SKETCHSONG: HIERARCHICAL SONG GENERATION WITH SKETCH PLANNING AND FINE-GRAINED MULTI-TRACK MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent song generation systems can synthesize realistic audio, yet generating complete songs remains challenging for two reasons. First, explicit song-level arrangement planning remains limited in existing methods, so models often need to organize overall arrangement development while generating low-level audio details. This often leads to incoherence in arrangements, such as weak section transitions and limited dynamic progression. Second, coarse modeling of different musical parts obscures their distinct roles and interactions, limiting arrangement richness of generated songs. In this paper, we present SketchSong, a hierarchical song generation framework that addresses these issues through song-level sketch planning and fine-grained multi-track modeling. Along the temporal dimension, SketchSong first predicts a compact sequence of high-level sketch tokens derived from compressed audio representations, and then generates audio tokens conditioned on these sketches. This coarse-to-fine process gives the model an explicit arrangement plan before detailed audio generation. Along the track dimension, SketchSong explicitly models four tracks, i.e., vocals, bass, drums and other instruments. This enables the model to capture the roles and interactions of different musical parts more precisely. Experiments on song generation benchmarks show that SketchSong consistently outperforms our baseline on both objective metrics and human listening tests.

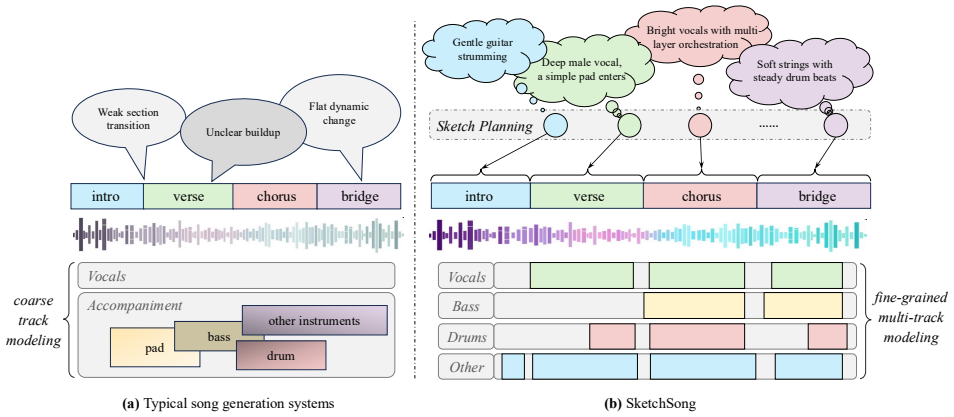
1 INTRODUCTION

Recent advances in generative audio models have substantially improved conditional song synthesis from structured inputs such as lyrics, text descriptions, and reference audio Lei et al. (2024); Ning et al. (2025); Lei et al. (2025); Yang et al. (2025). However, generating a convincing complete song requires more than locally realistic sound. A good song needs clear section development, natural changes in energy, and coordinated interactions among vocals and instrumental parts. These qualities should be maintained throughout the song, rather than only within short local spans. Despite encouraging progress in recent song generation systems, achieving both global coherence and rich arrangement remains challenging.

One central challenge is limited song-level arrangement planning. Prior work has explored hierarchical semantic generation Agostinelli et al. (2023); Prajwal et al. (2024), CoT- or sketch-like intermediate guidance Lam et al. (2025); Yang et al. (2025), and segment-level prompting for structured control Cai et al. (2025). These directions suggest that intermediate abstractions can help, yet explicit planning for arrangement development in complete songs remains limited. As a result, models often need to handle both high-level arrangement development (*e.g.*, section progression and dynamic contour) and low-level audio details within the same generation process. This can undermine coherence over the full song, leading to weak section transitions, limited dynamic progression, and less purposeful changes in instrumentation.

A second challenge lies in how different musical parts are modeled. Existing systems often generate songs as a single mixed signal Dhariwal et al. (2020); Ning et al. (2025), or adopt only coarse vocal–accompaniment decompositions Lei et al. (2024; 2025). Recent multi-track and stem-aware systems Yao et al. (2025b); Parker et al. (2024); Zhang et al. (2025) further suggest the value of

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.



(a) Typical song generation systems **(b)** SketchSong

Figure 1: Limitations of existing song generation systems versus SketchSong. (a) Existing systems typically exhibit limited song-level planning and entangled instrumental roles inside accompaniment. (b) SketchSong introduces song-level sketch planning and fine-grained multi-track modeling, leading to clearer arrangement development and richer musical instrumentation.

explicit part modeling. In practice, different musical parts (*e.g.*, vocals, bass, drums, and other instruments) do not contribute in the same way. Bass and drums strongly shape groove, rhythmic drive, and stylistic feel, while vocals and other instruments carry melody, harmony, and sonic texture. Therefore, better song generation benefits from finer part-aware modeling, so that the roles and interactions of different musical parts can be represented more precisely and used to support richer, better-coordinated arrangements.

To address these issues, we present SketchSong, a hierarchical song generation framework with song-level sketch planning and fine-grained multi-track modeling. For song-level planning, we compress high-level music representations into a low-rate sequence of discrete sketch tokens, and place them between the conditioning inputs and the audio token sequence. In this way, the model first predicts a sketch with high-level semantics before generating detailed audio, providing an explicit plan for arrangement development. We adopt a two-phase training strategy that first learns sketch prediction and then optimizes the full sketch-conditioned mixed-audio generation process. For track-aware modeling, we explicitly represent four tracks (*i.e.*, vocals, bass, drums, and other instruments), enabling more precise modeling of musical roles and cross-part interactions. These two components are complementary: sketch planning improves how a song develops over time, while multi-track modeling improves how musical parts are organized within the arrangement.

We instantiate SketchSong in a two-stage song generation pipeline, following the strong mixed-to-track refinement paradigm used in a recent hierarchical song generation system Lei et al. (2025). The first stage generates mixed audio tokens guided by song-level sketches, and the second stage refines the mixed audio tokens into finer-grained four-track tokens. We evaluate SketchSong on song generation benchmarks using both objective metrics and human listening tests. Experiments show that SketchSong consistently improves over our baseline and remains competitive with strong open-source systems, even without additional post-training for preference optimization or alignment. More importantly, the empirical results align well with our design motivations: song-level sketch planning mainly improves long-range arrangement development, section clarity, and perceived musical progression, while fine-grained multi-track modeling mainly enhances instrumental richness and production-related quality. Together, these results suggest that the two components are not redundant, but complementary in how they improve complete-song generation. In summary, the main contributions of this work are as follows.

- We introduce song-level sketch planning for song generation by inserting compact discrete sketch tokens before audio generation, enabling a coarse-to-fine process that provides explicit guidance for arrangement development.
- We propose fine-grained multi-track modeling with four explicit tracks–vocals, bass, drums, and other instruments–to better capture the roles and interactions of different musical parts.

- We integrate these approaches into a hierarchical song generation framework. Experiments show that SketchSong consistently improves over our baseline on both objective metrics and human listening tests, while ablations further indicate that sketch planning benefits structure-related qualities and multi-track modeling benefits arrangement richness and production quality.

2 Related Work

Song Generation. Early song generation systems either modeled songs directly in raw or mixed audio, or relied on relatively coarse vocal–accompaniment decomposition. Jukebox Dhariwal et al. (2020) demonstrated hierarchical raw-audio song generation with VQ-VAE compression and autoregressive Transformers. More recent systems, including SongCreator Lei et al. (2024), LeVo Lei et al. (2025), DiffRhythm Ning et al. (2025), DiffRhythm 2 Jiang et al. (2025), SongBloom Yang et al. (2025), Muse Jiang et al. (2026), and ACE-Step Gong et al. (2026), have substantially improved full-song generation quality and controllability. Seed-Music Bai et al. (2024) presents a suite of hybrid autoregressive and diffusion systems supporting vocal music generation and interactive editing from multi-modal conditions. YuE Yuan et al. (2025) scales open foundation models for long-form music generation with large language model backbones. However, these systems generally do not combine explicit song-level planning with finer-grained track refinement in the way targeted by SketchSong.

