HIGH-FIDELITY SONG GENERATION WITH JOINT DISCRETE AND CONTINUOUS MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating a full song from lyrics and a style description requires long-range musical planning without sacrificing vocal and instrumental detail. Low-rate discrete semantic tokens make this tractable, but quantization constrains the information available to acoustic rendering, while separately optimized stages prevent downstream generation objectives from shaping the planner. We propose **joint discrete and continuous modeling** for high-fidelity song generation. A VQ tokenizer supplies tokens for autoregressive planning, and a separately trained SSL-derived variational encoder supplies continuous semantic targets. Conditioned on the completed token plan and differentiable readouts of the planner's predictive states, a full-sequence flow model generates semantic features for acoustic rendering. Its flow-matching gradients reach the language model through the readouts, without differentiating through token sampling; the standard pathway keeps continuous features out of autoregressive feedback. On 192 shared prompts, its 48D configuration improves SongEval from 4.152 to 4.269 and reduces micro-PER from 20.26% to 14.47% relative to a discrete baseline, while receiving substantially higher expert listening scores. We further analyze the system through downstream semantic probes and conditioned reconstruction.

Code and demo: `https://github.com/JDCM12/JDCM12.github.io`

1 INTRODUCTION

Generating a full song from lyrics and a style description requires minute-scale organization of vocal melody, accompaniment, and form, as well as detailed acoustic realization. Discrete music representations provide an effective interface between high-level modeling and acoustic synthesis across song generation and music reconstruction (Borsos et al., 2022; Yang et al., 2025; Lei et al., 2025; Dai et al., 2026; Xu et al., 2024). In song generation, a language model predicts low-rate semantic tokens and a downstream model renders audio from their embeddings. This division makes long sequences manageable for the planner while leaving acoustic detail to the renderer.

The economy of a discrete interface also defines its limits. Vector quantization maps distinct self-supervised music features to the same token; looking up that token returns a continuous vector, but cannot recover distinctions lost at assignment. A second limitation concerns training. The planner learns next-token probabilities, whereas a downstream diffusion or flow model learns to synthesize audio. When sampled token IDs are the only connection, the synthesis objective does not update the language model parameters that produced them. The question is therefore how to enrich communication between planning and acoustic rendering without giving up the tractable discrete plan.

DisCon shows that discrete image predictions can guide generation in a continuous space (Zheng et al., 2025). We bring this principle to music through two aligned representation branches built on an SSL-initialized music encoder: a VQ tokenizer supplies the autoregressive discrete plan, and a separately trained variational encoder supplies continuous semantic targets. Lyric transcription, reconstruction, and teacher alignment shape the latter branch; its features are neither the VQ branch's prequantization features nor its codebook vectors.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our standard pathway retains discrete tokens as the language model's only autoregressive feedback. Aligned continuous readouts of its causal states, together with the completed token sequence, condition a full-sequence semantic flow; an acoustic renderer converts the generated features into audio. The flow-matching loss updates the planner through the readouts alongside token cross-entropy, without differentiating through hard token sampling. Thus, continuous features inform synthesis and training without entering the language model's autoregressive history.

Feeding generated continuous features back into later planning can introduce an additional exposure bias: subsequent steps use reference features during training but generated features at inference. Our interleaved Joint-SB configuration explores this alternative with feedback in the continuous *semantic* space. The standard pathway avoids this particular mismatch, although the language model still faces a train–inference mismatch in its discrete token history.

On 192 shared WildSongBench prompts, we evaluate six systems with objective quality and style measures, a transcription-derived phoneme-error score, and a listening study in which five music experts assess ten matched song groups. The joint family shows an overall advantage on principal objective and listening measures, while configurations trade off different strengths. MARBLE probes and source-conditioned reconstruction provide intermediate diagnostics alongside full-song evaluation. Our contributions are:

Joint Discrete and Continuous Modeling. We introduce a jointly trained approach to full-song generation that retains autoregressive planning with discrete semantic tokens while generating continuous semantic features, allowing the flow-matching objective to update the planner; across evaluated configurations, it shows an overall advantage over the discrete baseline on principal objective and expert-listening measures.

Comparing Generation Frameworks Within Our Method. Within the proposed joint approach, we compare semantic bottleneck widths, target spaces, and generation frameworks across complete systems, including full-sequence semantic synthesis, direct acoustic-latent prediction, and interleaved semantic feedback. Automatic measures of song quality and style alignment, together with expert ratings of sound quality and musicality, reveal trade-offs among these configurations.

Analysis Across Representation and Generation. We assess discrete and continuous semantic spaces with MARBLE probes, compare source-conditioned reconstruction on 100 in-house songs, and relate these diagnostics to objective and expert-listening results for complete songs. The differing outcomes offer empirical insight into semantic bottlenecks and generation configurations without assuming that reference-conditioned scores predict free generation.

2 Related Work

Discrete music representations. Jukebox models multiscale VQ-VAE codes autoregressively to produce multi-minute songs, including lyrics-conditioned singing (Dhariwal et al., 2020). AudioLM separates semantic tokens from acoustic codec tokens; MusicLM adopts a semantic–acoustic hierarchy for text-to-music generation, whereas MusicGen autoregressively predicts codec-token streams (Borsos et al., 2022; Agostinelli et al., 2023; Copet et al., 2023).

Lyrics-conditioned song systems. SongGen and YuE extend autoregressive token modeling to songs with lyrics (Liu et al., 2025; Yuan et al., 2025). HeartMuLa combines a song language model with a low-frame-rate music codec (Yang et al., 2026). LeVo models mixed and dual-track song tokens, and LeVo 2 uses mixed-token planning states to guide separate vocal and accompaniment token prediction (Lei et al., 2025; 2026). Qwen-Music predicts a 25-Hz, single-codebook semantic stream and uses a generative renderer to supply acoustic detail (Xu et al., 2026). SongBloom interleaves autoregressive sketches with diffusion refinement and feeds acoustic context into later generation (Yang et al., 2025). Dai et al. use hierarchical token planning before full-sequence flow matching in acoustic VAE latents (Dai et al., 2026).

Diffusion and continuous music generation. DiffRhythm generates full songs from lyrics and style with non-autoregressive latent diffusion; DiffRhythm 2 uses semi-autoregressive block flow matching (Ning et al., 2025; Jiang et al., 2025). ACE-Step combines diffusion with a compressed autoencoder and self-supervised representation alignment (Gong et al., 2025). AudioLDM 2 predicts

self-supervised audio features to condition latent diffusion, while MusicFlow cascades semantic and acoustic flows (Liu et al., 2023; Prajwal et al., 2024).

