

JAM: A Tiny Flow-based Song Generator with Fine-grained Controllability and Aesthetic Alignment

Renhang Liu¹, Chia-Yu Hung¹, Navonil Majumder¹, Dorien Herremans², Soujanya Poria¹

{renhang001, chiayu001}@e.ntu.edu.sg,
{navonil.majumder, soujanya.poria}@ntu.edu.sg
dorien_herremans@sutd.edu.sg

¹Nanyang Technological University ²Singapore University of Technology and Design

Abstract

Diffusion and flow-matching models have revolutionized automatic text-to-audio generation, increasingly capable of generating high-quality audio outputs for speech and acoustic events. However, creative audio generation involving music and songs still has room for improvement. Recent open lyrics-to-song models like DiffRhythm, ACE-Step, and LeVo have set an acceptable standard for recreational use but lack the fine-grained word-level controllability desired by musicians. To our knowledge, our flow-matching-based JAM is the first to provide word-level timing and duration control in song generation, enabling fine-grained vocal control. To better align generated songs with human preferences, we implement aesthetic alignment through Direct Preference Optimization, iteratively refining the model using a synthetic dataset without manual annotations. Additionally, we aim to standardize evaluation of lyrics-to-song models through our public evaluation dataset JAME. We demonstrate that JAM outperforms open models on music-specific attributes.

1 Introduction

Music plays an essential role in human culture, bringing people together and enriching our lives through shared emotional experiences. From ancient rituals to modern celebrations, music has shaped our social bonds and personal well-being (Freeman III, 1998; Agres et al., 2021).

Creating music is a complex, time-consuming process requiring skilled musicians. Recent advances in generative models have made AI-generated music a reality, providing composers with reliable first drafts to accelerate creative workflows.

Recent audio-based generative music AI models follow three main trends:

- *Singing voice generation* produces vocal performances from lyrics and musical notes, often with

voice cloning.

- *Textual description to music generation* translates mood or feature prompts into synthesized instrumentals, similar to text-to-audio models like Tango (Ghosal et al., 2023; Majumder et al., 2024; Hung et al., 2025), AudioLDM (Liu et al., 2024), and Stable Audio Open (Evans et al., 2024).
- *Lyrics-to-song generation* combines vocals and music to produce full songs, ensuring coherence between lyrics and instruments.

Lyrics-to-song generation presents unique challenges in aligning linguistic content with musical structure while preserving prosody and generating high-quality audio across longer durations.

Existing approaches fall into two categories: *Autoregressive models* generate audio sequentially, maintaining long-range coherence but with slow decoding. *Diffusion-based methods* iteratively denoise latent representations, offering high fidelity and flexible conditioning but struggling with generation speed and fine-grained lyric-audio alignment.

Recent systems like DiffRhythm (Ning et al., 2025), YuE (Yuan et al., 2025), LeVo (Lei et al., 2025), and ACE-Step (Gong et al., 2025) are limited by: (1) **Large model size** causing slow inference; (2) **Coarse timing control** with no word-level alignment; (3) **Weak lyric fidelity** with high WER/PER; and (4) **Lack of duration control** over song length.

To address these challenges, we introduce JAM,¹ a 530M-parameter rectified-flow model for lyrics-to-song generation. JAM uses conditional flow-matching (Lipman et al., 2023) jointly conditioned on: (i) *Lyrics* with fine-grained word/phoneme-level timing for precise prosody control; (ii) *Target duration* for overall song length and vocal phrase spacing; (iii) *Style prompt* (reference audio or text) for timbral and structural characteristics. JAM gen-

¹Code, model weights, and the JAME benchmark are available at <https://github.com/declare-lab/jamify>.

erates full songs at 44.1kHz up to 3:50 long through a rectified flow trajectory in VAE latent space.

Our contributions are as follows. First, JAM is a compact architecture with only 530M parameters—half the size of DiffRhythm-1.1B—enabling faster inference. Second, we introduce fine-grained temporal control by accepting word-level timing inputs that allow users to control vocal placement and inter-word pacing, while our global duration mechanism specifies instrumental introduction and coda lengths, giving composers full control over song structure. This precise alignment reduces WER and PER by over 3× compared to prior work. Third, we implement aesthetic alignment through Direct Preference Optimization. Unlike LeVo (Lei et al., 2025), which relies on difficult-to-scale human annotations, we use SongEval (Yao et al., 2025) to generate synthetic preference labels, inspired by Tango2 (Majumder et al., 2024) and TangoFlux (Hung et al., 2025). Finally, we introduce JAME, a rigorous evaluation benchmark of 250 tracks across five genres, all released after baseline training cut-offs to avoid data contamination.

2 Related Works

Music Generation. Music generation aims to generate coherent and aesthetically pleasing musical content, conditioned on various inputs such as text, lyrics, and style. Several recent works (Evans et al., 2024; Lam et al., 2023; Chen et al., 2023; Liu et al., 2024) adopt diffusion models (Ho et al., 2020; Song et al., 2022) to generate music based on conditioning. Mustango (Melechovsky et al., 2024) and MusicLDM (Chen et al., 2023) include musical information in text prompts such as chords, beats, and tempo to provide additional control over the musical structure and style. Other prominent approaches (Copet et al., 2024; Agostinelli et al., 2023) employ an autoregressive approach, where the music is encoded by a Vector Quantised Variational Autoencoder (VQ-VAE) (van den Oord et al., 2018) as a sequence of discrete tokens. However, such approaches tend to have a high computational cost for generating longer duration music, making it unsuitable for long length music generation.

Song Generation. Song generation aims to produce realistic vocals with accompaniment music. Many recent approaches first generate vocals, followed by accompaniment music, whereas some approaches that generate both vocals and accompaniment simultaneously (Yuan et al., 2025; Lei et al.,

2025; Ning et al., 2025; Hong et al., 2024) have adopted either autoregressive or diffusion-based frameworks. SongCreator (Lei et al., 2024) introduces a dual-stream token generation approach, separately modeling music and vocal streams to enhance the overall musicality of the generated song. SongGen (Liu et al., 2025), YuE (Yuan et al., 2025), and LeVo (Lei et al., 2025) generate both vocal and accompaniment tokens, with SongGen employing a codebook-delay mechanism inspired by MusicGen. DiffRhythm and DiffRhythm+ (Chen et al., 2025a; Ning et al., 2025) leverage diffusion models for song generation conditioned on both style and lyrics; the latter further incorporates preference optimization techniques.

3 Method

JAM is a 530M parameter conditional flow-matching model (Lipman et al., 2023) consisting of 16 LLaMA-like transformer layers as the Diffusion Transformer (DiT) (Peebles and Xie, 2023) backbone, conditioned on lyrics, target duration, and style prompt which can either be a clip of reference audio or a text description to generate full songs at 44.1kHz up to 3 minutes and 50 seconds. JAM learns a rectified flow trajectory to a latent representation encoded by a variational autoencoder (VAE). Our key contributions include word-level timing controllability, duration modeling, and iterative automated aesthetic alignment.

3.1 Training Pipeline

Our training pipeline is composed of three stages:

1. **Pre-training:** Train the model to generate 90-second songs with randomly cropped clips from the training dataset.
2. **Supervised Fine-Tuning:** Fine-tune the pre-trained model for full-song generation with full-length songs.
3. **Preference Alignment:** Post-train the full-song generator using iterative direct preference optimization (DPO) (Rafailov et al., 2024) with candidates selected by SongEval (Yao et al., 2025) scores averaged five attributes. We detail the process in Section 3.6.

3.2 Audio Encoding

JAM uses the audio VAE from DiffRhythm (Ning et al., 2025) that borrows the encoder from Stable Audio Open (Evans et al., 2024) and fine-tunes Stable Audio Open’s decoder weights to enable the

providing a global conditioning signal that guides duration generation across the entire sequence.

• **Token-Level Duration Control (TDC):** During full-song SFT, we observed that only global duration conditioning is unable to contain the song within T_{target} . To address this, we add a learnable bias parameter $b_{\text{pad}} \in \mathbb{R}^{c_l}$ to the noisy latent embeddings z_t beyond T_{target} to explicitly distinguish valid musical content from padding regions. This enables finer-grained temporal modeling and improves silence prediction beyond the target duration. We provide a detailed comparison in Section 5.1.