Hierarchical Planning and Structured Control. Intermediate representations and structured controls have become an important direction in music generation. MusicLM Agostinelli et al. (2023) uses semantic-to-acoustic generation with MuLan-based conditioning. MusicFlow Prajwal et al. (2024) adopts a cascaded flow-matching design that first generates semantic features from text and then acoustic features from semantics, reinforcing the now-common pattern of splitting generation into a semantic stage and an acoustic stage. MusiCoT Lam et al. (2025) introduces CoT-style latent musical thoughts, where CLAP-derived latent representations serve as an analyzable intermediate planning sequence ahead of audio-token generation. SongBloom Yang et al. (2025) applies sketch-based intermediate generation to full songs, and SegTune Cai et al. (2025) enables segment-level structured control. On the controllability side, JASCO Tal et al. (2024) combines global text control with local, time-aligned conditioning from symbolic or audio-derived signals such as chords, melody, and drum tracks. MusiConGen Lan et al. (2024) extends MusicGen Copet et al. (2023) with temporally aligned rhythm and chord control, while Mustango Melechovsky et al. (2024) enriches textual prompts with music-theoretic attributes such as beats, tempo, key, and chord progression. These works show the value of intermediate abstractions for long-range coherence and controllability, but they either target general music generation, rely on externally specified controls, or do not connect song-level planning to later fine-grained track modeling. In contrast, SketchSong predicts compact song-level sketch tokens and uses them as an internal planning stage before detailed song synthesis.

Part-Aware and Multi-Track Modeling. Explicit modeling of musical parts is important for arrangement quality and controllability. SongCreator Lei et al. (2024) and LeVo Lei et al. (2025) separate vocals and accompaniment, with LeVo further coupling mixed-token and dual-track modeling in a two-stage framework built on a specialized music codec Xu et al. (2025b). Beyond dual-track formulations, JEN-1 Composer Yao et al. (2025b) models conditional generation over multiple tracks for human–AI co-composition, StemGen Parker et al. (2024) generates missing stems conditioned on an existing context mix, and VersBand Zhang et al. (2025) decomposes song creation into dedicated vocal and accompaniment modules. These works highlight the benefits of part-aware generation, but fine-grained track organization remains relatively limited in end-to-end song generation. SketchSong extends this direction by refining songs into four explicit tracks—vocals, bass, drums, and other instruments—while coupling this finer-grained track modeling with song-level sketch planning.

3 Method

We present SketchSong, a hierarchical song generation framework for improving arrangement planning and track-aware refinement in full-song generation. SketchSong is built on the strong two-stage

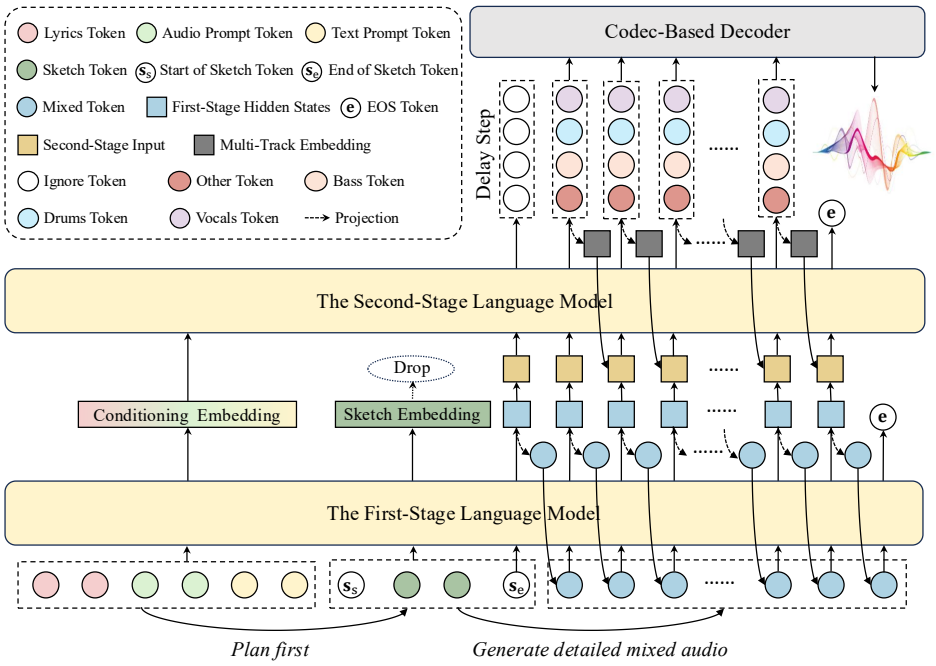


Figure 2: Overview of SketchSong. Given conditioning inputs, the first-stage language model predicts song-level sketch tokens before generating detailed mixed-song audio tokens. The second-stage language model then refines the mixed-song representation into four explicit tracks, *i.e.*, vocals, bass, drums, and other instruments, with additional conditioning from first-stage hidden states which provide sketch-informed arrangement cues.

autoregressive generation paradigm of LeVo Lei et al. (2025), where a first-stage language model generates an intermediate mixed-song representation, and a second-stage language model further refines it into separate vocal and accompaniment tracks. Within this general framework, SketchSong introduces two orthogonal methodological designs. First, we incorporate song-level sketch planning into the first generation stage. Second, we introduce a fine-grained four-track refinement stage over vocals, bass, drums, and other instruments. The overall framework is illustrated in Fig. 2.

3.1 Overview

Given structured lyrics, an optional text description, and an optional audio prompt, SketchSong generates a song in a hierarchical coarse-to-fine manner with two autoregressive language models, parameterized by θ_1 and θ_2, respectively. In the first stage, the model f_{θ_1} performs song-level planning and mixed-audio generation. Rather than directly predicting mixed audio tokens from the conditioning inputs, it first predicts a compact sequence of discrete sketch tokens and then generates mixed audio tokens conditioned on these sketches:

$$\mathbf{s} = f_{\theta_1}(\mathbf{c}), \qquad \mathbf{x}_{\text{mix}} = f_{\theta_1}(\mathbf{c}, \mathbf{s}), \tag{1}$$

where $\mathbf{c}$ denotes the condition tensors, $\mathbf{s}$ denotes the song-level sketch tokens, and $\mathbf{x}_{\text{mix}}$ denotes the mixed-song audio tokens. The sketch tokens are constructed offline by extracting MuQ-MuLan Huang et al. (2022) audio features from 5-second non-overlapping windows and mapping these features to discrete code indices with residual vector quantization. During inference, the sketch tokens can either be generated by the model itself or provided by the user, enabling controllable generation.

In the second stage, the model g_{θ_2} refines the intermediate mixed representation into four explicit tracks. Specifically, it takes the same conditioning inputs together with the mixed-song representation from the first stage, and predicts four token streams corresponding to vocals, bass, drums, and other instruments:

$$\mathbf{x}_{\text{trk}} = g_{\theta_2}(\mathbf{c}, \mathbf{x}_{\text{mix}}), \tag{2}$$

where $\mathbf{x}_{\mathrm{trk}}$ denotes the four-track tokens. This design enables finer part-aware generation and allows the model to represent the roles and interactions of different musical parts more explicitly.