Discrete-guided continuous synthesis. DisCon is the closest conceptual precedent: a pretrained discrete image model supplies conditions for a separate continuous autoregressive generator, so generation is not restricted to codebook lookups (Zheng et al., 2025). In our setting, discrete tokens remain explicit prediction targets and also guide a full-sequence semantic flow. The flow additionally receives readouts of the states that predict those tokens, and its loss updates the same language model. Flow-SLM jointly learns speech tokens and continuous acoustic codec frames, with generated frames entering later causal context (Chou et al., 2025). Our standard pathway retains discrete-only autoregressive feedback; the interleaved variant studies feedback in semantic space, unlike the acoustic feedback in SongBloom.

Music tokenization and evaluation. MuCodec combines semantic and acoustic information in a quantized music codec, while MuSeC examines the reconstruction–predictability trade-off of music tokens (Xu et al., 2024; Chen et al., 2026). Our continuous target comes from a separate SSL-derived encoder rather than the VQ branch's codebook vectors. MARBLE probes reference-audio representations on music tasks (Yuan et al., 2023); SongBench and SongEval assess generated songs (Wu et al., 2026; Yao et al., 2025). We use both types of evaluation to distinguish feature utility from the behavior of the full generator.

3 Joint Discrete and Continuous Modeling

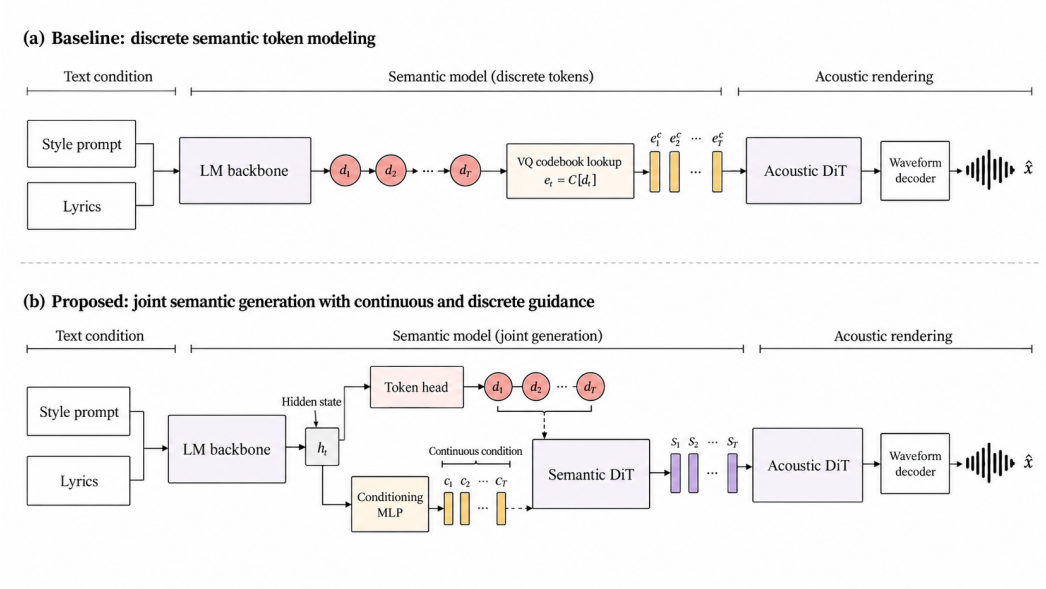


Figure 1: Discrete-token song generation and our extension. The baseline conditions acoustic rendering on codebook vectors; our model uses predicted tokens and language-model readouts to generate continuous semantic features. In the schematic, e_t^c is the looked-up vector e_t and s_t is a generated feature $\hat{s}_t$.

3.1 Song Generation with Discrete Semantic Tokens

Tokenization and Generation. Let p denote the text prompt formed from lyrics and a style description, and let x be a paired reference waveform. The discrete semantic tokenizer extracts low-rate continuous features $u = (u_1, \ldots, u_T)$ from x with an SSL-initialized music encoder (Chiu et al., 2022). A learned projection π maps each u_t into the space of a learned VQ codebook $C = \{C_k\}_{k=0}^{K-1}$, where K is the codebook size.

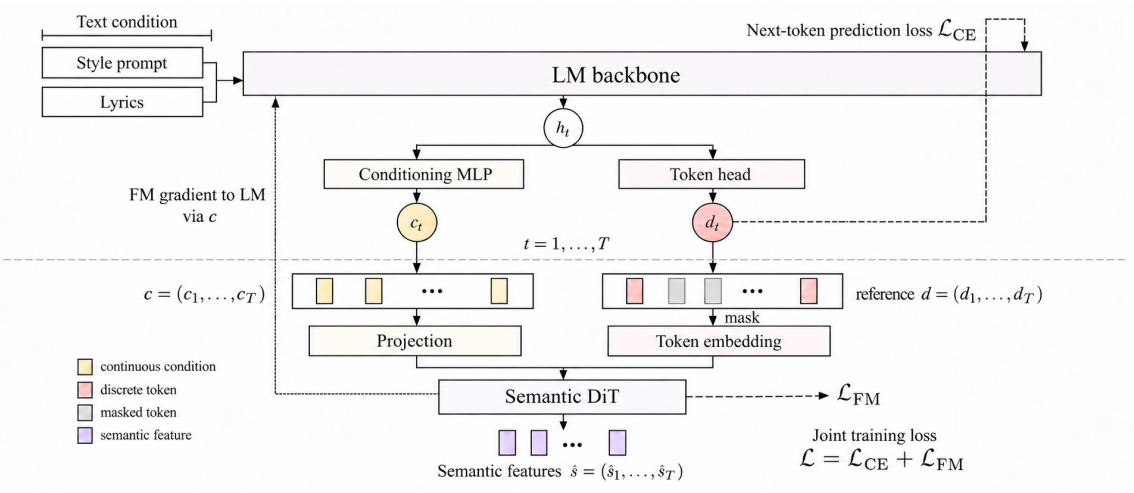


Figure 2: Joint next-token prediction and semantic flow training (Section 3.3). The token head is supervised to predict d_t; the Semantic DiT receives reference IDs with selected positions replaced by an unconditional ID (gray), yielding $\widetilde{d}$. Its continuous condition c remains intact. Flow-matching gradients reach the language model through c; $\hat{s}$ is a sampled semantic feature sequence, whereas Eq. 4 trains the velocity field.

