length	1.28 s	1.28 s	1.28 s
	Disc warmup step	—	0	0 (W) / 15k (S)
	G:D interleave	—	1:1	4:1
Architecture	STFT disc scales	—	8	8
	CQT disc	✗	✓	✓
	Spectral disc	✗	✗	✓
Loss weights	λ_{STFT}	1.0	1.0	0.5
	$\lambda_{\text{IF\_GD}}$	0	0	0.1
	λ_{KL}	10^{-6}	10^{-6}	0
	$\lambda_{\text{time-GAN}}$ (Adv/FM)	—	1.0 / 100	0.175 / 20
	$\lambda_{\text{spec-GAN}}$ (Adv/FM)	—	—	0.5 / 35

2.4.3 Acoustic Data Quality Pipeline

We train Qwen-Music-Render only on files whose acoustic evidence matches a genuine high-fidelity music target. Raw collections often contain lossless containers transcoded from MP3, duplicated-mono "stereo" tracks, clipped or over-limited masters, and upsampled files whose high-frequency band is empty. We therefore apply the automated pipeline summarized in Table 3 before renderer training and evaluation.

Table 3: Metric-level checklist used by the acoustic pipeline. Each row pairs a rule-engine identifier with a brief description and its pass/flag condition.

Check	**Metric ID**	**Evidence**	**Condition**
Meta & encoder	`encoder_tag`	Encoder string in container/stream tags	flag if `lavf/lavc/ffmpeg/lame`
	`mutagen_tags`	Metadata sweep for conversion traces	flag if `transcode/convert`
Bitrate	`has_non_monotonic_pts`	PTS monotonicity	pass if `false`
	`bitrate_utilization`	Realized / nominal bitrate	pass if ≥ 0.6
	`top10_bitrate_mean`	Complex-segment mean bitrate	pass if $\geq 1.2\times$ mean
	`vbr_variation`	Bitrate coefficient of variation	pass if $\sigma/\mu \geq 0.15$
	`quality_tier`	Bitrate-based quality level	HiRes/High/Med/Low/Trash
Loudness	`lufs_i`	Integrated loudness (EBU R128)	$[-25, -5]$ LUFS
	`true_peak`	dBTP level	$[-1, +2]$ dBFS
	`lra`	Loudness range	$[3, 15]$ LU
	`plr`	Peak-to-loudness ratio	≥ 6 LU
Fake stereo	`lr_diff_zero_ratio`	Fraction of zero $\lvert L - R \rvert$ samples	pass if < 0.99
Freq cutoff	`cutoff_ratio`	Cutoff / Nyquist freq ratio	ranking from $0.5 \rightarrow 1.0$
	`drop_db`	Energy gap around f_c	hard ≥ 40 dB; suspicious ≥ 20 dB
	`valid_segments`	Non-silent interior FFT windows	pass if > 0

Noise-floor-aware cutoff detection Container metadata alone cannot distinguish true high-resolution content from an upsampled or lossy-transcoded file. The key idea is to estimate the noise floor from track *edges* and detect the spectral cutoff from *interior* segments, keeping the two estimates independent.

We reserve the first and last 10% of the mono waveform as edge segments and compute Hann-windowed 8192-point RFFTs over non-overlapping windows. After peak-normalizing and averaging their magnitude spectra, the empirical noise floor is

$$\hat{\eta} = 20 \log_{10}\big(\text{median}(\bar{\mathbf{M}}_{\text{edge}}[\text{top 10\% bins}])\big). \tag{7}$$

For the cutoff, we process five equally spaced interior windows that skip near-silent regions and average the resulting spectra. The detected cutoff f_c is the highest FFT bin whose mean magnitude remains above

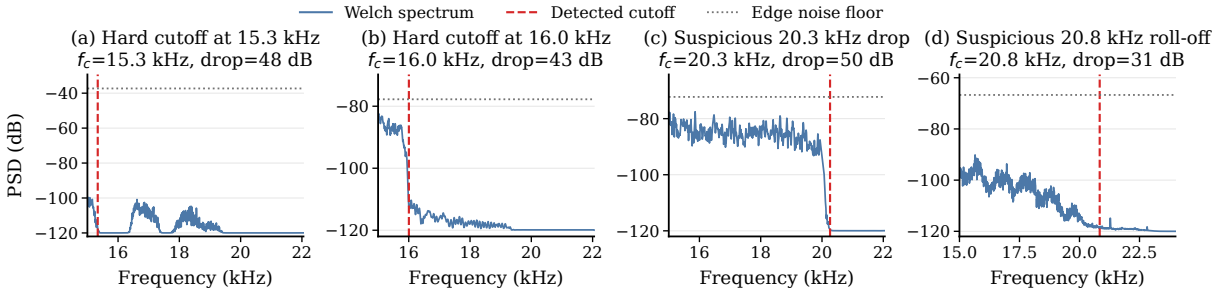


Figure 8: High-frequency cutoff detector examples. The blue curve is the Welch spectrum of a 10 s verification segment, the red dashed line is the detected cutoff f_c, and the gray dotted line is the independently estimated edge noise floor.