After token prediction, we reuse the codec-based decoding pipeline of LeVo Lei et al. (2025) to reconstruct waveforms. Since dedicated codecs for bass, drums, and other tracks are not separately trained in the current system, these three tracks share the accompaniment codec during decoding. Our two designs are complementary: song-level sketch planning improves how the song develops over time, while four-track refinement improves how musical parts are organized within the arrangement.

3.2 Song-Level Sketch Planning

Direct autoregressive song generation requires the first-stage model to organize high-level arrangement development (*e.g.*, section progression, dynamic contour, and layer evolution) while also predicting dense audio tokens within the same generation process, which makes long-range arrangement control difficult. To provide explicit planning guidance before detailed audio synthesis, we introduce song-level sketch planning, a low-rate discrete intermediate representation inserted between the conditioning inputs and the mixed-audio token sequence in the first-stage model.

Song-Level Sketch Tokens. We construct song-level sketch tokens offline from MuQ-MuLan Huang et al. (2022) audio features. Specifically, for each training song, we extract a sequence of MuQ-MuLan feature vectors from non-overlapping 5-second windows, corresponding to a sketch frame rate of 0.2 Hz. We choose MuQ-MuLan features because, as audio representations learned with audio-text contrastive supervision, they capture higher-level musical semantics than codec tokens while remaining temporally aligned with the evolving content of a song. This makes them suitable for representing coarse arrangement information such as section development, changes in density, and dynamic progression.

We then train a residual vector quantizer (RVQ) on these MuQ-MuLan audio features and use the resulting code indices as sketch tokens. In general, let each sketch frame be quantized by L RVQ layers, so that the t-th sketch frame is represented as:

$$\mathbf{s}_t = \left[s_t^{(1)}, s_t^{(2)}, \ldots, s_t^{(L)} \right]. \tag{3}$$

The full song-level sketch sequence is written as $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_T\}$.

Sketch-Conditioned First-Stage Generation. As illustrated in Eq. 1, the first-stage model first predicts the sketch sequence from the conditioning inputs, and then generates mixed audio tokens conditioned on the predicted sketch. In our implementation, the sketch sequence is inserted after the condition representations and before the mixed-audio token sequence. For each sketch frame, we first apply a delay pattern following MusicGen Copet et al. (2023) to the code indices of the L RVQ layers. We then convert the delayed indices of each layer into token embeddings, and sum the resulting layer embeddings to form the sketch representation for that frame. We further introduce learnable sketch-boundary embeddings, *i.e.*, the start of sketch tokens $\langle \mathbf{s}_{\mathrm{s}} \rangle$ and the end of sketch tokens $\langle \mathbf{s}_{\mathrm{e}} \rangle$, so that the first-stage model processes the sequence in the following order:

$$\left[\mathbf{c};\ \langle \mathbf{s}_{\mathrm{s}} \rangle;\ \mathbf{s}_1^{\mathrm{emb}}, \ldots, \mathbf{s}_T^{\mathrm{emb}};\ \langle \mathbf{s}_{\mathrm{e}} \rangle;\ \mathbf{x}_{\mathrm{mix}} \right], \tag{4}$$

where $\mathbf{s}_t^{\mathrm{emb}}$ denotes the summed embedding derived from the L RVQ-layer indices in $\mathbf{s}_t$. This design encourages the model to establish a coarse song-level plan before generating detailed mixed audio tokens.

Two-Phase Training. As learning song-level planning and detailed mixed-audio generation simultaneously is difficult, we adopt a two-phase training strategy, *i.e.*, the model first learns to predict a good sketch and then learns to generate audio conditioned on that sketch. The two phases described below both optimize the first-stage language model. In Phase 1, we optimize the model to predict the sketch sequence from the conditioning inputs alone. In other words, the model learns the mapping from the condition tensors (*i.e.*, structured lyrics with an optional text description or audio prompt) to song-level sketch tokens before learning mixed-audio generation. In this phase, we use only the sketch-token cross-entropy loss $\mathcal{L}_{\mathrm{sk}}$ as supervision, *i.e.*, $\mathcal{L}_{\mathrm{P1}} = \mathcal{L}_{\mathrm{sk}}$.

In Phase 2, we train the full sketch-conditioned mixed-audio generation process with joint sketch and audio supervision. That is, the model is trained on the full mapping from condition tensors to

sketch tokens and then to mixed audio tokens. Let $\mathcal{L}_{\text{mix}}$ denote the token-level cross-entropy loss over mixed audio tokens, and the objective of Phase 2 is:

$$\mathcal{L}_{\text{P2}} = \mathcal{L}_{\text{mix}} + \lambda_{\text{sk}}\mathcal{L}_{\text{sk}}, \tag{5}$$

where λ_{sk} denotes the weight of the sketch-token loss. This phase teaches the model to use the predicted sketch sequence as an intermediate planning scaffold for mixed-audio generation.

At inference time, SketchSong supports two modes. In the default mode, the first-stage model autoregressively predicts the song-level sketch sequence from the condition tensors, and then generates mixed audio tokens conditioned on that sketch. Alternatively, a user can provide a custom sketch sequence, or extract one from a reference song, which exposes song-level planning as an explicit control interface for arrangement-aware generation.

3.3 Fine-Grained Multi-Track Modeling

While song-level sketch planning improves how a song develops over time, full-song generation also requires finer modeling of how individual musical parts are organized in the arrangement. A mixed-song representation alone does not explicitly distinguish the roles of different musical parts within the accompaniment. We therefore model the second stage with four explicit tracks, *i.e.*, vocals, bass, drums, and other instruments.

Four-Track Target Streams. Let $\mathbf{x}_{\text{trk}} = \{\mathbf{x}_{\text{voc}}, \mathbf{x}_{\text{bass}}, \mathbf{x}_{\text{drums}}, \mathbf{x}_{\text{other}}\}$ denote the four target token streams. Given the same condition tensors $\mathbf{c}$ as in the first stage, *i.e.*, the representations derived from structured lyrics and an optional text description or audio prompt, the second stage takes the intermediate mixed-song tokens $\mathbf{x}_{\text{mix}}$ as upstream context and predicts these four streams autoregressively. This finer decomposition is motivated by the fact that a single accompaniment stream entangles multiple instrumental layers, and explicitly separating important instruments (*e.g.*, bass and drums) enables more explicit part-aware arrangement refinement.

Sketch-Informed Hidden-State Conditioning. To inject arrangement information from song-level sketch planning into the second stage, we condition the second-stage language model on the hidden states of the first-stage language model. Specifically, during second-stage training, we feed the sequence constructed in Eq. 4 into the frozen first-stage language model using ground-truth sketch tokens $\mathbf{s}$ and ground-truth mixed-song tokens $\mathbf{x}_{\text{mix}}$, and extract its last-layer hidden states. We denote the resulting hidden-state sequence as:

$$\mathbf{H}_1 = f_{\theta_1}^{\text{hid}}(\mathbf{c}, \mathbf{s}, \mathbf{x}_{\text{mix}}) = [\mathbf{H}_{\text{c}};\ \mathbf{H}_{\text{s}};\ \mathbf{H}_{\text{a}}], \tag{6}$$

where $\mathbf{H}_{\text{c}}$, $\mathbf{H}_{\text{s}}$, and $\mathbf{H}_{\text{a}}$ denote the hidden states corresponding to the condition, sketch, and mixed-audio segments, with their sequence lengths denoted as T_c, T_s, and T_a, respectively. Since the input of the second-stage language model does not contain sketch tokens, we discard the sketch segment and use the aligned hidden-state sequence $\tilde{\mathbf{H}}_1 = [\mathbf{H}_{\text{c}};\ \mathbf{H}_{\text{a}}]$, with a length of $T_c + T_a$.