L2 nearest-neighbor assignment (van den Oord et al., 2017) gives

$$d_t = \arg \min_{0 \le k < K} \|\pi(u_t) - C_k\|_2^2. \tag{1}$$

The index d_t is a discrete semantic token; lookup returns a continuous codebook vector $e_t = C_{d_t}$. The tokenizer is detailed in Appendix A.

An autoregressive language model predicts the semantic tokens $d = (d_1, \ldots, d_T)$ given p. With a beginning-of-sequence token implicit in the prefix, the probability of these T tokens factors as

$$p_{\text{LM}}(d_{1:T} \mid p) = \prod_{t=1}^{T} p_{\text{LM}}(d_t \mid p, d_{<t}). \tag{2}$$

Audio boundary tokens (BOS and EOS) are included in cross-entropy training; EOS terminates generation. At inference, the language model samples d, and codebook lookup gives $e = (e_1, \ldots, e_T)$ to a separately trained acoustic diffusion DiT. Starting from noise, the DiT generates acoustic latents that a decoder turns into a waveform (Figure 1a). The evaluated discrete baseline uses this token-to-codebook interface and a separately trained acoustic renderer.

Strengths and Limitations. The low token rate shortens the autoregressive sequence and supports song-level dependencies at practical training and inference cost. Codebook vectors condition the acoustic renderer, which synthesizes sound detail. Such staged systems have produced coherent, high-quality long-form songs.

The interface nevertheless has two limitations. First, vector quantization introduces representation error: distinct encoder features can receive the same token ID, and codebook lookup cannot recover the distinctions discarded by this assignment. The 25-Hz, single-codebook token sequence thus restricts the semantic information passed to the acoustic renderer.

Second, the language model is trained on reference-token cross-entropy, whereas the acoustic DiT is trained separately with a diffusion objective. The acoustic loss therefore does not backpropagate through the language model's predictive states to update its parameters, and token prediction is not jointly optimized for the fidelity of the final audio. Despite the musicality already achieved by this framework, generated songs may retain perceptible audio artifacts.

3.2 Connecting Discrete Planning to Continuous Semantics

We retain discrete tokens as the autoregressive plan and add a readout of the language model's predictive states as a continuous condition (Figure 1b). A semantic flow uses both conditions to generate features from a separately learned continuous target space before acoustic rendering. This design lets the semantic-generation objective update the language model while the flow targets are not limited to codebook vectors.

Continuous semantic tokenizer. In the standard configuration, we train an SSL-derived variational encoder q_ϕ, separately from the discrete tokenizer, that maps the reference waveform x to a distribution over continuous feature sequences $s = (s_1, \ldots, s_T) \in \mathbb{R}^{T \times D}$, where D is the feature width. Each $q_\phi(s_t \mid x)$ is Gaussian with an encoder-predicted mean and diagonal covariance. Samples $s \sim q_\phi(\cdot \mid x)$ are targets for the semantic flow. Their frames align with the discrete tokens d, but s_t is neither the prequantization feature u_t nor the codebook vector e_t; the discrete tokens are not obtained by quantizing s. We freeze this encoder before joint training (Appendix A).

The encoder is trained to retain several aspects of the reference audio in a compact sequence. Lyric-transcription supervision encourages vocal content; mel and chroma reconstruction constrain spectral and harmonic content; alignment encourages agreement with a frozen SSL teacher; and a KL term regularizes the posterior. Together these objectives define the continuous feature space used by the flow. They do not make the representation lossless, nor is the encoder trained to reconstruct a codebook vector. We use posterior samples as flow targets and posterior means for the reference-audio probes, keeping the training target and evaluation feature explicit.

Continuous condition from the language model. The conditional factor for d_t in Eq. 2 is parameterized by the state h_t after the language model reads p, a beginning-of-sequence symbol, and $d_{<t}$. A two-layer MLP readout g produces

$$c_t = g(h_t). \tag{3}$$

During training, h_t uses ground-truth preceding tokens under teacher forcing; at inference, it uses tokens sampled at earlier steps. After sampling d_t, only this discrete token enters the next language-model step: neither c_t nor generated semantic features are fed back in the standard pathway. The aligned sequence $c = (c_1, \ldots, c_T)$ excludes the state that predicts EOS. The readout c_t is a condition for the semantic generator, not the tokenizer output s_t or a direct regression estimate of it.

The semantic DiT embeds the discrete tokens with its own learned table and projects c into its feature space; both condition its velocity prediction from noisy features. The tokens provide explicit guidance, whereas c_t exposes the language model's continuous predictive state to the flow. Because c_t is determined by p and $d_{<t}$, it contains no independent information about a paired reference waveform beyond those inputs. Its contribution is a differentiable condition through which the semantic flow loss can update the planner. The separately learned target s defines the continuous features to be generated; it is not recovered by inverting the discrete quantizer.

Even with the same token prefix, different prompts can produce different readouts. The flow thus receives prompt-dependent planner states alongside the token plan. In the standard pathway, it processes the completed sequences bidirectionally after autoregressive planning. Continuous features do not enter the language model's history; the planner still sees sampled rather than reference token prefixes at inference.

The semantic DiT's token-embedding table is distinct from the tokenizer codebook C. In the standard configuration, we initialize its continuous-condition adapter to zero, leaving the pretrained token-conditioned flow's initial prediction unchanged. Training configurations are in Appendix B.

3.3 Joint Training

In the standard configuration, for each training pair (p, x), the fixed discrete tokenizer supplies reference tokens d, and the frozen continuous tokenizer supplies a sample $s \sim q_\phi(\cdot \mid x)$. As Figure 2 shows, the language model retains teacher-forced token cross-entropy $\mathcal{L}_{\mathrm{CE}}$, including sequence boundaries. For semantic flow matching, draw standard Gaussian noise $\epsilon \in \mathbb{R}^{T \times D}$ and time $\tau \sim \mathcal{U}[0, 1]$. The interpolated state $s_\tau = (1 - \tau)\epsilon + \tau s$ has target velocity $s - \epsilon$. Let $\widetilde{d}$ denote the DiT's token condition obtained from reference d. During training, independently selected positions

are replaced by a dedicated unconditional ID (gray entries in Figure 2); a separate example-level dropout replaces the entire sequence for selected examples. Neither operation changes the language model's reference-token prefix or its readout c. The semantic DiT predicts the velocity with v_θ, giving the flow loss for an unpadded sequence:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\frac{1}{TD}\left\|v_\theta(s_\tau, \tau; \widetilde{d}, c) - (s - \epsilon)\right\|_F^2\right]. \tag{4}$$

The expectation covers training pairs, posterior samples, noise, time, and token-condition corruption. We minimize $\mathcal{L}_{\mathrm{CE}} + \mathcal{L}_{\mathrm{FM}}$ with equal weights. The flow loss reaches the unfrozen language-model layers through the continuous readout and its adapter, while the token loss directly supervises autoregressive prediction; neither path requires a gradient through hard token assignments. Both token-condition perturbations are disabled at inference. Both tokenizers and the downstream acoustic renderer remain fixed. The padded-batch flow loss is given in Appendix B.