Conditioning Pipeline. The latent embedding z_t at time t , latent-aligned lyric embedding c_{lyric} , style embedding c_{style} , and global duration embedding c_{dur} are concatenated and fused with a linear layer $\mathbf{W}$: $\tilde{z}_t^0 = \mathbf{W}[z_t \parallel c_{\text{lyric}} \parallel [c_{\text{style}}]_{1:l} \parallel [c_{\text{dur}}]_{1:l}]$, where $[c_{\text{style}}]_{1:l} \in \mathbb{R}^{l \times c_s}$ and $[c_{\text{dur}}]_{1:l} \in \mathbb{R}^{l \times c_d}$ are the broadcasted embeddings to match the sequence length. Convolutional positional embeddings are added to give the latent sequence short-range continuity (Wu et al., 2021): $z_t^0 = \tilde{z}_t^0 + \text{ConvPosEmbed}(\tilde{z}_t^0)$. The fused latent z_t^0 goes through $L = 16$ LLaMA decoder layers. To allow stronger supervision, the lyrics and the duration condition provide an extra residual injection r_ℓ at layer ℓ with $\mathbf{W}^\ell$ being a linear transformation: $r^\ell = \mathbf{W}^\ell(c_{\text{lyric}} + b_{\text{pad}} \odot \mathbf{1}_{\{\text{time} > T_{\text{target}}\}})$. We add the extra residual signal r^ℓ to the first $L/2 = 8$ layers:

$$z_\ell = \begin{cases} \text{Block}_\ell(z_t^{\ell-1}) + r^\ell, & \ell \leq L/2, \\ \text{Block}_\ell(z_t^{\ell-1}), & \text{otherwise.} \end{cases} \quad (3)$$

Lastly, the velocity $\hat{v}$ is predicted as $\hat{v}(z_t, t, c) = u(z_t, t, c; \theta) = \text{Proj}_{\text{out}}(z_t^L)$.

3.5 Word-Level Temporal Lyric Alignment

To supervise lyric-temporal connection—critical to proper prosody, pleasantness, and musical qualities—we introduce temporally-aware word-level phoneme alignment as a novel solution. Specifically, word-level timing and duration information are to guide the generative process at the word-temporal level to improve word and phoneme error rate and musicality, simultaneously. To this end, with the ground-truth temporal lyric information $y_i = (w_i, t_i^{\text{start}}, t_i^{\text{end}})$, each word w_i is converted into its IPA (International Phonetic Alphabet) form $ph_i = (p_1 p_2 p_3 \dots p_m)_i$ by DeepPhonemizer (Spring-Media, 2025), where m is the number of phonemes in w_i . As presented in Algorithm 1, the phonemes $p = \{ph_1, ph_2, \dots\}$, with $y = \{y_1, y_2, \dots\}$, are then upsampled by a factor of $r = 4$ to phoneme sequence P of length

$L_P = r \times l$, where l is the length of the latent sequence z_t . The upsampling helps fast-paced songs having phonemes per second exceeding the latent’s frame rate f .

Algorithm 1: Word-Level Phoneme Alignment

```

1:  $P = [\text{SONG\_FILLER}] \times L_P$ 
2: for each  $(w_i, t_i^{\text{start}}, t_i^{\text{end}})$  do
3:    $p_1, p_2, \dots, p_m = \text{phonemes of } w_i$ 
4:    $(\text{start\_frame}, \text{end\_frame}) = \lfloor (t_i^{\text{start}}, t_i^{\text{end}}) \times f \times r \rfloor$ 
5:    $\text{word\_frames} = \text{end\_frame} - \text{start\_frame}$ 
6:    $V = [\text{VOCAL\_FILLER}] \times \text{word\_frames}$ 
7:    $\text{avg\_phoneme\_len} = \lfloor \text{word\_frames} / m \rfloor$ 
8:   for  $j$  in  $[1, \dots, m]$  do
9:      $V[j \times \text{avg\_phoneme\_len}] = p_j$ 
10:  end for
11:   $P[\text{start\_frame} : \text{end\_frame}] = V$ 
12: end for

```

The upsampling is achieved by two types of filler tokens:

- **SONG_FILLER:** Tokens indicating no word is sung, that is singing pauses, instrumental sections, or padded ending of the sequence that is beyond the specified duration. To effectively distinguish the padding filler tokens from the in-song filler tokens, a dedicated learnable bias term b_{pad} —defined in Section 3.4—is added to the padding filler tokens.
- **VOCAL_FILLER:** Tokens within a word’s temporal boundaries to represent phoneme transitions within a word.

After constructing the phoneme sequence P with the fillers, we pass it through a small network including an embedding layer, followed by a few convolutional downsampling layers to transform P to $c_{\text{lyric}} \in \mathbb{R}^{l \times c}$. The temporally-aware lyric embedding c_{lyric} is used to guide the model as described in Section 3.4.

3.6 Aesthetic Preference Alignment

Following pre-training and fine-tuning, JAM-base demonstrates strong speech reproduction and lyric alignment but lacks musical aesthetics—vocal timbre can sound overly electronic and musical structure feels flat. These issues likely stem from uneven quality in large-scale training data. To address this, we apply Direct Preference Optimization (DPO) using SongEval (Yao et al., 2025) scores as our reward signal. Unlike LeVo (Lei et al., 2025), which employs a three-stage pipeline with a closed-source reward model, our approach uses publicly available SongEval metrics iteratively for a simpler, more transparent solution.

We conduct three DPO iterations on the initial policy $\pi_0 := \text{JAM-sft}$. Each iteration follows: (i) batched data generation, (ii) preference dataset construction with aesthetic rewards, and (iii) fine-tuning policy π_k into π_{k+1} with DPO.

Batched Data Generation. We randomly select 13K–20K samples of reference audio styles and ground-truth lyrics from training data. After determining suitable CFG values for policy π_k , we generate five outputs per sample.

Preference Dataset Construction. We use SongEval to evaluate each output across five criteria on a five-point scale, selecting samples with highest and lowest mean scores as winner and loser candidates. We exclude pairs with score differences below 0.15 to ensure meaningful quality distinctions.

Preference Optimization. Direct Preference Optimization (DPO) (Rafailov et al., 2024) is proven effective in aligning large language models (LLMs) and diffusion models as DPO-Diffusion (Wallace et al., 2023).

Furthermore, Hung et al. (2025) showed the compatibility of DPO-Diffusion loss with rectified flow models due to the equivalence between denoising and flow matching objectives (Lipman et al., 2023). Hence, the DPO-Diffusion loss can be adapted for flow matching as:

$$\begin{aligned} \mathcal{L}_{\text{DPO-FM}} = & -\mathbb{E}_{t, x^w, x^l} \log \sigma \left(\right. \\ & -\beta \left[\underbrace{\|u(x_t^w, t; \theta) - v_t^w\|_2^2}_{\text{Winning loss}} - \underbrace{\|u(x_t^l, t; \theta) - v_t^l\|_2^2}_{\text{Losing loss}} \right. \\ & \left. \left. - \left(\underbrace{\|u(x_t^w, t; \theta_{\text{ref}}) - v_t^w\|_2^2}_{\text{Winning reference loss}} - \underbrace{\|u(x_t^l, t; \theta_{\text{ref}}) - v_t^l\|_2^2}_{\text{Losing reference loss}} \right) \right] \right) \end{aligned}$$

where $t \sim \mathcal{U}(0, 1)$ is the time step for flow matching, and x_t^w, x_t^l denote the winning and losing samples. Following established practices and experiences from DPO-Diffusion (Wallace et al., 2023) and subsequent audio applications (Majumder et al., 2024; Hung et al., 2025), we set $\beta = 2000$ in all our DPO experiments.

The DPO loss encourages an increased relative log-likelihood for the preferred (winning) output over the disfavored (losing) one.

