UniVocal: Unified Speech-Singing Code-Switching Synthesis

Yufei Shi[1] **Qian Chen**[2] **Wen Wang**[2] **Xiangang Li**[2] **Zhen-Hua Ling**[1] **Yang Ai**[1*]
[1]National Engineering Research Center of Speech and Language Information Processing,
University of Science and Technology of China, Hefei, P. R. China
[2]Independent Researcher
zkddsr2023@mail.ustc.edu.cn, yangai@ustc.edu.cn

Abstract

We propose **UniVocal**, a unified framework that implicitly infers vocal modes from text context to pioneer **Speech-Singing Code-Switching (SCS) Synthesis**—a task where transitions are autonomously driven by textual semantics, akin to seamless human language blending. Unlike single-mode generation or systems relying on switching-control tags, our proposed *UniVocal* implicitly infers vocal modes solely from text context. To achieve this, we employ a data-efficient two-stage curriculum learning strategy that progressively trains a competitive TTS system to acquire the desired SCS capability. Addressing data scarcity, we introduce a scalable pipeline to synthesize diverse code-switching data that is both semantically and acoustically natural, alongside a new multi-scenario benchmark, **SCSBench**. To address limitations of semantic tokenizers in capturing acoustic details, we also introduce refined cent token and Chain-of-Thought (CoT) generation for planning prosody before content generation, effectively enhancing empathetic speech generation and singing melody. Experimental results demonstrate that *UniVocal* achieves state-of-the-art performance on **SCSBench** while maintaining competitive performance on regular speech and singing tasks. Audio samples are available at https://project-univocal-demo.github.io/demo/. The code and dataset are released at https://github.com/FunAudioLLM/FunResearch/tree/main/UniVocal.

1 Introduction

Recent advances in neural architectures and large-scale training data have enabled significant progress in speech and singing synthesis (Du et al., 2024b; Zhang et al., 2025c; Ji et al., 2024). However, a natural vocal behavior remains largely unexplored: seamless code-switching between speech and singing within a single utterance. In daily communication, humans instinctively blend these vocal modes based on semantic context, such as casually humming a melody during conversation, incorporating melodic fragments into storytelling, or using song to aid memory in educational settings. Yet existing audio generation systems cannot automatically determine when to switch between modes based solely on text content. We define this capability as **Speech-Singing Code-Switching (SCS) Synthesis**, the task of generating vocal streams where speech and singing automatically switch based on textual semantics.

Existing audio generation approaches, as illustrated in Figure 1, are highly specialized and not suitable for the hybrid nature of SCS. Text-to-speech (TTS) systems (Du et al., 2024b; Zhou et al., 2025) are limited to spoken prosody, lacking melodic expression. Similarly, music generation (Liu et al., 2025; Lei et al., 2025) and singing voice synthesis (SVS) (Zhang et al., 2024a, 2025c) prioritize following musical rules or provided scores over linguistic content, restricting them to the singing mode. Current unified audio generation frameworks (Lei et al., 2023; Yang et al., 2023; Zhang et al., 2025b) generate only speech or singing based on the input prompt, unable to mix both within a single output. Bark [1] attempts to mix speech and singing generation and is closest to achieving the goal of SCS; however, it relies on *explicit tags* to control switching, lacking semantic awareness and also suffering from unstable performance. Therefore, existing models cannot facilitate automatic speech-singing switching driven solely by text content.

To address this gap, we introduce *UniVocal*, a unified framework for speech-singing code-switching synthesis. Our approach adapts CosyVoice 2 (Du et al., 2024b), a strong TTS

* Corresponding author.