-80 dB relative to the peak. We then measure a local energy drop across a ± 2 kHz band around f_c:

$$\Delta_{\mathrm{dB}} = \max \tilde{\mathbf{M}}_{\mathrm{int}}[f_c - 2\,\mathrm{kHz} : f_c] - \max \tilde{\mathbf{M}}_{\mathrm{int}}[f_c : f_c + 2\,\mathrm{kHz}], \tag{8}$$

where the computation replaces the post-cutoff maximum with $\hat{\eta}$ when no reliable content exists above f_c. The final label depends on both the cutoff ratio f_c / f_{Nyq} and the drop magnitude. The rule marks a file as `hard_cutoff` when the ratio falls below 0.85 and the drop reaches $\Delta_{\mathrm{dB}} \geq 40$ dB, or ≥ 25 dB near $\hat{\eta}$. It marks a file as `suspicious` when the ratio is below 0.90 with $\Delta_{\mathrm{dB}} \geq 20$ dB. Files that meet neither condition are labeled `natural`.

Figure 8 visualizes four representative outcomes. The first two panels show classic lossy-to-lossless signatures: a sharp wall well below Nyquist and post-cutoff energy collapsed to the noise floor. The last two panels are closer to Nyquist and therefore require the noise-floor-aware drop test; the detector flags them only when the roll-off is both low enough in frequency and steep enough in level.

3 Training

The remaining training pipeline targets Qwen-Music-LLM. Renderer-specific acoustic filtering appears in Section 2.4.3; here we train the backbone LLM through quality-aware pre-training and progressive post-training.

3.1 Quality-Graded Pre-training Curriculum

To effectively exploit large-scale music corpora with heterogeneous quality distributions, Qwen-Music adopts a quality-graded pre-training curriculum for the backbone LLM. Instead of simply discarding all lower-quality data, we organize the corpus into quality levels and schedule them progressively, allowing the model to first learn broad musical coverage and then concentrate on higher-quality generation.

Following the multi-aspect song-quality assessment protocols of SongBench (Wu et al., 2026) and SongEval (Yao et al., 2025), we train an internal MOS-based reward model. This model is trained on data annotated by human raters with professional music backgrounds, and its predicted scores are used to estimate and rank the general musical quality of each audio sample. Within each genre, we rank samples and partition them into seven quality buckets (Q1–Q7) using percentile thresholds at 90%, 75%, 50%, 25%, 5%, and 1%. Q1 denotes the highest-quality bucket, and Q7 denotes the lowest-quality bucket. This genre-normalized stratification reduces bias toward dominant styles and balances quality modeling across musical categories. We remove Q7 and use the remaining Q1–Q6 subsets for model training.

As described in Section 2.3, Qwen-Music-LLM is initialized from a 3B dense variant of Qwen3.5-Omni (Team, 2026), whose language-model backbone provides useful prior knowledge of language and music. We then train the model with a three-stage curriculum that gradually moves from broad data coverage to high-quality refinement.

Stage 1: General pre-training In the first stage, we train on Q3–Q6 data, which accounts for the majority of the over 5 million-hour corpus. The objective is to learn a general mapping from text conditions, including lyrics and musical tags, to music semantic tokens. To improve both robustness and generation quality, we use a dynamic quality sampling strategy in which lower-quality data is emphasized early in training and gradually replaced by higher-quality data as training progresses. We further apply language balancing to mitigate cross-lingual data imbalance and include 20% instrumental music to strengthen accompaniment modeling.

Stage 2: Annealing In the second stage, we train on Q2-level data with a learning-rate decay schedule. This stage serves as a quality consolidation phase, helping the model transition from broad data coverage to more structured and stable music generation. It improves musical structure, coherence, and overall audio quality while reducing artifacts introduced by noisier data in the first stage.

Stage 3: High-quality refinement In the final stage, we focus on Q1 data and additionally incorporate carefully selected high-quality samples from the full corpus. This stage further improves musicality, controllability, and instruction following. We maintain balanced genre and language distributions to avoid overfitting to dominant styles and to ensure stable generation under diverse control conditions.

3.2 Multi-Stage Post-training Alignment

After pre-training, we further align Qwen-Music-LLM with human musical preferences and user instructions through a multi-stage post-training pipeline. The pipeline optimizes two complementary objectives: improving musicality, measured by the internal musicality MOS predictor described above, and enhancing instruction following, including genre control and lyric following, with rewards computed by Qwen3.5-Omni (Team, 2026) and Qwen3-ASR (Shi et al., 2026). We organize post-training from stable to exploratory: supervised learning first establishes a clean generation prior, offline preference optimization then improves alignment in a controlled manner, and online policy optimization finally unlocks further gains in musicality and audio quality.

Phase I: Supervised Cold Start We first perform supervised fine-tuning on carefully curated high-quality data. This stage selects samples that are both aligned with human preference and reliably annotated with control information, especially genre tags and lyrics. By emphasizing high-quality music examples with accurate style labels and lyric annotations, the model establishes a clean generation prior with strong controllability before reward-based optimization begins.