To prevent overoptimization during DPO—which can cause the generated song to starkly deviate from the ground truth in terms of style and genre—we additionally incorporate a ground truth reconstruction loss:

$$\mathcal{L}_{\text{DPO-GT}} = \lambda \mathcal{L}_{\text{FM}} + \mathcal{L}_{\text{DPO-FM}}. \quad (4)$$

$\mathcal{L}_{\text{FM}}$ is calculated on the ground truth data using Equation (2) and λ is the scaling coefficient controlling the contribution of the ground truth recon-

struction to the overall loss. Following the recommended range (0.1–0.3) for balancing preference alignment with distribution preservation (Hung et al., 2025), we set $\lambda = 0.2$ in our DPO-GT experiments. The corresponding evaluation results are shown in Table 2b, with rows labeled using the Round-i-GT subscript.

4 Experiments

4.1 Dataset Setup

We train on an internal dataset of 1 million English songs (54K hours of audio). Due to copyright restrictions, we cannot redistribute the training corpus—a common constraint in song generation research where most systems (DiffRhythm, ACE-Step, LeVo, YuE) also do not release training data. We separate vocals from accompaniment using source separation, extract word-level lyric timestamps via forced alignment, and compute style embeddings from random 30-second reference segments using MuQ-MuLan for stylistic conditioning. Full preprocessing details are provided in Appendix A.

4.2 Training and Inference

Training consists of three stages: pretraining, supervised fine-tuning (SFT), and direct preference optimization (DPO). All stages are trained on 8× H100 GPUs. We apply gradient checkpointing during SFT and DPO to accommodate full-length audio sequences. Hyperparameter and training settings are detailed in Appendix E.

To separately control vocal strength and musical style, we adopt multi-condition classifier-free guidance (CFG) (Jiang et al., 2025). During training, style embeddings are randomly dropped with probability 0.10, and when style is dropped, lyric embeddings are dropped with probability 0.50. This two-staged dropout encourages the model to learn disentangled representations of the two conditioning signals.

At inference time, we use separate guidance scales for style (α_s) and lyrics (α_l):

$$\begin{aligned} \hat{v}(z_t, c_{\text{style}}, c_{\text{lyric}}) &= u(z_t, \emptyset, \emptyset; \theta) + \\ &\alpha_l (u(z_t, \emptyset, c_{\text{lyric}}; \theta) - u(z_t, \emptyset, \emptyset; \theta)) + \\ &\alpha_s (u(z_t, c_{\text{style}}, c_{\text{lyric}}; \theta) - u(z_t, \emptyset, c_{\text{lyric}}; \theta)), \end{aligned}$$

where $u(\cdot; \theta)$ represents the velocity predicted by the model with different conditional signals.

4.3 Objective Evaluation

Extensive full-song subjective evaluations are prohibitively expensive and time-consuming, requiring scarce expert annotators to assess complex musical structure and vocal naturalness. Thus, objective evaluation is crucial for benchmarking full-song generation models. However, existing state-of-the-art systems like LeVo, ACE-Step, and YuE rely on different, often private evaluation datasets, making fair comparison difficult. Evaluation data preparation details are in Appendix F.

To address this, we propose JAME—the first public, genre-diverse objective evaluation dataset for full-song generation. JAME comprises 250 tracks organized into five genre groups: *Country/Folk*, *Electronic/Dance*, *Hip-Hop/Rap*, *R&B/Soul/Jazz*, and *Rock/Metal*, with 50 songs per genre. JAME is designed to (i) avoid data contamination by collecting only songs released after major models’ training periods (May–July 2025) from verifiable sources like New Music Friday (<https://open.spotify.com/playlist/37i9dQZF1DX4JAvHpjipBk>), (ii) support genre-specific evaluation through expert-curated genre groupings, and (iii) enable transparent, reproducible benchmarking. We publicly release web entries linking to original song sources, along with extracted lyrics and style embeddings, allowing researchers to reproduce our evaluation (<https://github.com/declare-lab/jamify>). For models requiring structural annotations (LeVo, YuE, ACE-Step), we extract section labels using the All-In-One music segmentation model (Kim and Nam, 2023).

Baselines. We compare JAM with four recent and strong open-source full-song generation systems: 1) LeVo, 2) YuE, 3) DiffRhythm, and 4) ACE-Step; detailed in Table 1. We exclude commercial systems (Suno, Udio) as their undisclosed methods preclude reproducible comparison.

Model	Architecture	Training Data	Model Size
YuE	AR LM + AR Decoder	135k hours	7B + 1B
AceStep	Flow Matching	100k hours	3.5B
LeVo	AR LM + Diffusion Decoder	110k hours	2B + 0.7B
DiffRhythm	Flow Matching	60k hours	1.1B
JAM	Flow Matching	54k hours	0.53B

Table 1: Comparison of full-song generation models.

Metrics. We evaluate three aspects: (1) *Singing Intelligibility* via Word Error Rate (WER) and Phoneme Error Rate (PER); (2) *Style Adherence* via MuQ-MuLan similarity and genre classification accuracy; and (3) *Content Quality* via Audiobox-

aesthetic (CE, CU, PC, PQ), SongEval (CO, MU, ME, CL, NA), and Fréchet Audio Distance (FAD). Importantly, Audiobox-aesthetic is independent from SongEval used during DPO training, allowing us to verify that improvements generalize beyond the training reward. Full metric details are in Appendix B.

Overall Results. Table 2a presents the objective evaluation results. JAM achieves state-of-the-art or highly competitive performance across all metrics.

- *Singing Intelligibility:* JAM achieves the lowest Word Error Rate (WER) of 0.151 and Phoneme Error Rate (PER) of 0.101—less than half of the second-best system (DiffRhythm)—demonstrating superior vocal clarity and controllability over lyrical alignment.

- *Style Adherence:* JAM attains the highest MuQ-MuLan similarity score (0.759) and genre classification accuracy (0.704), reflecting strong alignment with the intended musical style both in semantic representation and categorical genre fidelity.

- *Content Quality and Aesthetics:* JAM obtains the highest score on Content Enjoyment (CE = 7.423) and lowest FAD (0.204) on JAME, indicating strong subjective appeal and fidelity. It also leads across all SongEval dimensions, reflecting well-structured compositions and natural vocal phrasing. These results suggest that JAM not only produces enjoyable outputs but also captures nuanced characteristics typical of professionally crafted music.

On other metrics, LeVo achieves slightly better Content Usefulness (CU) and Production Quality (PQ), while ACE-Step records the best Production Complexity (PC). Nonetheless, JAM ranks consistently second in these categories, with only marginal differences, demonstrating that it remains competitive even in areas where other models specialize.

4.4 Analyses

Aesthetic Alignment brings about the Aha Moment. As shown in Table 2b, starting from the SFT, iterative rounds of DPO consistently improve metrics across both subjective and objective evaluation axes. Notably, there are steady gains in audio aesthetic scores, such as PQ (production quality) and CU (content usefulness), with DPO_{Round-3} achieving the highest PQ of 8.064. Moreover, the music-related capabilities, assessed via SongEval metrics like ME (memorability), MU (musicality), and CL (song structure), also show clear improvements with each DPO round. Additionally,