Multi-Track Input and Prediction. For the second-stage language model, the token streams of vocals, bass, drums, and other instruments are first embedded by four track-specific embedding layers, respectively, and then summed to form a multi-track representation. This multi-track representation is concatenated after the condition sequence to form an intermediate second-stage sequence. We then concatenate this intermediate sequence with $\tilde{\mathbf{H}}_1$ along the feature dimension and map the result back to the hidden size expected by the second-stage model through an MLP, yielding the final input sequence of the second-stage model. In this way, the second-stage language model is jointly conditioned on the condition tensors, the multi-track token context, and the first-stage hidden representation of sketch-conditioned mixed-song generation.

The output hidden states of the second-stage language model are denoted by $\mathbf{H}_2$. We apply four track-specific linear heads to $\mathbf{H}_2$ to predict the token distributions of vocals, bass, drums, and other instruments. Let $\mathcal{L}_{\text{voc}}$, $\mathcal{L}_{\text{bass}}$, $\mathcal{L}_{\text{drums}}$, and $\mathcal{L}_{\text{other}}$ denote the corresponding token-level cross-entropy losses, and the training objective is computed as:

$$\mathcal{L}_{\text{trk}} = \frac{1}{4}\left(\mathcal{L}_{\text{voc}} + \mathcal{L}_{\text{bass}} + \mathcal{L}_{\text{drums}} + \mathcal{L}_{\text{other}}\right). \tag{7}$$

Thus, the second stage is supervised only by the four-track prediction loss, while the intermediate mixed-song representations provide coarse musical context and sketch-informed arrangement cues through the hidden-state conditioning from the first-stage language model.

Table 1: Comparison of different methods on objective evaluation metrics. CE/CU/PC/PQ denote content enjoyment, content usefulness, production complexity, and production quality; Coh./Mus./Mem./Cla./Nat. denote coherence, musicality, memorability, clarity, and naturalness. We include both the released LeVo checkpoint and a LeVo model trained on our dataset and setup, as the latter serves as our direct baseline. Note that all open-source methods (except YuE) adopt post-training, while our current SketchSong does not integrate post-training. Best and second-best results in each column are marked in bold and underline, respectively.

Methods	FAD ↓	MuQ-T ↑	PER ↓	AudioBox Aesthetics ↑				SongEval Aesthetics ↑				
				CE	CU	PC	PQ	Coh.	Mus.	Mem.	Cla.	Nat.
YuE	4.16	0.17	45.16	7.02	7.46	4.86	7.85	3.34	3.16	3.21	3.16	3.08
DiffRhythm 2	3.49	**0.40**	25.6	7.51	7.78	5.68	8.19	<u>3.95</u>	3.67	3.79	<u>3.68</u>	3.54
ACE-Step 1.5	**2.62**	0.27	<u>24.5</u>	7.57	7.88	5.86	8.21	**4.16**	**4.11**	**4.09**	**4.04**	**3.73**
LeVo (released)	3.79	0.24	**9.3**	7.71	7.93	5.85	8.40	3.92	<u>3.75</u>	<u>3.83</u>	3.66	<u>3.58</u>
LeVo (our trained)	3.73	0.29	32.3	7.61	7.83	5.80	8.35	3.52	3.42	3.38	3.36	3.25
+ Sketch planning	<u>3.05</u>	<u>0.33</u>	29.7	<u>7.72</u>	**7.95**	<u>6.11</u>	8.31	3.49	3.69	3.65	3.60	3.51
+ Multi-track modeling	3.82	0.29	31.4	7.58	7.90	5.98	**8.43**	3.88	3.35	3.36	3.28	3.20
SketchSong	3.06	0.32	27.8	**7.76**	<u>7.94</u>	**6.27**	<u>8.42</u>	3.84	3.59	3.67	3.65	3.48

3.4 Training and Inference

SketchSong is trained in a stage-wise manner. We first optimize the first-stage language model with the aforementioned two-phase training strategy. In Phase 1, the model predicts sketch tokens from the conditioning inputs, and is supervised only by $\mathcal{L}_{\mathrm{P1}}$. We then continue training the same model in Phase 2 on the full sketch-conditioned mixed-audio generation process, where the model is supervised by $\mathcal{L}_{\mathrm{P2}}$ in Eq. 5. After the first-stage model is trained, we then train the second-stage language model for four-track prediction with the loss defined in Eq. 7 while keeping the first-stage model frozen for hidden-state conditioning.

At inference time, SketchSong supports two modes at the first stage. In the default mode, the first-stage model predicts a song-level sketch sequence from the condition tensors, and then generates mixed-song audio tokens conditioned on that sketch. Alternatively, a user can provide a sketch sequence, in which case the first-stage model directly generates mixed-song audio tokens conditioned on the user-specified sketch. After the first stage finishes, the second-stage language model takes the same conditioning inputs together with the mixed-song representation generated by the first stage, and autoregressively predicts four-track tokens for vocals, bass, drums, and other instruments.

Finally, we reuse the same MuCodec-based waveform reconstruction pipeline as LeVo Lei et al. (2025) to decode the predicted multi-track tokens into audio. The vocal stream is decoded with the vocal codec, whereas the bass, drums, and other streams are decoded with the accompaniment codec, since dedicated codecs for these three tracks are not separately trained in the current system.

4 Experiments

4.1 Experimental Setup

Dataset. We train SketchSong on a corpus of 1 million songs (approximately 54k hours). Following the automated SongPrep preprocessing pipeline Tan et al. (2025), we construct structured lyrics, section-level structure annotations, and separated tracks for each song. Specifically, we use Demucs Rouard et al. (2023) to decompose each song into four tracks, *i.e.*, vocals, bass, drums, and other instruments. For music structure annotation, we adopt an All-In-One model enhanced with Dual-Path RNN (DPRNN) blocks Kim & Nam (2023) to predict section boundaries, section labels, and their timestamps. For lyric transcription, we directly use the SongPrep-provided recognition pipeline, which combines WER-FIX-based lyric refinement, a Zipformer ASR model Yao et al. (2023) with a dual-head ASR framework. We further apply a wav2vec 2.0-based word alignment

module Baevski et al. (2020) to calibrate structural outputs and correct timestamp mismatches. In addition, we generate a open-vocabulary text description for each song using Qwen2.5-Omni Xu et al. (2025a).

Evaluation. We evaluate SketchSong with both objective and subjective metrics. For objective evaluation, we report Fréchet Audio Distance (FAD) Kilgour et al. (2018) and Phoneme Error Rate (PER). To compute PER, we first extract the vocal track with Demucs Rouard et al. (2023) and then transcribe it using Whisper-large-v2 Radford et al. (2023). We also measure the similarity between the generated song and the input text description (denoted as "MuQ-T") based on pre-trained MuQ-MuLan representations Huang et al. (2022). For aesthetic evaluation, we report: 1) Audiobox-Aesthetic Tjandra et al. (2025), including content enjoyment (CE), content usefulness (CU), production complexity (PC), and production quality (PQ); and 2) SongEval Yao et al. (2025a), including coherence, musicality, memorability, clarity, and naturalness. For these objective metrics, we randomly sample 100 Chinese songs and 100 English songs from the validation set to provide conditioning inputs for generating songs. For subjective evaluation, we conduct a mean opinion score (MOS) listening test. Each listener is randomly assigned 10 test cases, and samples from different methods for the same case share identical conditioning inputs. We invite 5 music experts and 10 amateur listeners to rate generated songs on a 1–5 scale over five dimensions, *i.e.*, overall quality (OVL, focusing on musicality and naturalness), song structure clarity (SSC), arrangement development (AD), instrumental richness (IR), and vocal quality (VQ).