Phase II: Iterative Offline Preference Alignment Building on the SFT model, we perform iterative offline alignment with Direct Preference Optimization (DPO) (Rafailov et al., 2023). In each iteration, the current policy generates rollouts under diverse prompts and control conditions, and the musicality MOS predictor together with instruction-following reward models score these outputs. We then construct preference pairs from the scored candidates and optimize Qwen-Music-LLM with DPO. Repeating this rollout–rewarding–pair construction–optimization loop progressively improves musicality and controllability while producing a policy that is more stable and better aligned for subsequent online optimization.

Phase III: Online Musicality Optimization Finally, we apply on-policy Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025). Compared with offline DPO, GSPO enables stronger exploration by updating the policy directly from on-policy samples and sequence-level rewards. Because the model has already acquired reliable generation ability and instruction-following behavior in the first two phases, this stage can focus on fine-grained musicality and audio-quality improvement rather than correcting fundamental alignment issues.

Together, these three phases provide a stable optimization path for Qwen-Music-LLM, yielding stronger musicality, better controllability, and more reliable instruction following.

4 Evaluation

Qwen-Music is evaluated on two core tasks: text-to-music generation and cover song generation. In blind A/B preference tests conducted by professional human raters (Figure 2), Qwen-Music outperforms MiniMax Music 2.6, MiniMax Music 2.5+, Mureka V8, and Suno V5, while remaining competitive with Suno V5.5. Genre-wise Bradley–Terry analysis further shows strong preference across diverse musical styles (Table 4), and the Artificial Analysis leaderboard provides an external reference point for its performance among leading English vocal music generation systems (Figure 3). Beyond full-song generation, we also evaluate Qwen-Music-Render through generative rendering ablations, acoustic reconstruction benchmarks, and refiner-specific analyses. Together, these evaluations show that Qwen-Music combines strong musicality, controllability, lyric intelligibility, cover-song melody preservation, and high-fidelity stereo rendering.

4.1 Evaluation of Music Generation

We evaluate Qwen-Music on two music generation tasks: **text-to-music generation** and **cover song generation**. Text-to-music generation measures the ability to create complete songs from a user's natural-

Table 4: Genre-wise Bradley–Terry ratings from blind A/B preference tests conducted by professional human raters. Ratings are computed independently within each genre and should be interpreted only for within-genre comparison. Higher ratings indicate stronger preference. Best results are highlighted in **bold**, and second-best results are underlined.

Genre	Qwen-Music	Suno V5.5	Suno V5	Mureka V8	Minimax Music 2.6	Minimax Music 2.5+
Electronic / EDM	**1140**	880	853	938	759	866
Hip-Hop / Rap	1020	**1205**	1018	898	866	930
Jazz & Blues	**1112**	961	968	202	868	930
Metal & Hard Rock	1039	880	1058	**1070**	859	942
Pop	1020	1000	980	**1050**	977	880
Punk & Hardcore	**1147**	863	759	918	1120	202
R&B / Soul	**1056**	1032	950	1000	853	880
Rock	**1116**	859	782	961	720	1027

language prompt or request. Cover song generation measures the ability to reinterpret a reference song or melody according to the user's desired target style and vocal direction.

Baseline Systems For text-to-music generation, we compare Qwen-Music against five state-of-the-art proprietary music generation systems: Suno V5.5, Suno V5, Mureka V8, MiniMax Music 2.6, and MiniMax Music 2.5+. For cover song generation, we retain the leading proprietary services that support this capability: Suno V5.5, Suno V5, and MiniMax Cover. Since the Suno systems do not accept real-world songs as reference inputs, comparisons with Suno are limited to the AI-generated reference set.

Evaluation Metrics We use complementary objective metrics to evaluate musicality, audio quality, controllability, intelligibility, and melody preservation. For general musical quality, SongBench Wu et al. (2026) scores melody, arrangement, musicality, vocal quality, instrumental quality, mixing, and structure, with higher values indicating better performance. For text-to-music generation, we also report SongEval Yao et al. (2025) and AudioBox-Aesthetic Tjandra et al. (2025) as additional music-aesthetic evaluation frameworks. For lyric intelligibility and lyric following, Qwen3-ASR (Shi et al., 2026) transcribes the generated audio and we compute the Phoneme Error Rate (PER) against the input lyrics; lower PER is better, and we report PER after multiplying by 100. For tag following, Gemini 3.1 Pro Comanici et al. (2025) rates each generated song on a 1–10 scale across genre, mood, instrument, vocal gender, and vocal timbre, where higher scores indicate better adherence to the target musical tags. For cover song generation, we further measure reference-melody preservation using Melody MAE. Specifically, we convert both the generated cover and the reference song into relative MIDI representations following Eq. 4 in Section 2, align the two melody sequences with dynamic time warping, and compute the mean absolute error in semitones. Lower Melody MAE indicates stronger preservation of the reference melody.

4.1.1 Text-to-Music Generation

Subjective Evaluation To complement the automatic evaluation, we conduct a blind A/B preference test between Qwen-Music and competing systems, with judgments collected from professional human raters. We use the same generated audio samples as in the objective evaluation. For each prompt and each comparison system, the audio generated by Qwen-Music and the corresponding baseline are paired and presented in random order.