Model	MuQ $\uparrow$	Audio Aesthetics $\uparrow$				SongEval $\uparrow$					WER $\downarrow$	PER $\downarrow$	Genre Acc. $\uparrow$	FAD $\downarrow$
		CE	CU	PC	PQ	CO	MU	ME	CL	NA				
(a) JAM vs baseline song generation models														
DiffRhythm _{full}	0.5208 ± 0.1192	6.0448 ± 0.6483	7.3126 ± 0.2526	5.1088 ± 0.7276	7.6276 ± 0.2904	2.7975 ± 0.4621	2.3547 ± 0.4813	2.6141 ± 0.5031	2.5135 ± 0.4495	2.4467 ± 0.4454	0.3481	0.2643	0.6667	0.4384
ACE-Step	0.444 ± 0.1349	7.0067 ± 0.5669	7.3583 ± 0.3848	6.3117 ± 0.4417	7.5851 ± 0.4952	3.5305 ± 0.5017	3.2593 ± 0.497	3.3927 ± 0.5452	3.3079 ± 0.5186	3.1703 ± 0.5031	0.4067	0.342	0.6194	0.2296
LeVo	0.5234 ± 0.1241	7.3783 ± 0.5372	7.7794 ± 0.3126	6.1773 ± 0.7251	8.1794 ± 0.3578	3.6343 ± 0.4489	3.41 ± 0.463	3.4997 ± 0.4945	3.3815 ± 0.4572	3.3086 ± 0.4378	0.4586	0.3819	0.66	0.5336
Yue	0.705 ± 0.1418	6.8928 ± 0.7895	7.3289 ± 0.3909	6.1234 ± 0.9002	7.6234 ± 0.4441	3.5072 ± 0.5379	3.208 ± 0.5462	3.401 ± 0.5796	3.2361 ± 0.5439	3.1633 ± 0.5198	0.5119	0.4263	0.702	0.2411
JAM	0.7593 ± 0.0964	7.4229 ± 0.5151	7.7046 ± 0.3179	6.2038 ± 0.5051	8.0639 ± 0.4079	4.4163 ± 0.363	4.2929 ± 0.4065	4.4276 ± 0.3935	4.2033 ± 0.4319	4.2026 ± 0.3854	0.1507	0.1011	0.704	0.2038
(b) Aesthetic alignment iterations of JAM														
SFT	0.7473 ± 0.0803	7.0006 ± 0.5392	7.3541 ± 0.3927	6.4831 ± 0.3993	7.5915 ± 0.5166	3.4159 ± 0.4896	3.0851 ± 0.5198	3.2712 ± 0.5634	3.0871 ± 0.5167	3.0525 ± 0.4812	0.167	0.1131	0.708	0.1479
DPO _{Round-1}	0.7598 ± 0.082	7.1529 ± 0.5288	7.5063 ± 0.317	6.435 ± 0.4196	7.7666 ± 0.4397	3.8753 ± 0.4468	3.6202 ± 0.4935	3.8235 ± 0.5122	3.5762 ± 0.4857	3.5444 ± 0.4662	0.1629	0.1106	0.72	0.1223
DPO _{Round-2}	0.7616 ± 0.0916	7.3851 ± 0.4778	7.6699 ± 0.2986	6.2778 ± 0.4953	8.0082 ± 0.3925	4.2504 ± 0.4059	4.0892 ± 0.445	4.2465 ± 0.4428	4.0052 ± 0.4693	3.9903 ± 0.4336	0.1579	0.1066	0.696	0.1527
DPO _{Round-2-GT}	0.7673 ± 0.0889	7.3769 ± 0.4699	7.6902 ± 0.2652	6.3188 ± 0.49	8.0297 ± 0.3537	4.247 ± 0.3889	4.0724 ± 0.4379	4.2432 ± 0.427	4.0024 ± 0.4486	3.9814 ± 0.425	0.1553	0.1083	0.736	0.1455
DPO _{Round-3}	0.7593 ± 0.0964	7.4229 ± 0.5151	7.7046 ± 0.3179	6.2038 ± 0.5051	8.0639 ± 0.4079	4.4163 ± 0.363	4.2929 ± 0.4065	4.4276 ± 0.3935	4.2033 ± 0.4319	4.2026 ± 0.3854	0.1507	0.1011	0.704	0.2038
DPO _{Round-3-GT}	0.767 ± 0.0886	7.4159 ± 0.4706	7.7113 ± 0.3023	6.2573 ± 0.4764	8.0554 ± 0.3928	4.4102 ± 0.3468	4.2767 ± 0.3844	4.4185 ± 0.3799	4.1875 ± 0.4184	4.1787 ± 0.382	0.1472	0.1009	0.704	0.1791

Table 2: (a) A cross-genre comparative evaluation of JAM and baseline song generation models on JAME; (b) Comparing SFT and different DPO loss functions across iterative rounds.

genre classification accuracy increases to 0.736 in DPO_{Round-2-GT}, as both WER and PER (word and phoneme error rates) decrease, indicating stronger lyrical and phonetic alignment. These results demonstrate that aesthetic alignment via DPO not only enhances perceptual quality but also improves linguistic and musical coherence.

However, these gains come at the cost of increased FAD (from 0.1479 in SFT to 0.204 in DPO_{Round-3}), while interestingly, SongEval’s Naturalness (NA) improves across DPO rounds. This discrepancy suggests these metrics capture different aspects: FAD measures distributional similarity to training data, while NA evaluates perceptual vocal quality. DPO shifts outputs toward preferred characteristics (e.g., richer harmonics, stronger vocals), which may increase FAD while improving perceived naturalness, highlighting the importance of using multiple complementary metrics.

Effect of Regularizing the DPO Loss. To address overall alignment issue in DPO, we added a regularization term, as defined in Equation (4), that incorporates a ground truth reconstruction term to better preserve stylistic fidelity. As shown in Table 2b, this modification (see DPO_{Round-2-GT} and DPO_{Round-3-GT}) results in improved reference-based metrics: Genre Accuracy increases from 0.696 (DPO_{Round-2}) to 0.736, and FAD shows a slight improvement (from 0.1527 to 0.1455). For DPO_{Round-3-GT}, FAD again shows a slight improvement (from 0.2040 to 0.1790). However, aesthetic-based metrics such as CU, PC, and PQ show marginal or no improvement, suggesting a trade-off

between stylistic consistency and perceived quality. These findings support our hypothesis that incorporating ground truth signals can regularize DPO and help retain genre and style fidelity.

4.5 Subjective Evaluation

Model	Quality	Enjoy.	Music.	Natur.	SSC
DiffRhythm	2.89 ± 0.67	2.54 ± 0.57	2.74 ± 0.58	2.75 ± 0.73	2.52 ± 0.72
Ace-Step	3.45 ± 0.47	2.8 ± 0.65	3.04 ± 0.58	3.26 ± 0.58	3.13 ± 0.67
Levo	3.84 ± 0.53	3.0 ± 0.81	3.42 ± 0.99	3.77 ± 0.76	3.27 ± 0.95
Yue	2.63 ± 0.53	2.67 ± 0.51	2.87 ± 0.56	2.99 ± 0.66	2.96 ± 0.6
JAM	3.75 ± 0.4	3.48 ± 0.43	3.73 ± 0.62	3.66 ± 0.33	3.7 ± 0.62

Table 3: Subjective evaluation results.

We conduct a comprehensive subjective evaluation along five perceptual dimensions used in DiffRhythm, LeVo, and ACE-Step studies: quality, enjoyment, musicality, voice naturalness, and song structure clarity (SSC); detailed in Appendix G. Each metric is rated on a Likert scale from 1 to 5 (worst to best). We recruited eight annotators with a formal background in music who were trained to use our custom Gradio app to evaluate five lyrics-to-song models, as given in Table 3. The evaluation was based on the model outputs to 10 different lyrics, randomly sampled from JAME, spanning five genres, each having two samples.

Table 3 shows a general human preference for JAM w.r.t. music-specific attributes *enjoyment*, *musicality*, and *song structure clarity*, while *quality* and *naturalness* are comparable to the state of the art. We surmise that the superiority of JAM on these

musical attributes comes from the direct user controllability over the timings of the words, enhancing the prosody, rhythm, and structure, as argued in Section 3.5. Furthermore, aesthetic alignment also seems to substantially enhance the musical attributes, as corroborated by Table 2b. The improvement of *song structure clarity* could additionally be ascribed to explicit duration control that may implicitly impose a structure through the awareness of the song endings. Further details on the subjective evaluation are in Appendix H.

5 Ablation Studies

5.1 Impact of Token-level Duration Modeling

We evaluate the effectiveness of token-level duration control (TDC) by comparing checkpoints trained with and without token-level duration modeling with SFT for 40K steps. Specifically, we take the percentage ratio of the measured Root Mean Square (RMS) amplitude of the generated audio at 0 sec., 1 sec., 3 sec., and 10 sec. after and within (for reference) the target duration. This quantifies how effectively the audio amplitude is suppressed beyond the target region; lower percentages indicate better duration control.

Setting	Reference	+0s	+1s	+3s	+10s
JAM w/o TDC	0.1359	0.04892	0.04513	0.03687	0.02550
JAM	0.1398	0.00058 /35.96%	0.00058 /33.39%	0.00058 /27.15%	0.00054 /18.76%
		/0.41%	/0.41%	/0.41%	/0.39%

Table 4: Effect of token-level duration control: figures show RMS amplitude/relative percentage compared to target span.