[1] https://github.com/suno-ai/bark

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31479–31496

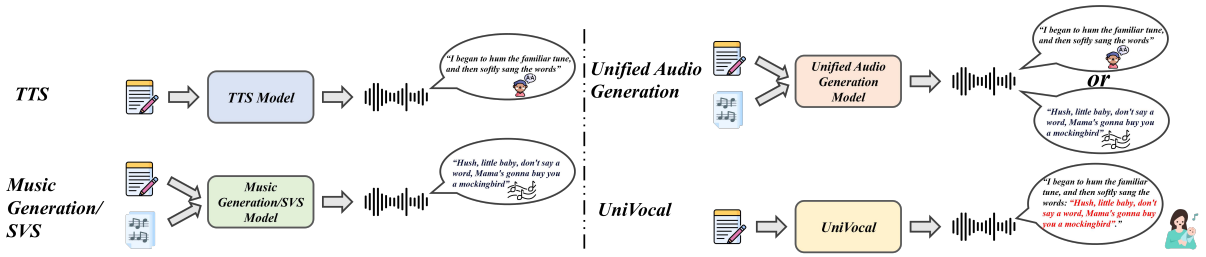

Figure 1: Common audio generation tasks, categorized into specialized tasks on the left and unified tasks on the right. Notably, in addition to its capabilities for regular speech and singing generation, *UniVocal* can also produce vocal streams where speech and singing naturally code-switch.

model, through a data-efficient two-stage curriculum learning strategy. The stage-1 aligns speech and singing representations within a unified latent space, while stage-2 develops automatic switching capability. A critical innovation in our framework is addressing the scarcity of SCS training data. We introduce a scalable pipeline to synthesize diverse code-switching data that is both semantically and acoustically natural. Specifically, we leverage LLMs to generate semantically coherent scripts across multiple scenarios that incorporate diverse switching triggers, ranging from explicit transitional phrases to implicit semantic cues. These scripts are then synthesized using our stage-1 model to ensure consistent acoustic quality, enabling the model to effectively learn semantic triggers for mode transitions.

Beyond the core SCS capability, we address the acoustic limitations of semantic tokenizers, which often discard fine-grained prosodic information. We integrate a Chain-of-Thought (CoT) approach that explicitly models pitch information via interleaved refined cent tokens and semantic tokens. By defining the refined cent token as a high-resolution pitch representation, this design enables the model to plan prosody before content generation, enhancing prosody in speech and melody in singing. This "plan-then-generate" mechanism not only enhances melody in singing but also naturally improves the model's textual empathy capabilities, enabling richer emotional expression in speech.

Experimental results demonstrate that *UniVocal* achieves state-of-the-art mode-switching F1 scores (0.871 objective, 0.810 subjective) on **SCSBench-Mixed**, our code-switching speech-singing test set, outperforming cascaded baselines. Furthermore, the model also enhances empathetic expression while maintaining competitive performance on regular speech and singing generation benchmarks.

Our contributions can be summarized as follows:

- We define the task of **Speech-Singing Code-Switching (SCS) Synthesis** and propose **UniVocal**, a unified framework that, guided by a global instruction defining the task scope, automatically infers vocal modes from textual semantics without explicit tags.
- We introduce a **scalable data synthesis pipeline** and a **two-stage curriculum learning strategy** to address data scarcity. By constructing semantically and acoustically natural synthetic data with diverse scenarios and cues, we enable the model to master SCS capabilities efficiently.
- We introduce a **refined cent token** within a CoT approach to mitigate the acoustic loss of semantic tokenizers. This explicitly models pitch information, enhancing prosodic planning and unlocking latent textual empathy capabilities.
- We construct **SCSBench**, a comprehensive code-switching test set covering multiple cue types and scenarios. Experiments demonstrate state-of-the-art performance on **SCSBench** while maintaining competitive results on regular generation benchmarks.

2 Related Work

Task-specific Audio Generation Audio generation has traditionally been divided into specialized domains with distinct architectures and training objectives. TTS models, such as Seed-TTS (Anastassiou et al., 2024) and CosyVoice (Du et al., 2024a,b) excel in speaker similarity and stability, but their training objectives limit them to spoken prosody patterns, preventing melodic expression. Conversely, music generation (Copet et al., 2023; Yuan et al., 2025) and SVS (Zhang et al., 2022; Wang et al., 2024) models are designed to follow explicit musical scores or structured lyrics, prioritizing musical accuracy over natural linguistic expression (Pan et al., 2025). This fundamental design difference creates a structural barrier: TTS mod-

els lack the melodic modeling capacity required for singing, while music/SVS models lack the prosodic flexibility needed for natural speech. As a result, these specialized approaches cannot support automatic code-switching between speech and singing within a single generation model, required by the SCS task.