Comparison Systems. Our primary baseline is LeVo Lei et al. (2025). Since the original LeVo training data are not publicly available, and LeVo also includes an additional DPO-based post-training stage that is not used in the current SketchSong system, we re-train a LeVo baseline on our own dataset under the same setup for fair comparison. We also include the performance of the released LeVo checkpoint in the experimental results. Beyond LeVo, we compare with several open-source song generation systems, including YuE Yuan et al. (2025), DiffRhythm 2 Jiang et al. (2025), and ACE-Step 1.5 Gong et al. (2026). For these open-source baselines, we use their released checkpoints and default configurations.

4.2 Objective Results

Main Results. Table 1 jointly reports results against open-source comparison systems and our internal ablations. Compared to LeVo trained under our setup, SketchSong improves on most metrics, reducing FAD from 3.73 to 3.06 and PER from 32.3 to 27.8, while improving MuQ-T from 0.29 to 0.32. It also achieves the best aesthetic scores on CE and PC, and the second-best scores on CU and PQ, among all compared methods. Notably, all open-source comparison systems (except YuE) in Table 1 are evaluated with their released post-trained checkpoints, whereas the current SketchSong does not use additional post-training such as dedicated lyric alignment, text-prompt alignment, or reward-based aesthetic optimization. As a result, while still lagging behind some released systems on PER and several SongEval dimensions, SketchSong is already competitive on FAD and several aesthetic metrics, demonstrating the effectiveness of our design. More comparisons are included in the supplementary.

Ablation Study. The ablation rows further show that the two proposed components contribute in different ways. Adding sketch planning on top of LeVo leads to the largest gains on FAD, CE, CU, musicality, and memorability, which is consistent with our motivation that song-level sketch planning mainly improves long-range arrangement development, thus generating songs with clearer progression and stronger overall musicality and memorability. By contrast, introducing fine-grained multi-track modeling on top of LeVo's dual-track paradigm most clearly improves production complexity and production quality, matching our expectation that finer part decomposition mainly benefits arrangement richness and production realism. When combined, the full SketchSong model achieves the best overall trade-off across these two dimensions.

At the same time, the "+ Multi-track modeling" variant alone does not improve FAD and only yields limited gain on PER. We attribute this to two reasons. First, multi-track modeling mainly refines the accompaniment, so its direct effect on vocal transcription accuracy is limited. Second, in the current system only the second-stage language model is made four-track-aware, while the generated bass, drums, and other tracks are still decoded by the shared LeVo accompaniment codec and then remixed with a simple $1/3$ amplitude scaling for each stream. This likely leads to less natural mixtures after remixing and can hurt FAD. Training a dedicated multi-track codec and diffusion decoder for

Table 2: MOS listening test results on subjective metrics. OVL/SSC/AD/IR/VQ denote overall quality, song structure clarity, arrangement development, instrumental richness, and vocal quality, respectively. Best and second-best results in each column are marked in bold and underline, respectively.

Method	MOS ↑				
	OVL	SSC	AD	IR	VQ
LeVo (our trained)	3.60	3.76	3.62	3.56	<u>3.82</u>
+ Sketch planning	<u>3.64</u>	**3.88**	**3.90**	3.62	**3.88**
+ Multi-track modeling	3.58	3.70	3.60	<u>3.78</u>	3.78
SketchSong	**3.68**	<u>3.84</u>	<u>3.87</u>	**3.84**	3.80

separated tracks is a promising direction for improving overall generation quality, which we point to as future work.

4.3 Subjective Results

Table 2 reports the MOS listening test results. Compared with LeVo trained under our setup, SketchSong improves the overall quality score from 3.60 to 3.68 and the instrumental richness score from 3.56 to 3.84, indicating that the proposed design yields a clear perceptual benefit in both overall listening experience and arrangement richness. It also remains competitive on song structure clarity, arrangement development, and vocal quality.

The subjective ablations again reveal complementary effects of the two designs. Adding sketch planning yields the clearest gains on song structure clarity, arrangement development, and vocal quality, which aligns with the role of sketch planning in clarifying section progression and improving perceived song development. Adding multi-track modeling most clearly improves instrumental richness, suggesting that explicit bass, drums, and other-instrument modeling helps listeners perceive denser and more structured accompaniment without reducing much vocal quality. The full SketchSong model combines these advantages, achieving the best overall quality and instrumental richness while remaining close to the best sketch-only variant on structure-related metrics.

5 Conclusion

In this paper, we presented SketchSong, a hierarchical song generation framework designed to improve full-song generation from two complementary aspects: long-range arrangement development and fine-grained part-aware modeling. To address the difficulty of organizing high-level song development while generating dense audio tokens, SketchSong introduces song-level sketch planning into the first stage, so that the model first predicts compact sketch tokens and then generates mixed-song audio conditioned on this plan. To better model musical roles and interactions within an arrangement, SketchSong further extends the second stage from a coarse vocal–accompaniment formulation to fine-grained four-track modeling over vocals, bass, drums, and other instruments. Experiments show that SketchSong consistently improves over a strong LeVo baseline trained under the same setup, and achieves competitive performance against open-source systems on both objective metrics and human listening tests. The ablation results further confirm that sketch planning mainly improves long-range coherence and arrangement development, while fine-grained multi-track modeling mainly enhances arrangement richness and production quality. The current system still has limitations, including the lack of additional post-training and the reuse of the accompaniment-side decoding pipeline for multiple non-vocal tracks. Future work may investigate dedicated post-training strategies for improved preference and condition alignment, jointly learned multi-track neural codecs that better preserve track-specific characteristics and inter-track dependencies, and more powerful waveform decoders that enhance audio fidelity and reduce synthesis artifacts. These directions could further improve the quality, structural coherence and fine-grained controllability of complete songs.

AI USE STATEMENT

We used generative AI tools to revise selected portions of the manuscript for clarity, concision, grammatical correctness, and academic style. We did not use generative AI tools to generate synthetic data, develop theoretical or conceptual frameworks, formulate mathematical claims, propose hypotheses, design the methodology or experiments, implement the proposed method, process or analyze data, or interpret experimental results.

All AI-assisted revisions were manually reviewed and edited by the authors. The authors independently verified all technical descriptions, experimental results, claims, and citations against the implementation, experimental records, and original literature. No experimental results, numerical values, or references were generated by generative AI. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

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.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Ye Bai, Haonan Chen, Jitong Chen, Zhuo Chen, Yi Deng, Xiaohong Dong, Lamtharn Hantrakul, Weituo Hao, Qingqing Huang, Zhongyi Huang, et al. Seed-music: A unified framework for high quality and controlled music generation. *arXiv preprint arXiv:2409.09214*, 2024.

Pengfei Cai, Joanna Wang, Haorui Zheng, Xu Li, Zihao Ji, Teng Ma, Zhongliang Liu, Chen Zhang, and Pengfei Wan. Segtune: Structured and fine-grained control for song generation. *arXiv preprint arXiv:2510.18416*, 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*, 36:47704–47720, 2023.

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

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

Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel PW Ellis. Mulan: A joint embedding of music audio and natural language. *arXiv preprint arXiv:2208.12415*, 2022.

Changhao Jiang, Jiahao Chen, Zhenghao Xiang, Zhixiong Yang, Hanchen Wang, Jiabao Zhuang, Xinmeng Che, Jiajun Sun, Hui Li, Yifei Cao, et al. Muse: Towards reproducible long-form song generation with fine-grained style control. *arXiv preprint arXiv:2601.03973*, 2026.