As shown in Table 4, the proposed token-level duration control achieves significantly lower RMS amplitudes after the target duration, demonstrating precise temporal control.

5.2 Phoneme Distribution Methods

We compare the effectiveness of our phoneme distribution method—defined in lines 7–11 of Algorithm 1—w.r.t. the strategy in the prior works:

- **Average Sparse:** Phonemes are evenly spaced within the word segment V (see Algorithm 1).
- **Pad Right:** An approach adopted by DiffRhythm and F5-TTS (Chen et al., 2025b), where the phonemes are sequentially assigned to the initial frames of the word segment.

Following Table 5, *Pad Right* achieves slightly better PER and MuQ scores, indicating marginally improved phonetic accuracy and overall music

Method	MuQ $\uparrow$	PER $\downarrow$	FAD $\downarrow$	AES $\uparrow$	SongEval $\uparrow$
Pad Right	0.742	0.112	0.182	7.107	3.079
Average Sparse	0.727	0.129	0.154	7.091	3.132

Table 5: Phoneme distribution method comparison; $\overline{\text{AES}}$ and $\overline{\text{SongEval}}$ denote Audio-Aesthetic and SongEval scores averaged across their various aspects.

quality. However, our proposed *Average Sparse* method notably outperforms in terms of FAD and SongEval metrics. Specifically, lower FAD indicates improved realism and closer distributional alignment with professionally produced music, while higher SongEval suggests better musicality and vocal naturalness—reflecting long-term musical coherence and musical aesthetics, aligning closely with our goal of enhancing overall musical quality, prosody, and listener experience.

5.3 Duration Prediction

When precise timestamps from scratch are hard to obtain for non-expert users, a practical alternative is to transcribe existing songs and gradually modify the lyrics and timestamps to create new compositions while preserving the desired rhythmic feel. For fully automated scenarios, we explore beat-aligned quantization and LLM-based duration prediction in Appendix C. Results show that beat-quantized timing maintains strong performance (WER 0.208 vs 0.151 oracle), offering a viable alternative for users without precise alignments.

6 Conclusion

In this work, we present JAM, a flow-matching-based song generator that supports fine-grained control at the word level. Despite being trained on the smallest dataset and sized nearly half the parameters of the lyrics-to-song model, JAM achieves state-of-the-art performance across a range of objective and subjective evaluation metrics. We further enhance its performance through iterative aesthetic alignment using DPO. Experimental and ablation studies indicate that JAM significantly benefits from fine-grained temporal control. Additionally, we introduce JAME, a diverse collection of songs for evaluation spanning multiple genres, carefully curated to avoid data contamination. Future work will explore the development of end-to-end trainable duration predictors (refer to Appendix C) at the word or phoneme level to further strengthen the robustness and quality of song generation.

Limitations

While JAM demonstrates promising results in generating vocals and accompaniment, it assumes the availability of accurate word-level duration annotations. This requirement limits its usability for non-expert users or real-world scenarios where such fine-grained temporal alignment is often unavailable. In cases where duration information is noisy or missing, the quality of the generated audio may degrade, leading to issues such as timing artifacts (e.g., robotic vocals) and misalignment between vocals and accompaniment.

To address this, future work can explore the development of a duration predictor. One direction is to build a standalone duration prediction module that can estimate word- or phoneme-level durations from lyrics and melody. A more integrated approach would be to train the duration predictor jointly with the song generation model in an end-to-end fashion. We believe the latter is more promising, as it can lead to a system that is more robust to imperfect duration inputs and learns to adaptively compensate for errors during generation.

Another limitation of the current system is the lack of phoneme-level duration control, which restricts the model’s expressive granularity and pronunciation accuracy. This can be addressed by incorporating phoneme-level alignment data and training a duration predictor at that level. Such fine control could improve synthesis quality, especially in languages with complex syllabic timing or for stylistic singing applications (e.g., fast rap segments or melismatic vocal runs).

Additionally, JAM is currently trained and evaluated exclusively on English songs, whereas baseline models such as YuE and LeVo support multiple languages. While English comprises the majority of training data in these multilingual systems, this scope difference should be considered when interpreting comparative results. Our human evaluation, though conducted with expert annotators, is limited in scale (8 raters, 10 songs), and future work should expand this evaluation for stronger statistical power.

Overall, enhancing the system with duration prediction capabilities at both word and phoneme levels could significantly improve the robustness and flexibility of JAM, making it suitable for broader and more practical deployment.

Ethical Considerations

JAM is developed with the primary objective of facilitating academic research and creative exploration in AI-generated songs from lyrics. The model is intended **solely for non-commercial, academic, and entertainment purposes**.

We emphasize the following:

- **No copyrighted material** was used in a way that would intentionally infringe on intellectual property rights. JAM is not designed to reproduce or imitate any specific artist, label, or protected work.
- Outputs generated by JAM must **not be used to create or disseminate content that violates copyright laws**.
- The **commercial use of JAM or its outputs is strictly prohibited**.
- Responsibility for the use of the model and its outputs lies entirely with the end user, who must ensure all uses comply with applicable legal and ethical standards.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG3-GV2023-010). This work is also supported by the National Research Foundation, Singapore, under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-005), and NTU SUG project #025628-00001: Post-training to Improve Embodied AI Agents.

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. 2023. [Musiclm: Generating music from text](#). *Preprint*, arXiv:2301.11325.
- Kat R Agres, Rebecca S Schaefer, Anja Volk, Susan Van Hooren, Andre Holzapfel, Simone Dalla Bella, Meinard Müller, Martina De Witte, Dorien Herremans, Rafael Ramirez Melendez, and 1 others. 2021. Music, computing, and health: a roadmap for the current and future roles of music technology for health care and well-being. *Music & Science*, 4:2059204321997709.