Unified Audio Generation Recent efforts have sought to unify multiple audio generation tasks within single frameworks, yet they fail to achieve automatic code-switching. Bark[2] attempts mixed-mode generation through explicit control tags, but this approach requires manual annotation of tags and also suffers from mode transition instability. More importantly, it lacks semantic awareness to determine switching points based on text content alone. Unified frameworks such as UniSyn (Lei et al., 2023) and UniAudio (Yang et al., 2023) employ multi-task training to handle both speech and singing, but they generate only one mode per input based on instruction prompts, preventing intra-sequence switching. Vevo2 (Zhang et al., 2025b) introduces intermediate prosody tokens to model both speech and singing, but it requires reference audio to determine the output vocal mode, making it unable to infer mode switches solely from textual semantics. Different from these works, the proposed UniVocal can automatically infer speech-singing vocal mode transitions based on text content, without explicit control signals.

Prosody Modeling and Tokenization Semantic tokenizers (Hsu et al., 2021; Du et al., 2024a,b) extract high-level linguistic content but discard acoustic details, resulting in flat prosody. Conversely, acoustic tokenizers (Défossez et al., 2022) achieve high fidelity but mix timbre, content, and prosody, making controllability difficult. To bridge this gap, research has explored explicitly separating prosody. Approaches using more F0 information (Zhao et al., 2020; Kharitonov et al., 2021) have shown promise for speech but often lack the melodic precision required for singing. Vevo2 (Zhang et al., 2025b) uses a chromagram-based tokenizer to model both speech and singing; however, the 12-semitone resolution of chromagrams is too coarse to capture the fine-grained prosody of natural speech. Also, above tokenizations require training a complete codec architecture, demanding additional computational and data resources. In contrast to these works, we introduce a refined cent token to supplement the fine-grained acoustic details that are often missing in semantic tokenizers. Furthermore, we employ a CoT strategy to explicitly model pitch information, effectively enhancing both speech prosody and singing melody.

[2] https://github.com/suno-ai/bark

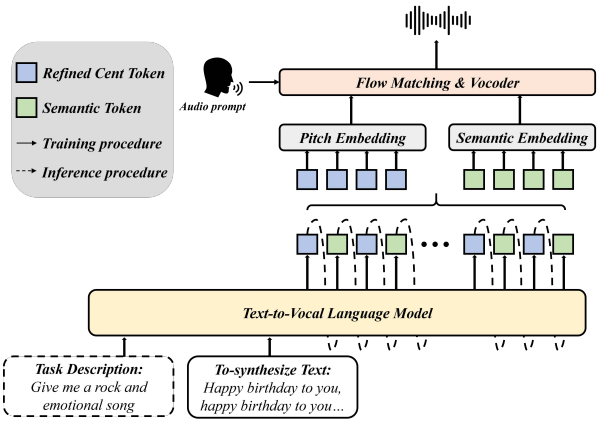

Figure 2: Overview of *UniVocal*. The Text-to-Vocal language model receives the text to be generated, along with an optional natural language description of the task. At each timestep, it autoregressively generates a refined cent token and a semantic token in sequence. These two types of predicted tokens are then fed, along with the prompt audio, into a downstream module to synthesize the final voice output.

3 Methodology

Figure 2 provides an overview of *UniVocal*, which is a unified framework capable of executing diverse vocal generation tasks, including TTS, SVS, and SCS. We build upon semantic-token-based TTS architectures to prioritize generation stability. While our framework is model-agnostic, we employ CosyVoice 2 (Du et al., 2024b) as the backbone in this work due to its robust performance. To guide the model across these distinct tasks, we utilize global task-specific instructions. These instructions define the overall task scope, whereas the fine-grained speech-singing switching within the SCS task is inferred autonomously from text semantics. Furthermore, to address the inherent limitation of semantic tokens in capturing acoustic details, we introduce a refined cent token in a CoT approach. This design explicitly supplements the missing pitch information required for fine-grained speech prosody and singing melody.