At initialization, the zero-initialized adapter preserves the pretrained token-conditioned flow. Its first update trains the adapter; subsequent flow gradients can reach the readout and unfrozen language-model layers. Token-condition masking is intended to reduce exclusive reliance on token identities. The present comparisons do not isolate its contribution.

3.4 Full-Song Generation

In the standard configuration, given a new prompt p, the language model samples discrete tokens until EOS while retaining their aligned readouts c. The semantic flow generates a complete continuous sequence from noise, conditioned on (d, c). A separately trained acoustic diffusion renderer converts this sequence into acoustic latents, which a decoder turns into the output waveform (Appendix C). The continuous tokenizer is used to provide training targets, not at inference.

In the standard pathway, planning proceeds at 25 semantic frames per second. Once tokens are complete, each semantic flow step updates the full feature sequence in parallel. The acoustic model then renders windows and decodes their latents. End-to-end latency depends on all three stages; it cannot be inferred from token rate alone.

4 Experiments

4.1 Experimental setup

Systems and training. Six complete song-generation systems are trained on in-house synthetic music data and share a LeVo-VAE waveform decoder (Lei et al., 2025). The standard joint pathway uses a Qwen2.5-1.5B language-model backbone (Qwen Team, 2024). Discrete sends VQ lookups to acoustic diffusion. Joint-STD-48D, Joint-STD-96D, and Joint-STD-128D pair token planning with full-sequence semantic flow and width-specific acoustic renderers. Joint-LeVo-64D generates acoustic latents directly by full-sequence flow, without a second acoustic DiT. Joint-SB-48D interleaves token planning and semantic-patch flow, feeding prior *semantic* patches back to the planner, unlike SongBloom's acoustic feedback (Yang et al., 2025). The LeVo and SB labels denote internal configurations. Joint training updates the planner, readout, and DiT flow while keeping the tokenizers and renderer fixed.

Evaluation and measures. Each system yields 192 retained songs on the same WildSongBench prompts (Multimodal Art Projection, 2026), for 1,152 outputs. The standard worker retries token sequences shorter than 30 seconds in fixed seed order, retaining the first sufficiently long result or the longest fallback; Joint-SB uses its own sampler. Shared prompts and scoring procedures permit system-level comparison. We report SongBench and SongEval means (Wu et al., 2026; Yao et al., 2025), AudioBox production quality (Tjandra et al., 2025), and MuQ–MuLan style similarity (Zhu et al., 2025). For subjective evaluation, five music experts rate ten matched song groups on sound quality and musicality using 1–5 scales; we report the mean ratings as MOS. Micro-PER pools phoneme edits after choosing the lowest-error of four Qwen3-ASR passes (Shi et al., 2026) on each saved song. MARBLE probes (Yuan et al., 2023) on reference-audio features are analyzed separately in Section 4.3.

4.2 SONG GENERATION RESULTS

Table 1 compares the six complete systems. Relative to Discrete, Joint-STD-48D improves SongBench (6.282 versus 6.217), SongEval (4.269 versus 4.152), and style alignment (0.514 versus 0.487), while reducing micro-PER from 20.260% to 14.470%. Its production-quality score is also slightly higher (8.300 versus 8.282), and the expert MOS scores favor it on both listening dimensions.

Table 1: Six systems on 192 WildSongBench prompts. PER is selected-transcript micro phoneme error (%). RTF is generation wall-clock time divided by audio duration. MOS denotes mean ratings by five music experts on ten matched song groups (sound quality and musicality; 1–5). Across all six systems, bold marks the best and underlining marks the second-best value in each column; tied second-best values are all underlined.

System	SongBench↑	$PQ_{AudioBox}$ ↑	SongEval↑	MuQ_{MuLan} ↑	PER↓	RTF↓	$MOS_{quality}$ ↑	MOS_{music} ↑
Discrete	6.217	8.282	4.152	0.487	20.26	**1.511**	3.14	2.76
Joint-STD-48D	6.282	8.300	4.269	0.514	**14.47**	1.879	**4.14**	**3.26**
Joint-STD-96D	6.263	**8.301**	4.222	0.503	17.25	1.779	3.38	3.22
Joint-STD-128D	6.251	8.276	4.145	**0.518**	18.86	1.682	3.76	3.14
Joint-LeVo-64D	6.207	8.281	4.230	0.507	18.02	1.728	3.04	3.10
Joint-SB-48D	**6.311**	8.282	**4.288**	0.495	18.21	2.434	3.38	2.88

RTF describes the observed cost of each complete generation pipeline under the current evaluation conditions.

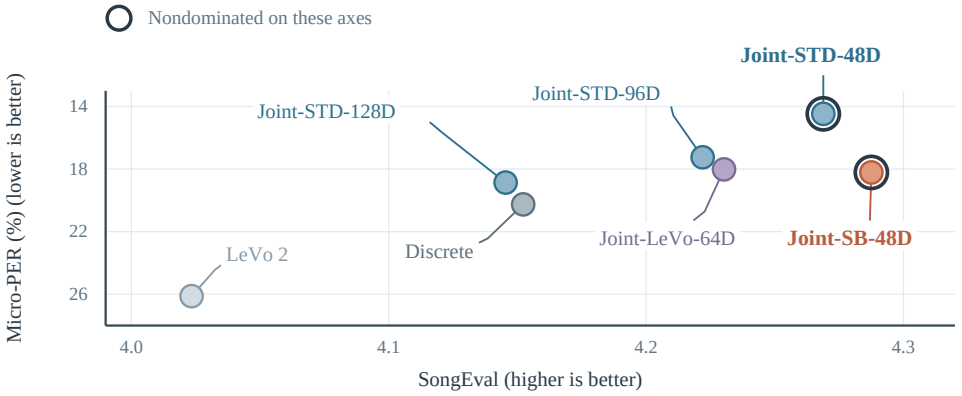


Figure 3: SongEval versus micro-PER for six evaluated systems and a published reference (Multimodal Art Projection, 2026). Outlined points are nondominated among the six systems. The external reference follows its source candidate-selection protocol.