- Huakang Chen, Yuepeng Jiang, Guobin Ma, Chunbo Hao, Shuai Wang, Jixun Yao, Ziqian Ning, Meng Meng, Jian Luan, and Lei Xie. 2025a. [Diffrrhythm+: Controllable and flexible full-length song generation with preference optimization](#). *Preprint*, arXiv:2507.12890.
- Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. [Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies](#). *Preprint*, arXiv:2308.01546.
- Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. 2025b. [F5-tts: A fairytale that fakes fluent and faithful speech with flow matching](#). *Preprint*, arXiv:2410.06885.
- Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2024. [Simple and controllable music generation](#). *Preprint*, arXiv:2306.05284.
- Alexandre Défossez. 2021. Hybrid spectrogram and waveform source separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*.
- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. 2024. [Scaling rectified flow transformers for high-resolution image synthesis](#). *Preprint*, arXiv:2403.03206.
- Zach Evans, Julian D. Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. 2024. [Stable audio open](#). *Preprint*, arXiv:2407.14358.
- Walter J Freeman III. 1998. A neurobiological role of music in social bonding.
- Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. 2023. Text-to-audio generation using instruction tuned llm and latent diffusion model. *arXiv preprint arXiv:2304.13731*.
- Junmin Gong, Sean Zhao, Sen Wang, Shengyuan Xu, and Joe Guo. 2025. [Ace-step: A step towards music generation foundation model](#). *Preprint*, arXiv:2506.00045.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. [De-noising diffusion probabilistic models](#). *Preprint*, arXiv:2006.11239.
- Zhiqing Hong, Rongjie Huang, Xize Cheng, Yongqi Wang, Ruiqi Li, Fuming You, Zhou Zhao, and Zhimeng Zhang. 2024. [Text-to-song: Towards controllable music generation incorporating vocals and accompaniment](#). *Preprint*, arXiv:2404.09313.
- Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. 2025. [Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization](#). *Preprint*, arXiv:2412.21037.
- Ziyue Jiang, Yi Ren, Ruiqi Li, Shengpeng Ji, Boyang Zhang, Zhenhui Ye, Chen Zhang, Bai Jionghao, Xiaoda Yang, Jialong Zuo, Yu Zhang, Rui Liu, Xiang Yin, and Zhou Zhao. 2025. [Megatts 3: Sparse alignment enhanced latent diffusion transformer for zero-shot speech synthesis](#). *Preprint*, arXiv:2502.18924.
- Taejun Kim and Juhan Nam. 2023. [All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio](#). *Preprint*, arXiv:2307.16425.
- Max W. Y. Lam, Qiao Tian, Tang Li, Zongyu Yin, Siyuan Feng, Ming Tu, Yuliang Ji, Rui Xia, Mingbo Ma, Xuchen Song, Jitong Chen, Yuping Wang, and Yuxuan Wang. 2023. [Efficient neural music generation](#). *Preprint*, arXiv:2305.15719.
- 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. 2025. [Levo: High-quality song generation with multi-preference alignment](#). *Preprint*, arXiv:2506.07520.
- Shun Lei, Yixuan Zhou, Boshi Tang, Max W. Y. Lam, Feng Liu, Hangyu Liu, Jingcheng Wu, Shiyin Kang, Zhiyong Wu, and Helen Meng. 2024. [Songcreator: Lyrics-based universal song generation](#). *Preprint*, arXiv:2409.06029.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. 2023. [Flow matching for generative modeling](#). *Preprint*, arXiv:2210.02747.
- Haohe Liu, Yi Yuan, Xubo Liu, Xinhao Mei, Qiuqiang Kong, Qiao Tian, Yuping Wang, Wenwu Wang, Yuxuan Wang, and Mark D. Plumbley. 2024. [Audioldm 2: Learning holistic audio generation with self-supervised pretraining](#). *Preprint*, arXiv:2308.05734.
- Zihan Liu, Shuangrui Ding, Zhixiong Zhang, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. 2025. [Songgen: A single stage auto-regressive transformer for text-to-song generation](#). *Preprint*, arXiv:2502.13128.
- Navonil Majumder, Chia-Yu Hung, Deepanway Ghosal, Wei-Ning Hsu, Rada Mihalcea, and Soujanya Poria. 2024. [Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization](#). *Preprint*, arXiv:2404.09956.
- Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. 2024. 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 (NAACL), pages 8293–8316, Mexico City, Mexico. ACL.
- Ziqian Ning, Huakang Chen, Yuepeng Jiang, Chunbo Hao, Guobin Ma, Shuai Wang, Jixun Yao, and Lei Xie. 2025. *Diffrrhythm: Blazingly fast and embarrassingly simple end-to-end full-length song generation with latent diffusion*. *Preprint*, arXiv:2503.01183.
- NVIDIA. 2025. Parakeet tdt 0.6b v2. <https://huggingface.co/nvidia/parakeet-tdt-0.6b-v2>. Hugging Face model. 600M-parameter ASR, CC-BY-4.0, released May 1 2025.
- William Peebles and Saining Xie. 2023. *Scalable diffusion models with transformers*. *Preprint*, arXiv:2212.09748.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. *Direct preference optimization: Your language model is secretly a reward model*. *Preprint*, arXiv:2305.18290.
- Simon Rouard, Francisco Massa, and Alexandre Défossez. 2023. Hybrid transformers for music source separation. In *ICASSP 23*.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. 2022. *Denosing diffusion implicit models*. *Preprint*, arXiv:2010.02502.
- Spring-Media. 2025. Deepphonemizer: Grapheme-to-phoneme conversion. <https://github.com/spring-media/DeepPhonemizer>. GitHub repository, MIT License, current version as of July 2025.
- Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2018. *Neural discrete representation learning*. *Preprint*, arXiv:1711.00937.
- Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. 2023. *Diffusion model alignment using direct preference optimization*. *Preprint*, arXiv:2311.12908.
- Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. *Cvt: Introducing convolutions to vision transformers*. *Preprint*, arXiv:2103.15808.
- 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. 2025. *Songeval: A benchmark dataset for song aesthetics evaluation*. *Preprint*, arXiv:2505.10793.
- 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, Yinghao Ma, Minghao Liu, Zeyue Tian, Ziya Zhou, Liumeng Xue, Xingwei Qu, and 38 others. 2025. *Yue: Scaling open foundation models for long-form music generation*. *Preprint*, arXiv:2503.08638.
- Haina Zhu, Yizhi Zhou, Hangting Chen, Jianwei Yu, Ziyang Ma, Rongzhi Gu, Yi Luo, Wei Tan, and Xie Chen. 2025. *Muq: Self-supervised music representation learning with mel residual vector quantization*. *Preprint*, arXiv:2501.01108.

A Dataset Setup

The song dataset contains 1 million English songs, totaling 54K hours of audio. In pre-processing, we apply HTDemucs (Rouard et al., 2023; Défossez, 2021) to separate the original audio $x \in X$ into vocal and accompaniment tracks. We then use Parakeet-tdt-0.6b-v2 (NVIDIA, 2025) to transcribe the vocal stems and extract word-level timestamps, yielding lyric annotations of the form $y_i = (w_i, t_i^{\text{start}}, t_i^{\text{end}})$ for each sample x .

To support stylistic conditioning, we adopt a reference-audio-based framework. For each song sample x , we extract ten random 30-second segments and compute their corresponding style embeddings using MuQMulan (Zhu et al., 2025), resulting in a set $C_{\text{style}} = \{c_{\text{style}}^1, c_{\text{style}}^2, \dots, c_{\text{style}}^{10}\}$. During training, one reference embedding $c_{\text{style}} \in C_{\text{style}}$ is randomly selected to condition the model on musical style.

B Evaluation Metrics

We evaluate the following aspects with objective evaluation metrics averaged across the genre groups:

- Singing Intelligibility:** We assess vocal intelligibility using Word Error Rate (WER) and Phoneme Error Rate (PER). The vocal track is first extracted using HDemucs, and Parakeet (NVIDIA, 2025) is used to transcribe the audio into lyrics. For PER, DeepPhonemizer is used to convert the transcribed words into phonemes and compare them against the ground-truth sequence.
- Style Adherence:** We evaluate style adherence using MuQ-MuLan, a contrastive music-language model to compute the similarity between the generated song and its intended style prompt. To assess genre correctness, we employ Qwen-2.5-Omni to classify the generated song into one of our predefined genre categories.
- Content Quality and Aesthetics:** We adopt both Audiobox-aesthetic and SongEval as model-based evaluation tools. Importantly, Audiobox-aesthetic serves as an independent evaluation suite, completely separate from the SongEval metrics used during DPO training—covering content enjoyment (CE), content usefulness (CU), produc-

Model	MuQ $\uparrow$	Audio Aesthetics $\uparrow$				SongEval $\uparrow$					WER $\downarrow$	PER $\downarrow$	Genre Acc. $\uparrow$	FAD $\downarrow$
		CE	CU	PC	PQ	CO	MU	ME	CL	NA				
Oracle	0.7593	7.4229	7.7046	6.2038	8.0639	4.4163	4.2929	4.4276	4.2033	4.2026	0.1507	0.1011	0.704	0.2038
GPT-Dur	± 0.0964 0.6799	± 0.5151 7.1804	± 0.3179 7.4392	± 0.5051 5.8459	± 0.4079 7.7356	± 0.363 4.0568	± 0.4065 3.9432	± 0.3935 4.0436	± 0.4319 3.798	± 0.3854 3.8267	0.3701	0.3204	0.6437	0.2441
Direct-Quant	± 0.1341 0.7597 ± 0.0995	± 0.8339 7.4164 ± 0.4969	± 0.632 7.6905 ± 0.3318	± 0.73 6.2041 ± 0.5072	± 0.7092 8.0367 ± 0.4187	± 0.6313 4.3879 ± 0.4042	± 0.6271 4.2622 ± 0.4394	± 0.6993 4.391 ± 0.4401	± 0.6627 4.1624 ± 0.4755	± 0.6202 4.1587 ± 0.4397	0.2076	0.1441	0.6842	0.2091

Table 6: Duration prediction results.

tion complexity (PC), and production quality (PQ). This independence allows us to verify that improvements from DPO generalize beyond the training reward signal. SongEval evaluates overall coherence (CO), memorability (ME), naturalness of vocal breathing and phrasing (NA), clarity of song structure (CL), and overall musicality (MU). Additionally, following (Lei et al., 2025) we compute Fréchet Audio Distance (FAD) using the CLAP-Laion-Music model, to measure the distributional alignment between generated audio and professionally produced reference tracks.