3.1 Refined Cent Token

In contrast to raw F_0 in speech, music requires a perceptually meaningful representation (Huang et al., 2020; Tzanetakis et al., 2003; Gupta et al.,

2017). In western music theory, notes are organized into octaves, each containing 12 semitones with a frequency ratio of $2^{1/12}$. Conversely, while semitone scale works for symbolic music, it is too coarse for capturing speech prosody. To bridge this gap, we adopt the cent scale—dividing each semitone into 100 cents—to achieve high-resolution modeling for both speech prosody and singing melody. We define the conversion from linear frequency f_{Hz} to the logarithmic cent scale f_{cent} as follows, using the standard A4 (440 Hz) as the reference pitch:

$$f_{cent} = 1200 \times \log_2 \left(\frac{f_{Hz}}{440} \right), \quad (1)$$

To construct a unified token space that aligns with musical intervals, we project the absolute f_{cent} into a single octave via a modulo operation. The discretized cent token $I(f_{cent})$ is formulated as:

$$I(f_{cent}) = \begin{cases} \lceil f_{cent} \pmod{1200} \rceil & \text{if } f_{Hz} \neq 0 \\ -1 & \text{if } f_{Hz} = 0 \end{cases}, \quad (2)$$

where $f_{Hz} = 0$ denotes unvoiced regions (assigned token -1), and the modulo 1200 operation wraps the pitch trajectory into a 1200-cent range (one octave). The ceiling operation $\lceil \cdot \rceil$ discretizes continuous cent values into integer tokens, introducing a maximum quantization error of 1 cent (approximately 0.08% frequency deviation), which is perceptually negligible. This approach provides a scalable and fine-grained discrete representation that effectively supplements the semantic tokens with rich rhythmic and melodic information. The 1200-bin resolution meets the optimal granularity for the unique demands of *UniVocal*, as validated by our ablation studies (Appendix D.3).

3.2 Model Architecture

The architecture of *UniVocal*, as shown in Figure 2, builds upon the CosyVoice 2 framework. The core backbone is a unified text-to-vocal language model (LM), implemented as a 24-layer causal Transformer with ∼0.5B parameters.

Instruction-Driven Conditioning. The original CosyVoice 2 features an instruction-following mechanism, where a natural language style description (suffixed with a special token *"<|endofprompt|>"*) prefixes the input text to guide generation. We extend this paradigm by designing distinct task-specific instructions for the singing and SCS tasks (see Appendix A.3 for full templates), while retaining the default instruction-free format for regular TTS. These instructions act as global high-level control signals. Specifically for the SCS task, the instruction (such as *"Generate a monologue.<|endofprompt|>"* sets the overall scenario, while the fine-grained switching between speech and singing is automatically driven by the content of the input text itself.

Interleaved Chain-of-Thought Generation. Unlike CosyVoice 2 that directly predicts semantic tokens, our LM autoregressively generates an interleaved sequence of refined cent tokens and semantic tokens (Figure 2). We expand the original LM vocabulary V by appending the set of refined cent tokens V_{cent} (comprising 1201 tokens: 1200 cent values plus the unvoiced token -1), initializing their embeddings randomly. We denote the input text conditioned by instructions as $\mathbf{X}$, and the target interleaved sequence of length T as $\mathbf{Y} = \{(c_1, s_1), \ldots, (c_T, s_T)\}$, where c_t and s_t represent the cent token and semantic token at step t, respectively. Inspired by CoT, we enforce a strictly sequential and interleaved generation order. Both token streams operate at a 25 Hz frame rate. For each frame t, the model first predicts the refined cent token c_t, which contains pitch information. This token is then appended to the history to explicitly guide the prediction of the subsequent semantic token s_t. The joint probability is factorized as:

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{T} P(c_t|\mathbf{X}, \mathbf{Y}_{<t}) \cdot P(s_t|\mathbf{X}, \mathbf{Y}_{<t}, c_t). \quad (3)$$

During inference, we enforce this structure by applying logit masks: at cent-token-prediction steps, we mask out semantic tokens (setting their logits to $-\infty$) to ensure valid sampling from V_{cent}, and vice versa for semantic-token-prediction steps. This masking mechanism ensures strict following of the interleaved generation order. **This interleaved generation strategy makes the model first draft a structural pitch contour—effectively "planning" the prosodic and melodic framework—before generating the specific linguistic content and the remaining acoustic details**. As a modularized component, the refined cent token enables flexible configuration: it can be omitted to prioritize stability for *alignment-heavy* tasks, or integrated to activate CoT prosodic planning for *aesthetic-driven* scenarios.