A total of 50 professional human raters with music creation or production experience are recruited. In each trial, raters listen to two anonymized audio samples without knowing their model identities or generation sources, and select the one they prefer based on overall listening experience. To reduce positional bias, the order of the two samples is randomized for each trial. Each A/B pair is independently judged by three different experts, and final preference rates are aggregated over all expert votes.

As shown in Figure 2, Qwen-Music is preferred over all comparison systems on average. It achieves a 59.1% win rate against MiniMax Music 2.5+, 66.7% against MiniMax Music 2.6, 58.3% against Mureka V8, and 55.4% against Suno V5. The closest comparison is against Suno V5.5, where Qwen-Music still obtains a slight preference advantage of 50.3% versus 49.7%, indicating comparable subjective quality to this strong commercial baseline.

To further analyze robustness across musical styles, we group the test samples according to their genre tags and compute genre-wise Bradley–Terry ratings from the A/B preference outcomes. Ratings are estimated independently within each genre based on pairwise outcomes between Qwen-Music and each

Table 5: Objective text-to-music generation results on a bilingual test set containing 300 Chinese and 300 English examples. Musicality and audio quality are evaluated using SongBench, SongEval, and AudioBox-Aesthetic. Tag following is evaluated by Gemini 3.1 Pro. Higher is better except for PER. Best results are shown in **bold**, and second-best results are underlined.

Metrics	Qwen-Music	Suno V5.5	Suno V5	Mureka V8	Minimax Music 2.6	Minimax Music 2.5+
			SongBench			
Melody	**7.03**	6.70	7.01	6.98	6.64	6.41
Arrangement	**7.35**	7.03	7.20	7.14	6.72	6.45
Musicality	**6.22**	5.83	6.12	6.05	5.67	5.50
Vocal	**7.44**	6.95	7.37	7.38	6.99	6.89
Instrumental	**7.24**	6.85	7.03	7.01	6.64	6.52
Mixing	**7.13**	6.88	7.07	6.97	6.59	6.38
Structure	6.94	6.83	6.98	**7.02**	6.55	6.27
			SongEval			
Coherence	**4.55**	4.28	4.44	4.49	4.33	4.28
Musicality	**4.42**	4.12	4.30	4.36	4.19	4.15
Memorability	**4.51**	4.27	4.43	**4.51**	4.29	4.23
Clarity	**4.45**	4.17	4.33	4.37	4.23	4.17
Naturalness	**4.37**	4.10	4.27	**4.37**	4.16	4.08
			AudioBox-Aesthetic			
Content Enjoyment	**7.47**	7.35	7.40	7.31	7.39	7.42
Content Usefulness	**7.86**	7.85	7.75	7.67	7.82	7.83
Production Complexity	6.64	6.67	**6.71**	6.67	6.64	6.49
Production Quality	8.15	8.08	8.08	7.95	8.15	**8.20**
			Tags Following			
Genre	8.08	8.33	8.28	**8.46**	8.01	7.60
Moods	8.94	8.93	**8.98**	8.67	8.57	8.20
Instruments	8.65	8.80	**8.89**	8.66	8.15	7.79
Vocal Gender	**7.85**	7.54	7.58	7.77	7.57	7.48
Vocal Timbre	8.68	**8.71**	8.67	8.38	8.31	8.58
			Intelligibility			
PER	6.10	**4.19**	6.80	9.04	6.40	6.29

comparison system, and therefore should not be compared across different genres.

As shown in Table 4, Qwen-Music achieves the highest Bradley–Terry rating in five out of eight genres, including Electronic/EDM, Jazz & Blues, Punk & Hardcore, R&B/Soul, and Rock. It also ranks second in Hip-Hop/Rap and Pop, and remains competitive in Metal & Hard Rock. These results suggest that Qwen-Music maintains strong subjective preference across diverse musical styles, rather than showing advantages only in a narrow set of genres.

Figure 3 provides an additional external leaderboard reference. On the Artificial Analysis Music with Vocals Leaderboard, Qwen-Music participates as `JazzCat` and ranks in the top tier, reaching third place among leading English vocal music generation systems as shown in the figure. Together with the professional-rater A/B preference test and genre-wise Bradley–Terry analysis, this result further supports the strong and stable perceptual quality of Qwen-Music.

Objective Evaluation As shown in Table 5, we conduct a comprehensive objective evaluation on a test set containing 300 Chinese and 300 English samples. To ensure a fair comparison, all systems are evaluated under identical input conditions, using AI-generated lyrics and musical tags with a balanced distribution across genres.

Qwen-Music achieves strong performance across musicality and audio-quality metrics. Across SongBench, SongEval, and AudioBox-Aesthetic, it obtains the best result in 13 out of 16 evaluated dimensions. On SongBench, Qwen-Music ranks first in six out of seven dimensions, including melody, arrangement, musicality, vocal quality, instrumental quality, and mixing. On SongEval, it consistently ranks first across all five dimensions: coherence, musicality, memorability, clarity, and naturalness. On AudioBox-Aesthetic, Qwen-Music achieves the best results in content enjoyment and content usefulness while remaining competitive on production-related metrics. These results indicate strong overall generation quality in both composition and rendering.