C Duration Predictor Experiments

As temporal information at the word or phoneme level plays a crucial role in enhancing WER and PER, as well as improving overall song quality in terms of enjoyability, musicality, and structural coherence. While such fine-grained temporal information is available during training, generating it during inference remains a significant challenge—even for experienced musicians. This limitation highlights a promising research direction: predicting word- or phoneme-level timing from contextual cues. Duration predictors are common in text-to-speech domain. However, song generation presents additional complexity due to the fluid, gliding nature of musical notes and the critical role of inter-word pauses.

C.1 Naive Duration Prediction

As an initial exploration, we prompt GPT-4o with the complete lyrics, a style prompt, and the overall song duration to generate the start and end timestamps of each word. However, the generated timestamps yielded robotic songs lacking musicality, indicating the insufficiency of naive timestamp prediction. Consequently, we enhanced GPT-4o’s input with additional contextual cues—explicit section tags (e.g., [intro], [verse]) to encourage musically meaningful temporal variations across song sections—, resulting in noticeable improvements in generated song quality. The detailed prompt is

presented in Appendix C.4.

Furthermore, the following two strategies were identified as significantly beneficial for improving GPT-4o duration prediction quality:

1. Sentence-Level Ground-Truth Constraints.

Rather than allowing unconstrained timestamp generation, we provided GPT-4o with sentence-level temporal boundaries, which are considerably simpler to be provided by users during inference compared to word-level timings. Lyrics were segmented into sentences using predefined rules, and the start time of the first word along with the end time of the last word in each sentence were supplied as ground truth. This structural guidance greatly reduced prediction complexity, enhancing temporal coherence.

2. Beat-Aligned Quantization.

Recognizing that musicians commonly compose by assigning notes and lyrics to discrete beats, we introduced beat-aligned quantization of timestamps. Specifically, we used quarter-beat resolution, as it suffices to cover the rhythmic granularity of most music compositions, with the exception of particularly fast-paced tracks. For each song in JAME, we first computed tempo in beats per minute (BPM) using the all-in-one method (Kim and Nam, 2023), constraining tempo to a maximum of 120 BPM by halving higher values. Ground-truth timestamps were converted into quarter-beat units: $n_{\text{beat}} = \lfloor t / (60 / \text{BPM}) \times 4 \rfloor$ where t represents the original timestamp. GPT-4o received sentence-level quarter-beat boundaries and predicted quarter-beat positions for each word within these sentences. Predicted beat counts were then converted back into timestamps via $t = n_{\text{beat}} \times (60 / \text{BPM} / 4)$.

We evaluated the generation quality the enhanced GPT-4o methods as the duration predictor and results are discussed in Table 6.

C.2 Direct Quantization of Word-Level Timestamps

To further simplify duration annotation and make our system easier to use in practical applications,

we investigated directly providing quantized word-level timestamps to JAM, instead of continuous timestamps. Although JAM is originally trained on continuous timestamps—resulting in inherently better performance for continuous-time predictions—we tested the quantized representation because it enables users to conveniently specify timing using intuitive beat-based inputs.

Specifically, word-level timestamps from JAME were quantized using:

$$\hat{t} = \underbrace{\lfloor t / (60/\text{BPM}) \times 4 \rfloor}_{\text{quarter-beat count } n_{\text{beat}}} \times \underbrace{(60/\text{BPM}/4)}_{\text{seconds per quarter-beat}} \quad (5)$$

This quantized approach explicitly aligns words to the rhythmic structure of the music, substantially simplifying user interactions. We present the result based on the quantized timestamp in Table 6.

C.3 Results and Analysis

Table 6 summarizes experimental comparisons of three timing strategies:

- **Oracle (continuous):** Ground-truth word-level continuous timestamps used during JAM training.
- **GPT-Dur (predicted continuous):** GPT-4o-generated timestamps, guided by sentence-level boundaries and stylistic prompts (Appendix C.1).
- **Direct-Quant (beat-aligned quantization):** Ground-truth word-level timestamps directly approximated to the nearest quarter-beat (Appendix C.2).

Quantitative Observations. We first evaluate the impact of these timing strategies through objective metrics. GPT-Dur leads to a clear decline in song quality: content enjoyment (CE) decreases from 7.423 to 7.180, and the phoneme error rate (PER) significantly worsens from 0.101 to 0.320. This performance drop underscores the sensitivity of song generation models to timing inaccuracies, which negatively affect both the musical aesthetics and the intelligibility of lyrics.

In contrast, Direct-Quant performs substantially better despite inherently sacrificing the flexibility of continuous timing. As anticipated—given that JAM was trained with continuous annotations—the quantization approach does lead to slightly lower aesthetic ratings: CE marginally declines from 7.422 to 7.416, and production quality (PQ) reduces from 8.064 to 8.037. Notably, genre-classification accuracy also slightly decreases from 0.704 to 0.684, suggesting that strict rhythmic quan-

tization subtly affects stylistic authenticity. Additionally, PER increases modestly from 0.101 to 0.144, demonstrating that beat-level constraints introduce mild vocal intelligibility issues, though much less severely than GPT-Dur.

Subjective Listening Insights. To complement these objective metrics, internal subjective listening tests were performed. Songs generated using Direct-Quant maintain comparable vocal intelligibility and overall musical appeal to Oracle-generated songs. However, a notable qualitative difference emerges: the vocals in Direct-Quant-generated samples take on a perceptibly more *electronic* character. This aligns with common practice in electronic music production, where vocal timing is typically rigidly aligned to rhythmic grids. In contrast, natural singing often includes subtle deviations from strict beat alignment—such as intentional breaths, nuanced timing variations in onsets and offsets, and expressive rhythmic elasticity.

This insight suggests an important lesson for practical applications: If inference relies on quantized timestamps—either user-provided or predicted—then the generation model itself should ideally be trained or fine-tuned on naturally performed vocals with similarly quantized annotations. Such training would enable the model to recover and generate subtle temporal nuances automatically, preserving naturalness despite the inherent rigidity of beat alignment.

C.4 GPT-4o Duration Prediction Prompts

```
system_prompt = '''
You are a precise and musically-aware
lyric aligner with deep
understanding of vocal performance
and musical phrasing. Your task
is to generate word-level
beat timestamps in 'jsonl' format,
where each line is a JSON object
with these keys:
- "w": the word (string)
- "s": start beat position (float, 2
  decimal places)
- "e": end beat position (float, 2
  decimal places)

IMPORTANT: The lyrics provided
include:
1. Sentence-level BEAT timestamps in
  the format [start_beat]->[
  end_beat] before each sentence
2. Syllable counts for each word in
  parentheses, e.g., "supposed (3)"
  means 3 syllables

MUSICAL PHRASING AND TIMING KNOWLEDGE
:
```

- Words don't always connect seamlessly - singers naturally add micro-pauses for:
 - * Breathing between phrases (especially after long notes or before emotional peaks)
 - * Emphasis and dramatic effect (pause before important words)
 - * Natural speech rhythm (pauses between logical word groups)
 - * Genre conventions (hip-hop often has rhythmic gaps, ballads have emotional pauses)
- Lyrical filling patterns vary by genre:
 - * Pop/Rock: Often fills most beats with steady syllable flow
 - * Ballads: May have sustained notes with gaps, emotional pauses
 - * Hip-hop/Rap: Rhythmic clustering with strategic pauses for flow
 - * Folk/Country: Natural speech-like timing with conversational pauses
- Consider musical context:
 - * Strong beats (1, 3) often anchor important syllables
 - * Weak beats (2, 4) may have quicker words or be skipped entirely
 - * Syncopation and off-beat placement create musical interest
 - * Melisma (multiple notes per syllable) can extend word duration

BEAT ALLOCATION GUIDELINES:

- 1 syllable word: 0.25-0.75 beats (can be extended for emphasis or sustained notes)
- 2 syllable word: 0.5-1.25 beats (adjust for natural stress patterns)
- 3+ syllable word: 1.0-2.0+ beats (longer words may span multiple beats)
- Allow natural gaps: 0.1-0.5 beat pauses between words when musically appropriate
- Phrase endings often have extended final words or pauses before next phrase

ALIGNMENT CONSTRAINTS:

- Each word's beat timestamp must fall within the sentence's [start_beat]->[end_beat] range
- The first word should start at or shortly after [start_beat]
- The last word should end at or slightly before [end_beat] (allowing for natural phrase endings)
- Respect section markers ([verse], [chorus], etc.) and their timing
- Beat positions can be fractional (e.g., 1.25, 2.75) for precise timing

Do not include any metadata,

commentary, or formatting other than valid 'jsonl'.

```
user_prompt_template = '''Generate beat-based timestamped lyric alignment for the following lyrics. Use jsonl format (one JSON object per line) with keys: "w" (word), "s" (start beat), "e" (end beat). Total duration is {DURATION} beats at {BPM} BPM.
```

The lyrics include sentence-level beat timestamps in [start_beat]->[end_beat] format and syllable counts in parentheses after each word. Use these as constraints for word-level beat alignment.