Figure 3 shows the SongEval–PER trade-off: Joint-STD-48D and Joint-SB-48D are the two nondominated internal systems on the plotted axes. Within the standard pathway, increasing target width from 48D to 96D and 128D does not improve SongEval or PER; 96D marginally leads production quality, while 128D leads style alignment. Joint-LeVo-64D explores direct acoustic-latent prediction, and Joint-SB-48D explores semantic feedback.

4.3 DOWNSTREAM UTILITY OF SEMANTIC REPRESENTATIONS

Table 2 separates the original 1024D SSL reference from compact representations. Among the listed compact rows, 128D leads genre, arousal, valence, and structure; VQ leads beat, while continuous 48D leads downbeat. The 48D generator nevertheless has the lowest PER in Table 1. Relative to VQ lookups, its reference features improve five of the six probes, with beat F1 the exception.

Table 2: MARBLE probes on reference-audio features. Column labels give the dataset or task, with the metric as a subscript: EMO is EmoMusic and GTZAN-R is beat tracking on GTZAN Rhythm; A and V denote arousal and valence. The original SSL row is a separate reference; bold and underlining mark the best and second-best compact representations. VQ uses looked-up vectors, and continuous rows use unprojected bottleneck features.

Representation	$\text{GTZAN}_{\text{Acc.}}$	$\text{EMO}_{\text{A},R^2}$	$\text{EMO}_{\text{V},R^2}$	GTZAN-R_{F_1}	Downbeat_{F_1}	$\text{Structure}_{\text{Acc.}}$
Original SSL-1024D	0.703	0.648	0.439	0.657	0.449	0.553
VQ lookup	0.459	0.447	0.143	**0.328**	0.107	0.483
Continuous-48D	0.472	0.514	0.223	0.323	**0.110**	<u>0.514</u>
Continuous-96D	<u>0.510</u>	<u>0.541</u>	<u>0.250</u>	<u>0.325</u>	0.107	0.510
Continuous-128D	**0.541**	**0.557**	**0.297**	0.317	<u>0.108</u>	**0.520**

4.4 Reference-conditioned reconstruction

Table 3 reports reconstruction diagnostics on 100 in-house songs under two source-conditioned protocols. The original audio is scored by the same procedure as a reference.

Table 3: Reconstruction on 100 in-house songs (lower is better). Mel Mag and Chroma Mag are magnitude errors; CTC is the transcription loss and CER is character error rate. The original audio is a scoring reference. Bold and underlining mark the best and second-best system within each reconstruction protocol.

System	Mel Mag↓	Chroma Mag↓	CTC↓	CER↓
Original audio	0.000	0.000	1.033	18.52%
Ground-truth acoustic-latent reconstruction				
Discrete	0.392	0.198	2.007	41.41%
Joint-STD-48D	<u>0.273</u>	<u>0.096</u>	<u>1.331</u>	<u>25.93%</u>
Joint-STD-96D	0.287	0.113	1.454	30.30%
Joint-STD-128D	0.296	0.126	1.465	28.62%
Joint-LeVo-64D	**0.256**	**0.078**	1.357	26.94%
Joint-SB-48D	0.275	0.098	**1.223**	**22.90%**
Teacher-forced reconstruction				
Discrete	0.450	0.248	1.746	31.31%
Joint-STD-48D	<u>0.440</u>	<u>0.247</u>	<u>1.586</u>	<u>30.64%</u>
Joint-STD-96D	0.488	0.300	2.008	42.76%
Joint-STD-128D	0.486	0.275	1.811	36.36%
Joint-LeVo-64D	0.495	0.284	2.094	40.07%
Joint-SB-48D	**0.362**	**0.205**	**1.348**	**28.28%**

Under ground-truth acoustic latents, Joint-LeVo-64D has the lowest Mel and Chroma magnitude errors among the systems, while Joint-SB-48D has the lowest CTC and CER. Joint-SB-48D leads all four reported measures under teacher forcing.

5 Discussion and Limitations

System-level gains and their attribution. The joint pathway retains a low-rate discrete plan while exposing the language model's predictive states to semantic synthesis through a differentiable readout. Relative to Discrete, Joint-STD-48D reduces micro-PER by 5.79 percentage points (20.26% to 14.47%), raises SongEval from 4.152 to 4.269, and raises the reported sound-quality MOS from 3.14 to 4.14 (Table 1). AudioBox PQ changes by only 0.018. These measures emphasize different properties of the same songs; a single quality score would obscure the concurrent changes in lyric fidelity and listening judgments. The comparison supports the viability of the complete joint system, but cannot assign its gains specifically to the flow gradient, the continuous target, or the renderer, all of which differ from the discrete baseline. The readout also cannot recover details lost from a particular reference recording: it is determined by the prompt and token history.

Semantic capacity and generation behavior. Larger reference-audio features do not yield uniformly stronger generation. From 48D to 128D, genre accuracy rises from 0.472 to 0.541 and valence R^2 from 0.223 to 0.297 in Table 2; across the corresponding complete systems, SongEval falls from 4.269 to 4.145 and micro-PER rises from 14.47% to 18.86% in Table 1. A richer feature space may be harder to predict or render, but these comparisons cannot establish that mechanism because the training paths and renderers are not fully matched. Figure 3 shows a separate trade-off in the generation schedule: Joint-SB-48D has the highest SongBench and SongEval point estimates (6.311 and 4.288), whereas Joint-STD-48D has the lowest micro-PER (14.47% versus 18.21%). Semantic feedback is therefore an informative design alternative, although these complete-system results do not isolate its effect.

What reconstruction can diagnose. The 100-song source-conditioned study in Table 3 probes a different failure mode from free generation. With ground-truth acoustic latents, Joint-LeVo-64D has the lowest Mel and Chroma magnitude errors (0.256 and 0.078), yet its free-generation SongBench score is 6.207, below Joint-STD-48D's 6.282. Under teacher forcing, Joint-SB-48D leads all four reconstruction measures, while its free-generation lyric error remains higher than Joint-STD-48D's. Access to reference conditions can expose what a renderer preserves, but cannot show how reliably the planner supplies those conditions. Even the original audio has nonzero transcription loss and CER in this diagnostic, so transcription-derived measures should be read with their measurement process in mind.

Evaluation and remaining controls. The automatic results are point estimates from one retained song per system and prompt; the five-expert MOS study covers ten matched groups rather than all 192 prompts. The 0.001 AudioBox PQ difference between Joint-STD-96D and Joint-STD-48D, for example, is too small to interpret without paired uncertainty. Micro-PER selects the lowest-error result of four Qwen3-ASR passes, and the standard worker's duration retries differ from Joint-SB's sampling protocol. These choices limit direct comparison with other candidate-selection rules and do not remove recognition or text-normalization error. Matched variants that block the flow gradient, remove the readout, or directly reconstruct semantic features from language-model states would identify the contribution of each connection under the same target, renderer, and training budget.