Qwen-Music also demonstrates competitive controllability and lyric intelligibility. On tag following, it

Table 6: Objective cover song generation results on the AI-generated reference-melody set, containing 100 English and 100 Chinese examples. The Qwen-Music (section) column uses section-level melody conditioning, while the Qwen-Music (unique section) column uses unique-section-level melody conditioning. Higher is better except for PER and Melody MAE. Best results are shown in **bold**, and second-best results are underlined.

Metric	Qwen-Music (section)	Qwen-Music (unique section)	Suno V5.5	Suno V5	MiniMax Cover
			SongBench		
Melody	6.86	6.85	7.13	**7.28**	6.64
Arrangement	7.27	7.25	7.56	**7.61**	6.70
Musicality	6.05	6.04	6.36	**6.37**	5.72
Vocal	7.30	7.29	7.35	**7.47**	7.14
Instrumental	7.31	7.26	7.28	**7.34**	6.74
Mixing	6.96	6.94	7.29	**7.33**	6.66
Structure	6.83	6.80	7.19	**7.27**	6.56
			Intelligibility		
PER (↓)	19.28	17.47	**5.77**	8.46	25.35
			Melody Following		
Melody MAE (↓)	**1.48**	1.80	2.00	1.87	1.89
			Tag Following		
Genre	5.92	7.34	**8.27**	7.14	5.42
Mood	7.32	8.51	**8.77**	7.92	7.19
Instruments	7.55	8.42	**9.14**	8.35	7.22
Vocal Gender	7.96	8.89	**9.38**	9.16	6.35
Vocal Timbre	7.77	8.69	**9.23**	8.53	7.88

achieves an average score of 8.44 across five control dimensions, within 0.04 points of the best-performing system on average. It performs particularly well on fine-grained vocal control, achieving the best result on vocal gender following and ranking second on both mood and vocal timbre control. Although Mureka V8 and Suno V5 obtain slightly higher scores on genre and instrument control respectively, Qwen-Music remains competitive across all dimensions, indicating balanced controllability across different musical attributes.

For intelligibility, Qwen-Music achieves a PER of 6.10, the second-lowest among all systems. Overall, these results show that Qwen-Music offers a strong balance among musicality, audio quality, controllability, and lyric intelligibility.

4.1.2 Cover Song Generation

We evaluate cover song generation under reference-melody conditioning. Given a reference melody, target musical tags, and lyrics, the model is expected to generate a new song that preserves the reference melody while following the requested style and vocal attributes. We construct two evaluation sets. The first uses reference songs generated by AI music models, including Suno V5.5 and Mureka V8, with 100 English and 100 Chinese examples. The second uses real-world popular songs as reference melodies, also with 100 English and 100 Chinese examples. In the tables, *section-level melody* denotes conditioning with melody tokens for each lyric-bearing section, while *unique-section-level melody* denotes conditioning with one representative melody for each repeated section label.

On the AI-generated reference set (Table 6), section-level Melody-CoT achieves the lowest Melody MAE, indicating stronger direct reference-melody cloning, while unique-section-level conditioning substantially improves tag following over the section-level mode. Commercial systems remain strong on several general musicality and tag-following metrics. On the real-world popular-song set (Table 7), Qwen-Music outperforms MiniMax Cover across most objective dimensions. The unique-section-level mode yields better musicality, intelligibility, and tag following on most metrics, whereas the section-level mode gives the best direct melody following. These results show that the two Melody-CoT modes provide complementary control trade-offs for cover song generation.

4.2 Evaluation of Qwen-Music-Render

Qwen-Music-Render performs generative rendering from Music Semantic Tokens and the rewritten textual condition into high-fidelity 48 kHz stereo waveforms. This section evaluates two key aspects of this rendering module. We first study the DiT rendering pipeline, focusing on the effects of rewritten

Table 7: Objective cover song generation results on the real-world popular-song reference-melody set, containing 100 English and 100 Chinese examples. The Qwen-Music (section) column uses section-level melody conditioning, while the Qwen-Music (unique section) column uses unique-section-level melody conditioning. Higher is better except for PER and Melody MAE. Best results are shown in **bold**, and second-best results are underlined.

Metric	Qwen-Music (section)	Qwen-Music (unique section)	MiniMax Cover
		SongBench	
Melody	6.55	**6.57**	6.18
Arrangement	6.84	**6.86**	6.16
Musicality	5.75	**5.79**	5.30
Vocal	7.07	**7.09**	6.73
Instrumental	6.96	**7.00**	6.41
Mixing	6.66	**6.67**	6.25
Structure	**6.48**	6.46	6.13
		Intelligibility	
PER (↓)	20.20	**18.49**	22.48
		Melody Following	
Melody MAE (↓)	**1.44**	1.80	1.76
		Tag Following	
Genre	6.90	**7.35**	5.84
Mood	7.48	**8.05**	7.05
Instruments	8.05	**8.30**	7.23
Vocal Gender	8.88	**9.18**	6.80
Vocal Timbre	8.21	**8.54**	7.75

textual conditioning, CFG strategy, and latent backbone choice. We then evaluate whether the Band-Mode Refiner improves acoustic reconstruction quality when paired with Spec-VAE. Overall, the results show that rewritten textual conditioning with *text-drop* CFG gives the strongest generation quality, and that the refiner consistently improves spectral, mel-scale, and stereo reconstruction metrics.