Waveform Reconstruction. Following the LM stage, we modify the flow matching module of CosyVoice 2 by integrating a randomly initialized embedding layer to process the refined cent tokens

as a supplementary condition. This adapted module generates Mel-spectrograms, which are subsequently reconstructed into waveforms via a pre-trained HiFi-GAN (Kong et al., 2020) vocoder.

3.3 Scalable Data Synthesis Pipeline

To address the scarcity of SCS data, we introduce a scalable pipeline (detailed in Appendix A.1) to synthesize diverse code-switching data that is both semantically and acoustically natural. The pipeline consists of three steps:

Semantic Text Generation. We leverage Gemini 2.5 Pro to generate naturally "boundary-blurring" scripts across diverse scenarios (such as monologues, podcasts). To ensure natural transitions, we design two types of triggers: (1) *Implicit Cues*, which rely on the inherent semantic distinction between conversational prose and lyrical verses; and (2) *Explicit Cues*, where transitional phrases (such as "*reminds me of a tune...*") serve as semantic anchors. This ensures the switches are intrinsically driven by context rather than random splicing.

Unified Acoustic Synthesis. We utilize our stage-1 aligned model to synthesize the audio. Crucially, to maintain acoustic consistency, both speech and singing segments are conditioned on the same speaker embedding, eliminating timbre mismatches. Speech segments are further conditioned on emotion-specific reference audio to match the textual sentiment. Finally, the generated speech and singing segments are concatenated to form the complete code-switching samples.

Quality Control. Finally, we apply a filtering mechanism based on Word Error Rate (WER) to discard samples with severe articulation issues or misalignment, ensuring high-quality training data.

3.4 Two-Stage Curriculum Learning

To progressively equip the baseline model with expanded vocal generation capabilities, we implement a curriculum learning strategy, allowing the model to first bridge the distributional gap between modes and then master the complex task of automatic switching.

Stage-1: Latent Representation Alignment. We align speech and singing distributions within a unified latent space via continued pre-training on CosyVoice 2. We use a 4:1 singing-to-speech ratio (empirically determined to balance melodic learning and speech quality) and format data with task-specific instructions: singing includes style instructions, speech uses standard format. This establishes independent generation capabilities for both modes.

Stage-2: Autonomous Switching Learning. In the SFT stage, we induce SCS capability using synthetic code-switching data. To prevent catastrophic forgetting, we adopt a balanced 1:1:1 mixture of code-switching, speech, and singing data, ensuring competitive performance on regular tasks while mastering SCS.

4 Experimental Setup

4.1 Dataset

Our training data spans three categories.

Code-Switching Data: Following the pipeline in Section 3.3, we constructed 11,769 synthetic samples (262 hours) for the SCS task.

Speech Data: We utilize 960 hours from LibriTTS (Zen et al., 2019), employing the full dataset for stage-1 and a 200-hour subset for stage-2.

Singing Data: We curated 3,700 hours from Suno[3] (cleaning details in Appendix A.2) and included GTSinger (Zhang et al., 2024b). We use the full Suno dataset for stage-1, while sampling a balanced 200-hour subset alongside 10 hours from GTSinger for stage-2.

4.2 Training and Hyperparameters

We build *UniVocal* upon CosyVoice 2. The model is optimized via AdamW with a stage-dependent learning rate schedule. The entire two-stage training requires approximately 6 days on 4 NVIDIA A800 GPUs. Detailed hyperparameters and schedules are provided in Appendix B.

4.3 Evaluation Methodology

4.3.1 Evaluation Datasets

We evaluate *UniVocal* across three distinct capability domains. More details of evaluation settings are provided in Appendix C.1.

Speech-Singing Code-Switching (SCS) Synthesis: To assess mode-switching capabilities, we construct **SCSBench** as a held-out subset of the synthetic data generated in Section 3.3. It is stratified into: *(1)* **SCSBench-Implicit** containing exclusively implicit semantic cues; *(2)* **SCSBench-Explicit** containing explicit trigger phrases cues; and *(3)* **SCSBench-Mixed** incorporating both cue types. This stratification allows us to analyze the model's sensitivity under different conditions.