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

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Taejun Kim and Juhan Nam. All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio. In *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5. IEEE, 2023.

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

Yun-Han Lan, Wen-Yi Hsiao, Hao-Chung Cheng, and Yi-Hsuan Yang. Musicongen: Rhythm and chord control for transformer-based text-to-music generation. *arXiv preprint arXiv:2407.15060*, 2024.

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

Shun Lei, Yaoxun Xu, Zhiwei Lin, Huaicheng Zhang, Wei Tan, Hangting Chen, Jianwei Yu, Yixuan Zhang, Chenyu Yang, Haina Zhu, et al. Levo: High-quality song generation with multi-preference alignment. *arXiv preprint arXiv:2506.07520*, 2025.

Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward controllable text-to-music generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8293–8316, 2024.

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, Janne Spijkervet, Katerina Kosta, Furkan Yesiler, Boris Kuznetsov, Ju-Chiang Wang, Matt Avent, Jitong Chen, and Duc Le. Stemgen: A music generation model that listens. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1116–1120. IEEE, 2024.

KR Prajwal, Bowen Shi, Matthew Lee, Apoorv Vyas, Andros Tjandra, Mahi Luthra, Baishan Guo, Huiyu Wang, Triantafyllos Afouras, David Kant, et al. Musicflow: Cascaded flow matching for text guided music generation. *arXiv preprint arXiv:2410.20478*, 2024.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

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

Or Tal, Alon Ziv, Itai Gat, Felix Kreuk, and Yossi Adi. Joint audio and symbolic conditioning for temporally controlled text-to-music generation. *arXiv preprint arXiv:2406.10970*, 2024.

Wei Tan, Shun Lei, Huaicheng Zhang, Guangzheng Li, Yixuan Zhang, Hangting Chen, Jianwei Yu, Rongzhi Gu, and Dong Yu. Songprep: A preprocessing framework and end-to-end model for full-song structure parsing and lyrics transcription. *arXiv preprint arXiv:2509.17404*, 2025.

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.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):5314–5321, 2022.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report, 2025a. URL https://arxiv.org/abs/2503.20215.

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, 2025b.

Chenyu Yang, Shuai Wang, Hangting Chen, Wei Tan, Jianwei Yu, and Haizhou Li. Songbloom: Coherent song generation via interleaved autoregressive sketching and diffusion refinement. *arXiv preprint arXiv:2506.07634*, 2025.

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*, 2025a.

Yao Yao, Peike Li, Boyu Chen, and Alex Wang. Jen-1 composer: A unified framework for high-fidelity multi-track music generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14459–14467, 2025b.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zengrui Jin, Long Lin, and Daniel Povey. Zipformer: A faster and better encoder for automatic speech recognition. *arXiv preprint arXiv:2310.11230*, 2023.

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.

Yu Zhang, Wenxiang Guo, Changhao Pan, Zhiyuan Zhu, Ruiqi Li, Jingyu Lu, Rongjie Huang, Ruiyuan Zhang, Zhiqing Hong, Ziyue Jiang, et al. Versatile framework for song generation with prompt-based control. *arXiv preprint arXiv:2504.19062*, 2025.

A Appendix

A.1 Implementation Details.

Our implementation follows the two-stage autoregressive framework of LeVo Lei et al. (2025), while introducing song-level sketch planning in the first stage and four-track refinement in the second stage. We train both language models from scratch. For sketch token construction, we extract MuQ-MuLan audio features from non-overlapping 5-second windows, and train an RVQ codebook over these features. The RVQ adopts four quantizers (*i.e.*, $L = 4$), a codebook size of 4096, and an embedding dimension of 512. Its encoder and decoder both contain four ResMLP layers Touvron et al. (2022) with a hidden dimension of 2048. We train the RVQ for 50 epochs with batch size 512 and learning rate 3×10^{-4}, using cosine reconstruction loss together with commitment loss. The learned code indices are then used as song-level sketch tokens. The two language models are both trained on 64 NVIDIA A100 GPUs with batch size 2 per GPU. For the first-stage language model, both Phase 1 and Phase 2 are trained for 100k steps. The second-stage language model is trained for 50k steps. All stages use 4k warmup steps. The embedding layers for conditioning inputs, sketch tokens, and mixed-audio or four-track tokens are all trainable. Components not directly related to our two main contributions, including the codec-based waveform decoder pipeline, are kept fixed from the open-source checkpoint of LeVo. Unless otherwise stated, other settings follow LeVo, and more detailed configurations are provided in the supplementary material.

A.2 Generated Song Audio Demos

We provide generated song audio samples as supplementary attachments to facilitate qualitative evaluation. The audio files are located in the `text_to_song_generated_cases/` folder of the supplementary material. We include four representative cases spanning diverse genres: **song_01** (Trap / boom-bap), **song_02** (J-Rock / Pop-Punk), **song_03** (K-Pop ballad), and **song_04** (Indie Rock / Alternative Rock). For each case, we provide the generated songs from all four ablation variants: LeVo (our trained baseline), + Sketch Planning, + Multi-Track Modeling, and the full SketchSong. Each folder also contains a `meta.json` file listing the text prompt and lyrics used for generation. We encourage reviewers to listen to these samples for a direct perceptual comparison.

Table 3: Comparison of different methods on objective evaluation metrics under the *audio-to-song* setting. CE/CU/PC/PQ denote content enjoyment, content usefulness, production complexity, and production quality; Coh./Mus./Mem./Cla./Nat. denote coherence, musicality, memorability, clarity, and naturalness. Best and second-best results in each column are marked in bold and underline, respectively.

Methods	FAD ↓	MuQ-A ↑	PER ↓	AudioBox Aesthetics ↑				SongEval Aesthetics ↑				
				CE	CU	PC	PQ	Coh.	Mus.	Mem.	Cla.	Nat.
LeVo (our trained)	<u>2.53</u>	<u>0.74</u>	0.39	<u>7.17</u>	7.56	5.85	8.06	<u>3.32</u>	3.03	<u>3.12</u>	3.08	<u>3.07</u>
+ Sketch planning	**2.34**	**0.76**	0.41	**7.40**	**7.72**	**6.11**	8.07	**3.34**	**3.14**	**3.16**	**3.16**	**3.11**
+ Multi-track modeling	3.03	0.70	**0.35**	7.02	7.64	5.64	<u>8.11</u>	3.21	2.91	3.08	<u>3.12</u>	2.99
SketchSong	2.65	0.73	<u>0.38</u>	7.12	<u>7.68</u>	<u>6.07</u>	**8.19**	3.25	<u>3.06</u>	3.06	<u>3.12</u>	3.04

A.3 Audio-to-Song Generation Results

In the main paper, we evaluate all methods under the *text-to-song* setting, where a text description serves as the conditioning input. Here, we further report results under the *audio-to-song* setting, where an audio reference is provided instead of a text description. In this setting, we replace the text-audio similarity metric MuQ-T used in the main paper with MuQ-A, which measures the cosine similarity between the MuQ-MuLan audio features of the input audio prompt and the generated song. We focus on comparing our internal ablation variants. All other evaluation protocols remain identical to those described in the main paper. Table 3 summarizes the results.

Consistent findings with the text-to-song setting. Several observations from the audio-to-song results align well with the conclusions drawn in the main paper. First, adding sketch planning on top of LeVo consistently improves nearly all evaluation metrics. In particular, the "+ Sketch planning" variant achieves the best FAD (2.34 vs. 2.53 for the baseline), and leads to notable gains on AudioBox Aesthetics dimensions such as CE (7.40 vs. 7.17), CU (7.72 vs. 7.56), and PC (6.11 vs. 5.85). Similar improvements are also observed across all five SongEval dimensions. These results suggest that sketch planning remains effective even when an audio prompt is provided, helping the model produce songs with better overall musical quality and aesthetic appeal.