4.2.1 Effect of Text Conditioning and CFG in DiT Rendering

Table 8 studies three design choices in the DiT rendering pipeline: the latent backbone used as the prediction target, the rewritten textual condition provided to the renderer, and the null-branch design used for classifier-free guidance (CFG). "None" indicates no text conditioning, while frame-aligned LM token conditioning is still provided.

The results show that rewritten textual conditioning provides useful guidance beyond the song-level scaffold carried by LM tokens: under the matched Spec-VAE + LM-token drop setting, musical tags improve over no-text conditioning, and adding lyrics further improves most metrics. Among CFG variants, *Text-drop* CFG achieves the best overall performance, supporting our design choice in Section 2.4.1: LM tokens should be retained as the primary rendering scaffold, while musical tags and lyrics serve as complementary semantic guidance. Under matched text-conditioning and CFG settings, Spec-VAE also generally outperforms Levo 2 VAE (Lei et al., 2026b), indicating that a stronger latent representation leads to better final rendering quality. Overall, the best configuration combines **Spec-VAE, musical tags + lyrics, and *Text-drop* CFG**.

4.2.2 Does the Band-Mode Refiner Improve Acoustic Reconstruction?

We next isolate the acoustic decoder from LM generation and evaluate whether the Band-Mode Refiner improves reconstruction fidelity. Spec-VAE and Spec-VAE + Refiner are evaluated on the full 546 tracks from the Song Describer Dataset (Manco et al., 2023), together with four recent open-source audio VAE systems: εar-VAE (Wang et al., 2025), Stable Audio Open (SA-Open) (Evans et al., 2024), Levo 2 (Lei et al., 2026b), and SAME-L (Parker et al., 2026). All systems are evaluated at their native operating conditions. We report the baseline (**Spec-VAE**) and the full pipeline with Band-Mode Refiner (**Spec-VAE + Refiner**) separately to isolate the contribution of the refiner.

Metrics We measure reconstruction quality along three axes: spectral fidelity, mel-scale frequency response, and stereo image preservation. **(1) Spectral reconstruction:** We use SI-SDR (dB, ↑) (Le Roux et al., 2019), STFT Distance (↓), and $\text{STFT}_{\text{log1p}}$ (↓) to measure time-domain fidelity, broadband spectral

Table 8: DiT rendering ablation on our bilingual EN/ZH set (100 samples each). We compare latent backbones, rewritten textual conditioning, and CFG null-branch designs. For fair comparison, Spec-VAE results here bypass the Band-Mode Refiner. "None" indicates no text conditioning, while frame-aligned LM token conditioning is always provided. *Text-drop*, *Full-drop*, and *LM-token-drop* respectively remove text conditioning only, both text and LM token conditioning, and LM token conditioning only from the null branch. Best results are highlighted in **bold**, and second-best results are underlined.

Latent Backbone	Levo 2 VAE	Levo 2 VAE	Spec-VAE	Spec-VAE	Spec-VAE	Spec-VAE
Text Conditioning	None	Tags + Lyrics	Tags only	Tags + Lyrics	Tags + Lyrics	Tags + Lyrics
CFG Strategy	LM-token drop	LM-token drop	LM-token drop	Full-drop	LM-token drop	Text-drop
SongBench						
Melody	6.55	6.56	6.88	6.89	6.88	**6.96**
Arrangement	6.87	6.88	7.15	7.17	7.17	**7.25**
Musicality	5.66	5.71	6.04	6.05	6.04	**6.17**
Vocal	6.80	6.88	7.21	7.29	7.30	**7.37**
Instrumental	6.96	6.95	7.08	7.20	7.19	**7.21**
Mixing	6.55	6.59	6.89	**7.02**	**7.02**	7.01
Structure	6.36	6.38	6.69	6.75	6.75	**6.83**
SongEval						
Coherence	4.28	4.31	4.44	4.38	4.39	**4.47**
Musicality	4.11	4.17	4.28	4.25	4.25	**4.34**
Memorability	4.18	4.24	4.38	4.33	4.33	**4.42**
Clarity	4.14	4.20	4.33	4.29	4.29	**4.37**
Naturalness	4.05	4.10	4.22	4.18	4.18	**4.26**

deviation, and low-amplitude spectral details. **(2) Mel reconstruction:** We use Mel Distance (↓) and $\text{MEL}_{\text{log1p}}$ (↓) to evaluate reconstruction quality on a perceptually spaced frequency axis. **(3) Stereo image quality:** We use CCPC (Cross-Channel Phase Coherence; $\in [0, 1]$, ↑) (Wang et al., 2025) and Spectral Pan Error (↓). CCPC evaluates inter-channel phase coherence, while Spectral Pan Error measures the L1 distance in the frequency-wise stereo pan profile $\text{pan}(f) = |L(f)|^2/(|L(f)|^2 + |R(f)|^2)$ (Avendano & Jot, 2002).