Inference mismatch and cost. The standard pathway keeps generated continuous features out of the language model's autoregressive history, avoiding semantic-feature feedback mismatch. It still learns from reference token prefixes and generates from sampled ones; the acoustic renderer likewise encounters predicted rather than reference semantic features at inference. Interleaved semantic feedback adds another train–inference mismatch to examine. Full-sequence semantic flow also waits for the completed token plan, and the additional representation and acoustic models incur training and inference cost. The present experiments establish neither a compute advantage over direct acoustic generation nor an optimal width across datasets, languages, and budgets.

6 Conclusion

Joint discrete and continuous modeling extends token-guided song generation with a continuous condition and joint language-model and semantic-flow training. Discrete tokens remain the autoregressive history and guide synthesis, while semantic generation uses the language model's predictive states through a learned readout. On a common 192-prompt evaluation, Joint-STD-48D improves the main automatic metrics over the discrete baseline; macro-level bottleneck and architecture ablations show different quality and style-alignment outcomes. Representation probes further show that no compact target width leads every measured task. Finer component ablations remain a direction for future work.

AI USE STATEMENT

Generative AI assisted with polishing. The authors are responsible for verifying the AI-assisted text and claims and for the final content of the paper.

ETHICS STATEMENT

Song synthesis can reproduce recognizable vocal characteristics and musical material. Generated audio and automatic assessments should not be presented as recordings or judgments from real performers or listeners.

REPRODUCIBILITY STATEMENT

The main text specifies the representation paths, joint-training objective, and generation systems; the appendix details the representation targets, semantic DiT pretraining, joint semantic training, and acoustic rendering. The code and demo address is given below the abstract.

REFERENCES

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. MusicLM: Generating Music From Text, 2023. URL https://arxiv.org/abs/2301.11325.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: a Language Modeling Approach to Audio Generation, 2022. URL https://arxiv.org/abs/2209.03143.

Huakang Chen, Guobin Ma, Yuepeng Jiang, Dake Guo, Jingbin Hu, Hanke Xie, Wenhao Li, Lingxin Xiong, Jian Zhao, Zhonglin Jiang, Yong Chen, Lei Xie, and Pengcheng Zhu. Rethinking music tokenization: A semantic codec toward high-fidelity LLM music generation, 2026. URL https://arxiv.org/abs/2609.21240.

Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised Learning with Random-projection Quantizer for Speech Recognition, 2022. URL https://arxiv.org/abs/2202.01855.

Ju-Chieh Chou, Jiawei Zhou, and Karen Livescu. Flow-SLM: Joint learning of linguistic and acoustic information for spoken language modeling, 2025. URL https://arxiv.org/abs/2508.09350.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and Controllable Music Generation, 2023. URL https://arxiv.org/abs/2306.05284.

Junyu Dai, Xinyue Fan, Weiqin Li, Xiangang Li, Yunjia Li, Bin Ma, Yukun Ma, Chongjia Ni, Yufei Shi, Biao Tian, Haoxu Wang, Menglin Wu, Jianwei Yu, Huaicheng Zhang, Han Zhao, Shengkui Zhao, and Haina Zhu. Pushing the Frontier of Full-Song Generation: Hierarchical Autoregressive Planning Meets Flow-Matching Rendering, 2026. URL https://arxiv.org/abs/2607.20253.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A Generative Model for Music, 2020. URL https://arxiv.org/abs/2005.00341.

Chengli Feng, Zhiyue Wu, Jiahao Song, Zheqi Dai, Boyang Wang, Ruibin Yuan, Junming Gong, Wenxiao Zhao, Jing Guo, Gang Yu, Xiangyu Zhang, Xuerui Yang, and Chao Yan. StepAudio 3 Music Technical Report, 2026. URL https://arxiv.org/abs/2609.16034.

Junmin Gong, Sean Zhao, Sen Wang, Shengyuan Xu, and Joe Guo. ACE-Step: A Step Towards Music Generation Foundation Model, 2025. URL https://arxiv.org/abs/2506.00045.

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, 2025. URL https://arxiv.org/abs/2510.22950.

Shun Lei, Yaoxun Xu, Zhiwei Lin, Huaicheng Zhang, Wei Tan, Hangting Chen, Jianwei Yu, Yixuan Zhang, Chenyu Yang, Haina Zhu, Shuai Wang, Zhiyong Wu, and Dong Yu. LeVo: High-Quality Song Generation with Multi-Preference Alignment, 2025. URL https://arxiv.org/abs/2506.07520.

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, 2026. URL https://arxiv.org/abs/2606.30642.

Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. AudioLDM 2: Learning Holistic Audio Generation with Self-supervised Pretraining, 2023. URL https://arxiv.org/abs/2308.05734.

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, 2025. URL https://arxiv.org/abs/2502.13128.

Multimodal Art Projection. WildSongBench: Results and evaluation scope, 2026. URL https://huggingface.co/datasets/m-a-p/WildSongBench/. Benchmark results updated September 12, 2026.

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, 2025. URL https://arxiv.org/abs/2503.01183.

K R Prajwal, Bowen Shi, Matthew Le, Apoorv Vyas, Andros Tjandra, Mahi Luthra, Baishan Guo, Huiyu Wang, Triantafyllos Afouras, David Kant, and Wei-Ning Hsu. MusicFlow: Cascaded Flow Matching for Text Guided Music Generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41052–41063, 2024. URL https://proceedings.mlr.press/v235/prajwal24a.html.

Qwen Team. Qwen2.5 Technical Report, 2024. URL https://arxiv.org/abs/2412.15115.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, Jin Xu, Jingren Zhou, and Junyang Lin. Qwen3-ASR Technical Report, 2026. URL https://arxiv.org/abs/2601.21337.

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. URL https://arxiv.org/abs/2502.05139.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

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, 2026. URL https://arxiv.org/abs/2604.25937.

Jin Xu, Kangdi Wang, Ruibin Yuan, Shun Lei, Xiong Wang, Xize Cheng, Xueyao Zhang, Yang Zhang, Yiheng Chen, Yongqi Wang, Yue Wang, Zhifang Guo, Zihan Liu, Zijian Lin, Dake Guo, Hangrui Hu, Lei Xie, Linhan Ma, Wei Xue, Wenxiang Guo, Xinfa Zhu, Xipin Wei, Yangze Li, Yuanjun Lv, Yuxuan Wang, Yunfei Chu, and Zhiyong Wu. Qwen-Music Technical Report, 2026. URL https://arxiv.org/abs/2607.11699.