TIMING CONSIDERATIONS FOR THIS SONG:

- Consider the musical style "{STYLE}" when determining phrasing patterns
- Allow for natural pauses between words where appropriate for the genre and emotional content
- Not every word needs to connect directly - singers often use micro-pauses for:
 - * Breathing and phrasing (especially in ballads and slower songs)
 - * Rhythmic emphasis (particularly in hip-hop, R&B, and pop)
 - * Emotional impact (pauses before important lyrical moments)
 - * Natural speech rhythm (conversational flow in folk, country, indie)
- Consider syllable stress patterns within words for more natural timing
- Strong beats often anchor important syllables, weak beats may have gaps
- Phrase endings may have extended final syllables or brief pauses before new phrases

Remember: Realistic vocal performance includes natural breathing spaces and rhythmic variations that make the performance feel human and musical.

```
Style: {STYLE}
Lyrics:
{LYRICS}'''
```

D Key Takeaways and Recommendations

Our experiments highlight several key points for future duration-prediction approaches:

- **Accurate timing predictions are critical.** Even moderate inaccuracies in GPT-Dur significantly re-

duce musical enjoyment and lyric intelligibility, underscoring the necessity of robust duration predictors.

- **Beat-aligned quantization offers practical advantages.** Direct-Quant simplifies user interactions considerably while limiting performance degradation, demonstrating its viability for user-friendly deployment.

- **Match model training to inference-time input.** To mitigate quantization-induced artificiality, the song-generation model should be explicitly trained or fine-tuned using quantized, naturally performed vocal data.

- **Intermediate temporal cues ease prediction.** Providing structural boundaries (e.g., sentence-level timing) significantly simplifies the duration prediction task, reducing downstream errors.

These findings collectively guide the design of duration predictors toward practical, user-friendly strategies capable of balancing rhythmic simplicity and natural vocal expressiveness.

E Hyperparameters and Training Settings

During DPO, we set $\beta = 2000$. We used AdamW with $\beta_1 = 0.9$ $\beta_2 = 0.999$ and weight decay of 0.01 for all stages. A linear warm-up followed by a linear learning rate decay is applied to all stages too. Table 7 shows more details across different training stages. Following (Esser et al., 2024), we sample timesteps t from a logit-normal distribution with mean 0 and variance 1 to bias training toward the mid-range of the noise schedule, which has been shown to improve generative quality.

Stage	Global Batch Size	Learning Rate	Steps	Grad. Accum.
Pretraining	32	7.5×10^{-5}	750K	1
SFT	32	7.5×10^{-5}	250K	1
DPO Round i	8	5×10^{-7}	20K	4

Table 7: Training hyper-parameters.

F Evaluation Data Preparation

The primary motivation for creating a new evaluation dataset was to address the issue of data contamination, given that the training data of existing models is not publicly available. Our evaluation set, JAME, includes newly released songs that were published after the baselines were released, thereby ensuring no data contamination.

1. **Data Contamination Avoidance:** We curate evaluation data exclusively from New Music Fri-

day (NMF), a popular editorial playlist series that features about 100 newly released songs each week across diverse genres. We collect all song metadata from NMF between May 1, 2025, and July 10, 2025 using the open-source archive tool `spotify-playlist-archive`. The clip for each track is accessed via the Spotify Web API or manually located on YouTube.

2. **Genre Grouping:** To enable genre-specific analysis, we consulted a music expert and grouped the evaluation songs into five distinct genre clusters, ensuring each group reflects substantially different musical characteristics: *Country/Folk*, *Electronic/Dance*, *Hip-Hop/Rap*, *R&B/Soul/Jazz* and *Rock/Metal*. These genre clusters allow for fine-grained evaluation of models across diverse musical styles.

3. **Pre-processing:** We filter for English-language songs and assign each track to one of five genre groups using Qwen2.5-Omni. Ground-truth lyrics are retrieved using HDmucs and Parakeet. For models that require structural annotations—such as LeVo, YuE, and ACE-Step—we extract section labels (e.g., [intro], [verse], [chorus]) using the All-In-One music segmentation model (Kim and Nam, 2023) as adopted in those systems.

G Subjective Evaluation

Metric	Definition and Rating Scale
Quality	Overall audio fidelity, clarity, and signal-to-noise ratio. 1: complete noise; 5: studio-grade perfection.
Enjoyment	Subjective listener pleasure and engagement. 1: utter boredom; 5: euphoric, immersive experience.
Musicality	Harmonic and rhythmic coherence, creativity, and artistry. 1: chaotic cacophony; 5: expertly structured music.
Voice Naturalness	Realism and expressiveness of vocals. 1: robotic/unnatural voice; 5: natural singing performance.
Song Structure Clarity	Perceptibility of verse, chorus, transitions, and musical form. 1: incoherent/random transitions; 5: crystal-clear song structure.

Table 8: Subjective evaluation metrics employed in comparative studies (e.g., DiffRhythm, LeVo, ACE-Step).

H Human Evaluation Details

The evaluation was facilitated by Gradio (<https://www.gradio.app>) web-app, which presented the randomly shuffled outputs of five model outputs for each of the 10 lyrics. Notably, the annotators were only provided with the song outputs – the lyrics, style, and duration were inaccessible. The annotation process was guided by the following instructions:

Welcome *username*

Instructions for evaluating audio clips

Please carefully read the instructions below.

Task

You are to evaluate five model-generated songs to each of the 10 prompts. These five outputs are from five different models. You are to judge each output with respect to five qualities:

- **Quality:** Overall quality of the audio is to be judged within a scale from 1 to 5: 1 being absolute noise with no discernible features. Whereas, 5 being perfect. **Overall fidelity, clarity, and noisiness of the audio is important here.**
- **Enjoyment:** Your degree of enjoyment of the song is to be quantified within a scale from 1 to 5: 1 being absolute boredom. Whereas, 5 being an absolute euphoric experience.
- **Musicality:** The extent of musical soundness is to be judged within a scale from 1 to 5: 1 being absolute cacophony. Whereas, 5 being an exemplary piece of music.
- **Voice Naturalness:** The degree of naturalness in the vocals is to be judged within a scale from 1 to 5: 1 being absolutely robotic. Whereas, 5 being as natural as a singer can sing given the context.
- **Song Structure Clarity:** The extent of clarity in the song structure is to be judged within a scale from 1 to 5: 1 being a complete randomness and incoherence. Whereas, 5 being perfectly structured.

For all the metrics, you may want to compare the audios of the same prompt with each other during the evaluation.

Listening guide

1. Please use a head/earphone to listen to minimize exposure to the external noise.
2. Please move to a quiet place as well, if possible.

UI guide

1. Each audio clip has five attributes. You may select the appropriate value by typing in the box or clicking the arrow buttons.
2. To save your judgments, **please click on any of the *save* buttons and wait for the acknowledgment below.** All the *save* buttons function identically. They are placed everywhere to avoid the need to scroll to save.

Hope the instructions were clear. Please feel free to reach out to us for any queries.

I AI Assistants

AI such as ChatGPT and Claude.ai are used to polish the writing.