Table 9: Objective reconstruction quality on the Song Describer Dataset. All models are evaluated at their native operating conditions. Subscript values denote 95% confidence intervals computed over the per-file metrics. Best results are highlighted in **bold**, and second-best results are underlined.

Metric	**ϵar-VAE**	**SA-Open**	**Levo 2**	**SAME-L**	**Spec-VAE**	**Spec-VAE +Refiner**
STFT-Scale Reconstruction						
SI-SDR ↑	12.4 ±0.4	6.7 ±0.3	8.1 ±0.3	**12.5** ±0.4	10.9 ±0.4	11.3 ±0.4
STFT Dist ↓	0.880 ±0.014	1.016 ±0.016	0.971 ±0.016	0.986 ±0.015	0.916 ±0.016	**0.870** ±0.014
$\text{STFT}_{\text{log1p}}$ ↓	0.079 ±0.005	0.089 ±0.005	0.086 ±0.005	0.079 ±0.005	0.078 ±0.005	**0.075** ±0.005
Mel-Scale Reconstruction						
Mel Dist ↓	0.509 ±0.008	0.612 ±0.007	0.599 ±0.008	0.539 ±0.010	0.572 ±0.010	**0.461** ±0.006
$\text{MEL}_{\text{log1p}}$ ↓	0.095 ±0.006	0.106 ±0.006	0.103 ±0.006	0.096 ±0.006	0.093 ±0.006	**0.089** ±0.005
Stereo Image Quality						
CCPC ↑	**0.973** ±0.003	0.933 ±0.004	0.947 ±0.004	0.970 ±0.003	0.966 ±0.003	**0.973** ±0.002
Spectral Pan Err ↓	0.267 ±0.004	0.276 ±0.004	0.273 ±0.004	0.276 ±0.004	0.268 ±0.004	**0.264** ±0.004

Results Table 9 summarizes the reconstruction results. The full rendering decoder (**Spec-VAE + Refiner**) achieves the best STFT Distance (0.870), $\text{STFT}_{\text{log1p}}$ (0.075), Mel Distance (0.461), $\text{MEL}_{\text{log1p}}$ (0.089), and Spectral Pan Error (0.264), while matching the best CCPC score of 0.973. Compared with Spec-VAE alone, the refiner improves all reported metrics, with the largest gain on Mel Distance (0.572 to 0.461). These results support the band-specific correction strategy and show that the refiner improves perceptual frequency response and stereo spatial fidelity without sacrificing overall reconstruction quality.

5 Conclusion

In this report, we presented Qwen-Music, a unified large-scale music generation system for both text-to-music generation and reference-melody-based cover song generation. Qwen-Music connects

semantic-level composition with high-fidelity audio synthesis through three core components: **Qwen-Music-Tokenizer**, which represents music as compact 25 Hz single-codebook Music Semantic Tokens; **Qwen-Music-LLM**, which performs melody-aware autoregressive music modeling with a Melody-CoT mechanism; and **Qwen-Music-Render**, which performs generative stereo rendering from both Music Semantic Tokens and rewritten textual conditions. Together, these components enable controllable full-song generation with coherent lyrics, stable musical structure, realistic vocals, and high-fidelity stereo waveforms.

To train this system at scale, we introduced a **quality-graded pre-training curriculum** over more than 5 million hours of multilingual music data, allowing Qwen-Music-LLM to first learn broad musical coverage and then progressively refine generation quality. We further adopted a **multi-stage post-training alignment** pipeline that combines supervised learning, iterative offline DPO, and online GSPO, improving musicality, instruction following, and controllability. For Qwen-Music-Render, acoustic quality filtering provides reliable high-fidelity supervision for 48 kHz stereo rendering.

Extensive evaluations demonstrate that Qwen-Music achieves strong performance across subjective preference studies, genre-wise Bradley–Terry analysis, external leaderboard comparison, and objective metrics. In text-to-music generation, Qwen-Music shows competitive or superior performance against leading proprietary systems in musicality, audio quality, controllability, and lyric intelligibility. In cover song generation, it demonstrates strong reference-melody preservation and stylistic adaptation, while render-specific evaluations confirm the benefits of rewritten textual conditioning, text-drop CFG, and Band-Mode Refiner for high-fidelity stereo reconstruction.

Qwen-Music represents a step toward unified, scalable, and controllable music generation systems that bridge long-range musical semantics and waveform-level acoustic realization. Future work will explore more flexible long-context musical structure modeling, finer-grained expressive control over singing and performance, and more efficient rendering architectures for high-quality music generation.

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.

Carlos Avendano and Jean-Marc Jot. Frequency domain techniques for stereo to multichannel upmix. In *AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, 2002.

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)*, pp. 1206–1210. IEEE, 2024.

Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3915–3924. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/chiu22a.html.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv:2210.13438*, 2022.

Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. *arXiv preprint arXiv:2407.14358*, 2024.

Junmin Gong, Sean Zhao, Sen Wang, Shengyuan Xu, and Joe Guo. Ace-step: A step towards music generation foundation model. *arXiv preprint arXiv:2506.00045*, 2025.