Yaoxun Xu, Hangting Chen, Jianwei Yu, Wei Tan, Rongzhi Gu, Shun Lei, Zhiwei Lin, and Zhiyong Wu. MuCodec: Ultra Low-Bitrate Music Codec, 2024. URL https://arxiv.org/abs/2409.13216.

Chenyu Yang, Shuai Wang, Hangting Chen, Wei Tan, Jianwei Yu, and Haizhou Li. SongBloom: Coherent Song Generation via Interleaved Autoregressive Sketching and Diffusion Refinement, 2025. URL https://arxiv.org/abs/2506.07634.

Dongchao Yang, Yuxin Xie, Yuguo Yin, Zheyu Wang, Xiaoyu Yi, Gongxi Zhu, Xiaolong Weng, Zihan Xiong, Yingzhe Ma, Dading Cong, Jingliang Liu, Zihang Huang, Jinghan Ru, Rongjie Huang, Haoran Wan, Peixu Wang, Kuoxi Yu, Helin Wang, Liming Liang, Xianwei Zhuang, Yuanyuan Wang, Dingdong Wang, Haohan Guo, Junjie Cao, Zeqian Ju, Songxiang Liu, Yuewen Cao, Heming Weng, and Yuexian Zou. HeartMuLa: A Family of Open Sourced Music Foundation Models, 2026. URL https://arxiv.org/abs/2601.10547.

Jixun Yao, Guobin Ma, Huixin Xue, Huakang Chen, Chunbo Hao, Yuepeng Jiang, Haohe Liu, Ruibin Yuan, Jin Xu, Wei Xue, Hao Liu, and Lei Xie. SongEval: A benchmark dataset for song aesthetics evaluation. *arXiv preprint arXiv:2505.10793*, 2025. doi: 10.48550/arXiv.2505.10793. URL https://arxiv.org/abs/2505.10793.

Ruibin Yuan, Yinghao Ma, Yizhi Li, Ge Zhang, Xingran Chen, Hanzhi Yin, Le Zhuo, Yiqi Liu, Jiawen Huang, Zeyue Tian, Binyue Deng, Ningzhi Wang, Chenghua Lin, Emmanouil Benetos, Anton Ragni, Norbert Gyenge, Roger Dannenberg, Wenhu Chen, Gus Xia, Wei Xue, Si Liu, Shi Wang, Ruibo Liu, Yike Guo, and Jie Fu. MARBLE: Music Audio Representation Benchmark for Universal Evaluation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/7cbeec46f979618beafb4f46d8f39f36-Abstract-Datasets_and_Benchmarks.html.

Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, Xinrun Du, Zhen Ye, Tianyu Zheng, Zhengxuan Jiang, Yinghao Ma, Minghao Liu, Zeyue Tian, Ziya Zhou, Liumeng Xue, Xingwei Qu, Yizhi Li, Shangda Wu, Tianhao Shen, Ziyang Ma, Jun Zhan, Chunhui Wang, Yatian Wang, Xiaowei Chi, Xinyue Zhang, Zhenzhu Yang, Xiangzhou Wang, Shansong Liu, Lingrui Mei, Peng Li, Junjie Wang, Jianwei Yu, Guojian Pang, Xu Li, Zihao Wang, Xiaohuan Zhou, Lijun Yu, Emmanouil Benetos, Yong Chen, Chenghua Lin, Xie Chen, Gus Xia, Zhaoxiang Zhang, Chao Zhang, Wenhu Chen, Xinyu Zhou, Xipeng Qiu, Roger Dannenberg, Jiaheng Liu, Jian Yang, Wenhao Huang, Wei Xue, Xu Tan, and Yike Guo. YuE: Scaling Open Foundation Models for Long-Form Music Generation, 2025. URL https://arxiv.org/abs/2503.08638.

Peng Zheng, Junke Wang, Yi Chang, Yizhou Yu, Rui Ma, and Zuxuan Wu. Rethinking Discrete Tokens: Treating Them as Conditions for Continuous Autoregressive Image Synthesis, 2025. URL https://arxiv.org/abs/2507.01756.

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. *arXiv preprint arXiv:2501.01108*, 2025. doi: 10.48550/arXiv.2501.01108. URL https://arxiv.org/abs/2501.01108.

A Semantic Representation Targets

We use three separately trained representation paths: the original 1024D SSL encoder for MARBLE probing, a single-codebook VQ tokenizer for discrete semantic tokens, and an SSL-VAE for continuous semantic targets. All representations use 24-kHz audio and operate at 25 Hz. The discrete tokenizer adopts a multi-stage representation-learning and tokenization paradigm similar to prior music tokenization systems (Feng et al., 2026; Xu et al., 2026), consisting of BEST-RQ-style self-supervised pretraining, CTC/Mel/Chroma supervision, and single-codebook vector quantization. The SSL-VAE is initialized from a separate Stage-2 checkpoint and introduces a Gaussian bottleneck for continuous semantic modeling. The three representation paths use separate weights.

For each frame, the SSL-VAE defines a diagonal Gaussian posterior and samples the continuous semantic target through reparameterization:

$$q(s_t|x) = \mathcal{N}(\mu_t, \mathrm{diag}(e^{\nu_t}))\,, \qquad s_t = \mu_t + \exp(\nu_t/2) \odot \xi_t, \quad \xi_t \sim \mathcal{N}(0, I). \tag{5}$$

Posterior samples s_t are used as semantic-flow training targets, whereas posterior means μ_t are used for deterministic representation probing.

The SSL-VAE is optimized with

$$\mathcal{L}_{\mathrm{rep}} = 2\mathcal{L}_{\mathrm{CTC}} + \mathcal{L}_{\mathrm{mel}} + \mathcal{L}_{\mathrm{chr}} + 10^{-5}\mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{align}}. \tag{6}$$

For $r \in \{\mathrm{mel}, \mathrm{chr}\}$, the spectral reconstruction loss is the sum of spectral convergence and magnitude L_1 losses:

$$\mathcal{L}_r = \frac{\|Y_r - \hat{Y}_r\|_F}{\|Y_r\|_F} + \frac{1}{N_r}\|Y_r - \hat{Y}_r\|_1, \tag{7}$$

where Y_r and $\hat{Y}_r$ denote the reference and predicted Mel or Chroma features, respectively, and N_r is the number of feature elements. Key representation configurations are summarized in Table A.1.