Differences from the text-to-song setting. We also observe several noteworthy differences compared to the text-to-song setting. First, the contribution of multi-track modeling appears more limited in the audio-to-song setting. While the "+ Multi-track modeling" variant achieves the best PER (0.35) and competitive PQ (8.11), it underperforms the baseline on FAD (3.03 vs. 2.53) and most SongEval dimensions (e.g., musicality drops from 3.03 to 2.91, naturalness from 3.07 to 2.99). As discussed in the main paper, we attribute this primarily to the fact that our current system reuses LeVo's accompaniment codec to decode the bass, drums, and other instrument tracks, without training a dedicated multi-track codec. This shared decoder may struggle to faithfully reconstruct separated tracks, and the subsequent remixing with simple amplitude scaling could introduce artifacts that negatively affect both FAD and perceptual quality metrics. Second, the overall improvement from sketch planning is somewhat less pronounced than in the text-to-song setting. For instance, MuQ-A scores are fairly similar across all variants (ranging from 0.70 to 0.76), suggesting that with an explicit audio reference as the conditioning input, the model already receives strong stylistic guidance, which reduces the relative benefit of sketch-level planning for capturing global style characteristics. In contrast, under the text-to-song setting where the conditioning signal is a more abstract text description, sketch planning plays a more critical role in bridging the gap between the text prompt and the desired musical style. Finally, the full SketchSong model still achieves a favorable trade-off by combining both components, obtaining the best PQ (8.19) and competitive results on FAD (2.65), CU (7.68), and PC (6.07), while maintaining reasonable performance on the remaining metrics.

A.4 Case Studies on Generated Audio and Sketch Controllability

Mel-Spectrogram Visualization. Fig. 3 provides a qualitative case study to further illustrate how SketchSong affects the generated audio.

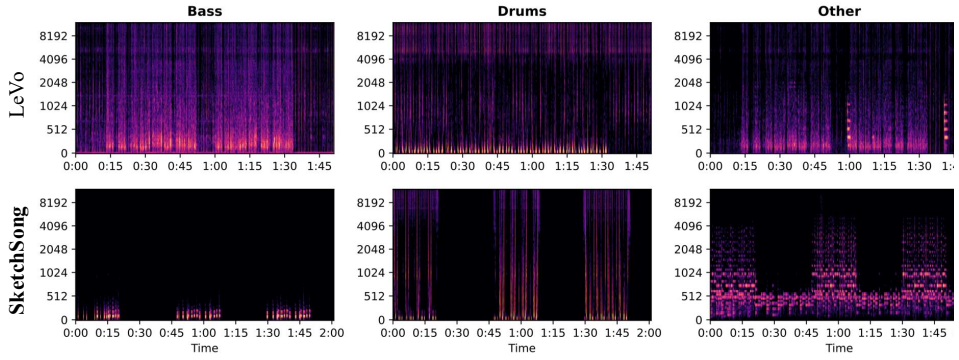


Figure 3: Case study on separated mel-spectrograms of generated songs. For the same test case, we first generate a song with LeVo and SketchSong, respectively, and then re-apply source separation to each generated waveform to obtain the bass, drums, and other-instrument tracks for visualization. Compared with LeVo, SketchSong produces clearer frequency-role separation across tracks and more structured time-varying arrangement development over the song.

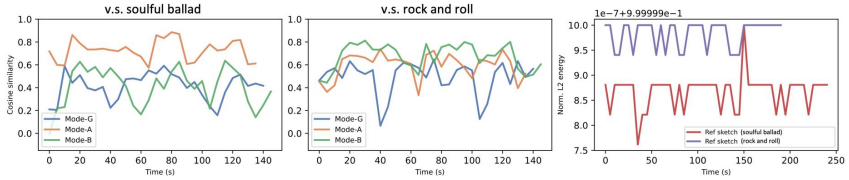


Figure 4: Sketch controllability under three inference modes. With the same text prompt (*i.e.*, *hip-hop*) fixed, replacing the sketch with a *soulful ballad* or *rock & roll* reference shifts the generated audio toward the corresponding reference in MuQ-MuLan similarity and energy, showing that sketch substitution provides effective control over the final song.

From the perspective of fine-grained multi-track modeling, the contrast is clear. In the LeVo result, the separated bass spectrogram spreads across almost the full frequency range, which is not physically plausible for a bass-dominant track, since bass content should mainly concentrate in low frequencies. This suggests that bass-related content is entangled with other instrumental components and is therefore difficult to separate cleanly, matching the coarse-track motivation illustrated in Fig. 1(a). By contrast, in SketchSong the separated bass, drums, and other-instrument tracks are much more clearly organized: the bass is mainly concentrated in low frequencies, while the drums cover a broader frequency range, which is reasonable because drums include both low-frequency kick components and high-frequency percussive content such as hi-hats and cymbals. This indicates that fine-grained multi-track modeling helps the model generate more explicit and separable instrumental roles, especially for bass and drums.

The same case also reveals the effect of song-level sketch planning on arrangement development over time. In the LeVo result, the spectrograms of different instrumental tracks remain relatively flat across the full duration, with little obvious structural change along the time axis. In other words, the instrumental energy tends to remain spread over the whole song without clear section-dependent variation. In SketchSong, however, the time-varying structure is much more apparent. For example, in the other-instrument track, the intro section (roughly 0:00–0:20) and the chorus sections (around 0:50–1:10) show fuller and richer spectral coverage, reflecting denser instrumentation and stronger emotional buildup; these sections are also accompanied by more active bass and drums. By contrast, in the verse sections (roughly 0:20–0:50 and 1:10–1:30), the spectral energy is noticeably sparser and mainly concentrated in the mid-to-low range, which is consistent with a softer pad-like accompaniment and the near absence of bass and drums. This suggests that sketch planning helps the model organize section-level development more purposefully, leading to clearer changes in density, instrumentation, and emotional progression across the song.

Table 4: MuQ-MuLan audio-feature cosine similarity statistics over 20 diverse prompts under three inference modes. "vs. Ballad" and "vs. Rock" denote the cosine similarity between the MuQ-MuLan audio features of each generated song and those of the *soulful ballad* and *rock & roll* reference songs, respectively. "vs. Orig." denotes the similarity to the original reference song associated with each prompt (only applicable to Mode-G). Higher values indicate stronger stylistic alignment with the corresponding reference. Best result in each column is marked in bold.

Mode	vs. Ballad ↑	vs. Rock ↑	vs. Orig. ↑
Mode-G (model sketch)	0.643 ± 0.120	0.643 ± 0.083	0.662 ± 0.156
Mode-A (ballad sketch)	$\mathbf{0.828 \pm 0.048}$	0.603 ± 0.038	—
Mode-B (rock sketch)	0.558 ± 0.069	$\mathbf{0.770 \pm 0.036}$	—

Sketch Control. To further verify that song-level sketch planning provides user-level controllability, we compare three inference modes while keeping the same text prompt fixed: *hip-hop*. In **Mode-G**, the model predicts its own sketch from the text prompt and then generates songs conditioned on that sketch. In **Mode-A**, the text prompt remains unchanged, but the sketch is replaced with a fixed sketch extracted from a *soulful ballad* song. In **Mode-B**, the text prompt also remains unchanged, but the sketch is replaced with a fixed sketch extracted from a *rock & roll* song. Fig. 4 visualizes how these different sketch conditions influence the final generated audio through MuQ-MuLan-based analyses.