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

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 369–376, 2006. doi: 10.1145/1143844.1143891. URL https://doi.org/10.1145/1143844.1143891.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Proc. Interspeech 2020*, pp. 5036–5040, 2020. doi: 10.21437/Interspeech.2020-3015. URL https://www.isca-archive.org/interspeech_2020/gulati20_interspeech.html.

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.

ITU-R. BS.1770-4: Algorithms to measure audio programme loudness and true-peak audio level. Technical report, International Telecommunication Union, 2015.

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.

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.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pp. 1558–1566, 2016.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. SDR – half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 626–630, 2019.

Sang-gil Lee, Wei Ping, Boris Ginsburg, Bryan Catanzaro, and Sungroh Yoon. BigVGAN: A universal neural vocoder with large-scale training. In *International Conference on Learning Representations*, 2023.

Shun Lei, Yixuan Zhou, Boshi Tang, Max WY Lam, Hangyu Liu, Jingcheng Wu, Shiyin Kang, Zhiyong Wu, Helen Meng, et al. 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, 2026a.

Shun Lei, Huaicheng Zhang, Dapeng Wu, Yaoxun Xu, Lishi Zuo, Wei Tan, Hangting Chen, Guangzheng Li, Jianwei Yu, Zhiyong Wu, and Dong Yu. Levo 2: Stable and melodious song generation via hierarchical representation modeling and progressive post-training, 2026b. URL https://arxiv.org/abs/2606.30642.

Yuzhe Li et al. SpectroStream: Spectral-domain neural audio codec. *arXiv preprint arXiv:2508.05207*, 2025.

Guanrou Liao et al. SAME: Scaling audio with masked experts. *arXiv preprint arXiv:2605.18613*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 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.

Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 564–572, 2024.

Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, Elio Quinton, György Fazekas, and Juhan Nam. The song describer dataset: a corpus of audio captions for music-and-language evaluation. In *Machine Learning for Audio Workshop at NeurIPS 2023*, 2023.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Ziqian Ning, Huakang Chen, Yuepeng Jiang, Chunbo Hao, Guobin Ma, Shuai Wang, Jixun Yao, and Lei Xie. Diffrhythm: Blazingly fast and embarrassingly simple end-to-end full-length song generation with latent diffusion. *arXiv preprint arXiv:2503.01183*, 2025.

Julian D. Parker, Zach Evans, CJ Carr, Zachary Zukowski, Josiah Taylor, Matthew Rice, and Jordi Pons. Same: A semantically-aligned music autoencoder, 2026. URL https://arxiv.org/abs/2605.18613.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

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

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Qwen Team. Qwen3.5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, et al. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*, 2025.

Kangdi Wang et al. Back to ear: Perceptually driven high fidelity music reconstruction. *arXiv preprint arXiv:2509.14912*, 2025.

Haojie Wei, Xueke Cao, Tangpeng Dan, and Yueguo Chen. RMVPE: A robust model for vocal pitch estimation in polyphonic music. In *INTERSPEECH*, pp. 5421–5425. ISCA, 2023.

Dapeng Wu, Shun Lei, Wei Tan, Guangzheng Li, Yunzhe Wang, Huaicheng Zhang, Lishi Zuo, and Zhiyong Wu. Songbench: A fine-grained multi-aspect benchmark for song quality assessment. *arXiv preprint arXiv:2604.25937*, 2026.

Yaoxun Xu, Hangting Chen, Jianwei Yu, Wei Tan, Shun Lei, Zhiwei Lin, Rongzhi Gu, and Zhiyong Wu. Mucodec: Ultra low-bitrate music codec for music generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 689–698, 2025.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6199–6203, 2020.

Dongchao Yang, Yuxin Xie, Yuguo Yin, Zheyu Wang, Xiaoyu Yi, Gongxi Zhu, Xiaolong Weng, Zihan Xiong, Yingzhe Ma, Dading Cong, et al. Heartmula: A family of open sourced music foundation models. *arXiv preprint arXiv:2601.10547*, 2026.

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.

Xueyao Zhang, Junan Zhang, Yuancheng Wang, Chaoren Wang, Yuanzhe Chen, Dongya Jia, Zhuo Chen, and Zhizheng Wu. Vevo2: A unified and controllable framework for speech and singing voice generation. *IEEE ACM Trans. Audio Speech Lang. Process.*, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

6 Authors

Core Contributors[2]

Jin Xu*	Shun Lei	Xueyao Zhang	Yongqi Wang	Zihan Liu
Kangdi Wang	Xiong Wang	Yang Zhang	Yue Wang	Zijian Lin
Ruibin Yuan	Xize Cheng	Yiheng Chen	Zhifang Guo	

Contributors[2]

Dake Guo	Linhan Ma	Xinfa Zhu	Yuanjun Lv	Zhiyong Wu
Hangrui Hu	Wei Xue	Xipin Wei	Yuxuan Wang	
Lei Xie	Wenxiang Guo	Yangze Li	Yunfei Chu	

[2]Alphabetical by given name; in the multi-column layout, read top-to-bottom and then left-to-right. * denotes the corresponding author.