Setting	Discrete Tokenizer	Continuous SSL-VAE
Bottleneck Position	After block 12	After block 12
Bottleneck Type	Single-codebook VQ	Diagonal Gaussian
Codebook Size	32,768	–
Training Objectives	CTC + Mel + Chroma	CTC + Mel + Chroma + KL + Align
Training Target	Discrete token d_t	Posterior sample s_t
Evaluation Feature	Codebook vector $C[d_t]$	Posterior mean μ_t

Table A.1: **Semantic representation configuration**. Both representation paths use 24-kHz audio, a 24-block non-causal Conformer, and temporally aligned 25-Hz features.

B Joint Semantic Generator and Training

Discrete LM pretraining. The discrete song planner is initialized from Qwen2.5-1.5B (Qwen Team, 2024) and trained as a text-conditioned autoregressive music-token model. Each training sequence consists of a formatted lyric-and-style text prefix followed by audio BOS, 25-Hz semantic tokens, and EOS. The complete text prefix is visible to all music-token positions but is excluded from the training loss; next-token cross-entropy is applied only to the audio-token sequence, including the audio boundaries. The learning rate is linearly warmed up for 2,000 steps to 3×10^{-4}, kept constant for most of training, and linearly annealed to zero over the final 10% of training steps.

Standalone semantic DiT pretraining. Before joint training, the semantic DiT is trained to generate SSL-VAE features from noise using reference semantic IDs alone. This ID-conditioned model initializes the semantic DiT in the joint pathway; the language-model readout is added only during joint training. Table B.1 summarizes the standalone configuration.

Setting	Semantic DiT Warm-up
Architecture	1D DiT-XL; 28 blocks, width 1152, 16 heads
Conditioning	Ground-truth 25-Hz semantic IDs; no LM hidden-state readout
Prediction Target	SSL-VAE posterior features, $D \in \{48, 96, 128\}$
Objective	Flow-matching velocity MSE, $\|v_\theta(s_\tau, \tau; d) - (s - \epsilon)\|_F^2/(TD)$
Flow Path	$s_\tau = (1-\tau)\epsilon + \tau s$, with $\epsilon \sim \mathcal{N}(0, I)$
Sequence	Frame-aligned full song at 25 Hz; maximum 8192 frames
Trainable	Semantic DiT only
Frozen	Semantic tokenizer and SSL-VAE target encoder; LM not used
LR Warmup	5,000 steps
Peak LR	5×10^{-5}
LR Schedule	Warmup followed by cosine decay to 10^{-7} at step 500,000
Sampling	32 Euler flow steps

Table B.1: **Standalone semantic DiT pretraining.** The model learns SSL-VAE features conditioned on reference semantic IDs before joint semantic training.

Joint semantic training. The standard joint pathway uses the pretrained song LM with a separate audio-token embedding and prediction head. A two-layer readout maps the language model's predictive states to continuous conditions c_t for a semantic DiT. The semantic DiT predicts continuous SSL-VAE features from noise, conditioned jointly on discrete token embeddings and the continuous readouts. The continuous tokenizer and downstream acoustic renderer remain fixed during joint training.

The readout c_t is aligned with the token predicted by the corresponding language-model state: the BOS state predicts the first semantic frame, and the state after d_{T-1} predicts the final frame. The state predicting EOS has no aligned semantic target. Training uses reference-token prefixes, whereas inference uses sampled prefixes.

The joint objective is

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{FM}}, \tag{8}$$

where token cross-entropy supervises autoregressive semantic-token prediction and the flow-matching loss trains the semantic DiT. For a padded minibatch, the implemented flow loss is

$$\ell_{\text{FM}} = \frac{\sum_{i,t} m_{it} \|v_{it} - (s_{it} - \epsilon_{it})\|_2^2 / D}{\sum_{i,t} m_{it}}, \tag{9}$$

where m_{it} marks valid semantic frames, s is sampled from the frozen SSL-VAE posterior, and ϵ is the flow noise. Token corruption is applied only to the semantic DiT condition and does not modify the teacher-forced language-model history. The continuous readout is not detached, allowing flow-matching gradients to update the language model without differentiating through discrete token selection.

Key configurations are summarized in Table B.2.

Setting	Discrete LM Pretraining	Joint Semantic Training
Base Model	Qwen2.5-1.5B	Pretrained Music LM + Semantic DiT
Conditioning	Lyrics + style	IDs + LM readout
Prediction Target	25-Hz semantic IDs	SSL-VAE features
Objective	Audio-token NTP	$\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{FM}}$
Sequence	Text + BOS + IDs + EOS	Frame-aligned conditions
Trainable	LM	LM + readout + DiT
Frozen	Semantic tokenizer	Tokenizers + acoustic renderer
LR Warmup	2,000 steps	None
Peak LR	3×10^{-4}	DiT + latent head: 5×10^{-5}; LM: 1×10^{-5}
LR Schedule	Constant; final 10% decay to 0	Constant; no scheduler

Table B.2: **Discrete planner pretraining and joint semantic training configuration**. The discrete LM is first trained with text-conditioned next-token prediction. Joint training subsequently optimizes the pretrained planner, continuous readout, and semantic DiT while keeping the representation encoders and acoustic renderer fixed.

C ACOUSTIC RENDERING

For the standard SSL-VAE pathway, a separately trained acoustic DiT maps semantic features to LeVo VAE acoustic latents. It receives reference SSL-VAE features during its own training and generated semantic features at song generation time. The acoustic renderer is frozen during joint semantic training. Table C.1 reports the selected width-specific renderers.

Setting	**Selected Acoustic DiT Renderers**
Architecture	Continuous DiT; 24 blocks, width 1536, 24 heads
Conditioning	Reference SSL-VAE features ($D = 48, 96, 128$); preceding 50 acoustic latent frames as context; 128D uses condition LayerNorm
Prediction Target	64-channel LeVo VAE acoustic latents
Objective	Trigonometric diffusion v-prediction
Sequence	25-Hz, 750-frame (30-s) windows with 50-frame (2-s) context
Trainable	Acoustic DiT; trained separately and frozen during joint semantic training
Frozen	SSL-VAE feature extractor and pretrained LeVo VAE decoder
LR Warmup	5,000 steps
Peak LR	5×10^{-5}
LR Schedule	Warmup followed by cosine decay to 10^{-7} at step 500,000
Sampling	32 deterministic v-prediction updates, then 48-kHz stereo LeVo decoding

Table C.1: **Selected acoustic DiT renderers for the standard SSL-VAE route.** Each representation width uses a separate acoustic checkpoint. †The 96D/128D learning-rate values come from the selected checkpoint directory names; the original trainer configuration was not recovered.