The left and middle panels show reference-audio similarity under the three modes. In the left panel, we compute the MuQ-MuLan audio-feature similarity between each generated song and the original *soulful ballad* reference song. Mode-A consistently produces the highest similarity, even though the text prompt itself does not describe a soulful ballad. In the middle panel, we instead compute the similarity to the original *rock & roll* reference song, and Mode-B becomes the highest. This indicates that replacing the sketch alone can strongly steer the style of the generated song toward the reference associated with that sketch, even when the text prompt remains unchanged and does not explicitly match the injected sketch style.

The right panel further shows that sketch replacement also changes the energy profile of the generated audio. Comparing the MuQ-MuLan feature energy of the generated songs conditioned on the *soulful ballad* sketch and the *rock & roll* sketch, the rock-conditioned results exhibit substantially higher energy than the soulful-ballad-conditioned results. This is consistent with the expected difference between the two sketch styles and further supports that the sketch does not merely affect a subtle latent representation, but has a strong and measurable impact on the final generated audio. Taken together, these results provide direct evidence that song-level sketch planning functions as an effective control interface: users can modify the overall style and high-level musical development of the generated song by replacing the sketch, even under the same text prompt.

A.5 Quantitative Sketch Controllability Analysis

In the preceding qualitative case study, we present a qualitative case study on sketch controllability using a single text prompt (*hip-hop*) and two reference sketch sources (*soulful ballad* and *rock & roll*), illustrating through MuQ-MuLan similarity curves and energy profiles that replacing the sketch can steer the style of the generated song. Here, we extend this analysis to a larger-scale quantitative evaluation over 20 diverse prompts spanning a wide range of genres and languages (including Chinese pop, J-pop, rock, rap, electronic, ballad, wedding songs, etc.), and report the MuQ-MuLan audio-feature similarity with mean and standard deviation to provide more robust evidence for the controllability claim.

A.5.1 Experimental Setup

We design three inference modes, following the same protocol as in the main paper but scaling to 20 prompts:

- **Mode-G** (model-Generated sketch): For each prompt, the model first predicts sketch tokens from the given text prompt, and then generates the song conditioned on these model-predicted sketch tokens. This represents the standard inference pipeline of SketchSong.
- **Mode-A** (reference sketch A — *soulful ballad*): For each prompt, the text prompt remains unchanged, but the sketch tokens are replaced with a fixed sketch extracted from a *soulful ballad* reference song, characterized by soft rock, piano, acoustic guitar, slow tempo, and emotional atmosphere. That is, the model skips its own sketch prediction step and directly uses the externally provided *soulful ballad* sketch as the planning signal, even when the text prompt describes a completely different genre (*e.g.*, EDM or rap). The model then generates the song conditioned on both the original text prompt and this injected sketch.
- **Mode-B** (reference sketch B — *rock & roll*): Similarly, the text prompt remains unchanged, but the sketch tokens are replaced with a fixed sketch extracted from a *rock & roll* reference song, characterized by synthesizer, drums, clapping, fast tempo, upbeat, and energetic style.

All three modes use identical inference hyperparameters and operate under the *text-to-song* setting (*i.e.*, only a text prompt is provided as the conditioning input, without any audio prompt). The 20 text prompts used in this experiment are not manually written; instead, they are automatically generated captions produced by Qwen2.5-Omni Xu et al. (2025a) from 20 reference songs that span diverse genres and languages (including Chinese pop, J-pop, rock, rap, electronic, ballad, wedding songs, etc.). For evaluation, we compute the **MuQ-MuLan audio-feature cosine similarity** between each generated song and the reference songs: we extract MuQ-MuLan audio features from non-overlapping 5-second windows and compute the cosine similarity against the MuQ-MuLan features of the two fixed reference songs (*soulful ballad* and *rock & roll*). We report the mean and standard deviation across all 20 prompts. For Mode-G, we additionally report the similarity between each generated song and its corresponding original reference song (*i.e.*, the specific song from which the text prompt was derived via captioning), denoted as "vs. Orig.". This metric reflects how well the model-predicted sketch captures the stylistic characteristics of the intended source song through the text prompt alone.

A.5.2 Results and Analysis

Table 4 reports the MuQ-MuLan audio-feature similarity statistics across the three modes.

Sketch replacement effectively steers generation style. The most prominent finding is that replacing the sketch tokens with those from a specific reference song strongly shifts the generated audio toward the style of that reference, regardless of the original text prompt. When injecting the *soulful ballad* sketch (Mode-A), the mean similarity to the ballad reference reaches 0.828, substantially higher than Mode-G's 0.643 and Mode-B's 0.558. Conversely, when injecting the *rock & roll* sketch (Mode-B), the mean similarity to the rock reference reaches 0.770, again notably higher than Mode-G's 0.643 and Mode-A's 0.603. This pattern is consistent across all 20 prompts and holds for prompts from very different genres (*e.g.*, Chinese pop, J-pop, rap, electronic), confirming that the sketch controllability observed in the main paper's single-case study is not an isolated or cherry-picked phenomenon, but generalizes robustly across diverse inputs.

Injecting a mismatched sketch reduces similarity to the other reference. The results also reveal a clear complementary pattern: injecting one reference sketch not only increases similarity to that reference, but simultaneously *decreases* similarity to the other reference. Specifically, Mode-A achieves the highest similarity to the ballad reference (0.828) but the lowest similarity to the rock reference (0.603), while Mode-B shows the opposite pattern (0.558 vs. 0.770). This bidirectional shift provides additional evidence that the sketch exerts genuine stylistic control, rather than simply adding a generic quality boost.

Model-predicted sketches show prompt-adaptive behavior. Under Mode-G, the generated songs exhibit roughly balanced similarity to both reference songs (0.643 vs. 0.643), with relatively higher standard deviations (0.120 and 0.083) compared to Mode-A and Mode-B. This is expected: since Mode-G allows the model to freely predict sketches from diverse text prompts, the generated songs naturally vary widely in style depending on the prompt content. The moderate similarity to both references indicates that Mode-G does not systematically bias toward either style. Furthermore, the mean similarity between Mode-G generations and their own original reference songs (vs. Orig. =

0.662) is the highest among the three similarity columns for Mode-G, suggesting that the model's predicted sketches do capture prompt-relevant stylistic information reasonably well.

Low variance under sketch injection indicates reliable control. Another noteworthy observation is that Mode-A and Mode-B exhibit substantially lower standard deviations than Mode-G. For instance, Mode-A's similarity to the ballad reference has a standard deviation of only 0.048, compared to Mode-G's 0.120. This suggests that externally provided sketches serve as a strong and reliable conditioning signal that overrides much of the prompt-induced stylistic variation, leading to more consistent and predictable generation outcomes. This property is desirable for practical applications where users wish to precisely control the overall style of the generated song.

Taken together, these quantitative results over 20 diverse prompts provide robust support for the claim that song-level sketch tokens function as an effective and reliable control interface in SketchSong. By simply replacing the sketch at inference time, users can meaningfully steer the global style and high-level musical development of the generated song, complementing the text prompt as an additional axis of creative control.

Audio demo. We provide a sketch controllability audio demo in the `sketch_control_case/` folder of the supplementary attachments. The folder contains generated songs under all three modes (Mode-G, Mode-A, and Mode-B) for the same text prompt, along with a `meta.json` file listing the prompt and lyrics. Note that the provided audio samples are decoded directly from the mixed audio tokens produced by the first-stage language model, *without* passing through the second-stage model. This isolates the effect of sketch tokens on the generation, allowing reviewers to more directly perceive how replacing the sketch alters the overall style, tempo, and energy of the generated song. We encourage reviewers to listen and compare the three modes.