[3] https://huggingface.co/datasets/nyuuzyou/suno

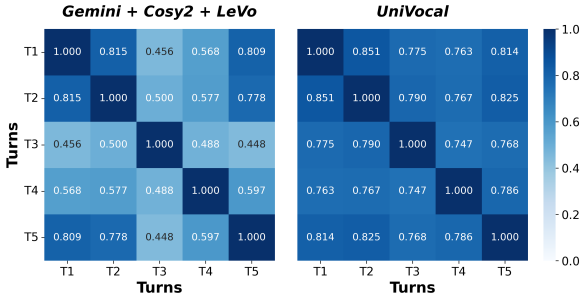

Figure 3: Intra-sample speaker consistency. Pairwise similarity heatmap between five temporal segments, averaged across all generated samples from each system. Darker colors indicate higher speaker stability.

Single-Mode Tasks: We evaluate regular and expressive TTS using the SeedTTS test set (Anastassiou et al., 2024) and a curated textual empathy test set, respectively. For singing, we utilize a held-out GTSinger (Zhang et al., 2024b) subset for short phrases and a constructed *fullsong* set for long-form assessment.

4.3.2 Evaluation Metrics

We employ a comprehensive suite of objective and subjective metrics. Detailed definitions, implementation tools, and validation procedures are provided in Appendix C.2 and C.2.4.

Objective Metrics. We follow standard evaluation protocols (Du et al., 2024b; Lei et al., 2025). For regular TTS, we report WER (semantic consistency) using Whisper-v3 (Radford et al., 2023), SIM (speaker similarity) using ERes2Net (Chen et al., 2023), and UTMOS (Saeki et al., 2022) (naturalness). For singing generation, we evaluate WER, SIM, AES (Tjandra et al., 2025) (aesthetic quality, averaged from CE, CU, PC, and PQ), and QUA (audio quality, averaged from OVRL and SRMR) using ClearVoice[4]. For SCS transition accuracy, we report F1-scores based on segment-level singing mode identification by Gemini 2.5 Pro via In-Context Learning.

Subjective Metrics. Human annotators provide ratings using the 3-point scale: (1) E-MOS (empathy) and P-MOS (prosody) for empathetic TTS; (2) M-MOS (musicality) and N-MOS (naturalness) for singing; and (3) human-labeled F1-scores for SCS. Inter-annotator agreement was substantial ($\kappa = 0.684$).

[4] https://github.com/modelscope/ClearerVoice-Studio/tree/main/speechscore

4.3.3 Model Configuration

UniVocal offers two configurations: standard (omitting refined cent token) for alignment-heavy tasks (SCS, TTS), and expressive (with CoT) for aesthetic tasks (empathetic TTS, singing). All reported results use the optimal configuration per domain.

5 Results

5.1 Main Results

Seamless Mode Switching. We evaluate automatic mode-switching against two constructed cascaded baselines: *Gemini + Bark* and *Gemini + Cosy2 + LeVo*, which utilize Gemini 2.5 Pro for explicit text segmentation (see C.1.1 for baseline details). As shown in Tables 1 and 2, *UniVocal* achieves state-of-the-art F1 scores across SCSBench, surpassing these strong cascaded systems on most subsets. On **SCSBench-Mixed**, it attains F1(O) of 0.871 and F1(S) of 0.810, proving its ability to infer mode switching when given sufficient semantic cues. Our model also achieves the lowest WER and highest UTMOS globally, reflecting superior content consistency and naturalness. While speaker similarity (SIM) slightly lags behind the *Gemini + Cosy2 + LeVo* baseline—likely due to poor quality of the singing training data—our speaker similarity stability analysis shows different results. Adapted from inter-turn speaker consistency metrics in dialogue generation (Zhang et al., 2025a), we partition samples into five temporal segments to compute the average pairwise speaker similarity (Figure 3). Results confirm *UniVocal* maintains significantly higher identity consistency across switches than cascaded baseline, minimizing timbre drift.

Expressive Speech Preservation. *UniVocal* maintains competitive zero-shot TTS performance (Table 3), ranking first in UTMOS on SeedTTS-EN with minimal SIM degradation. Crucially, on the empathy task, it significantly outperforms the CosyVoice 2 baseline (E-MOS/P-MOS +0.48), achieving emotional consistency comparable to commercial systems like ElevenLabs, thus confirming the unlocking of latent empathetic capabilities.

Balancing Melodiousness and Quality. For singing (Table 4), *UniVocal* ranks first in WER and QUA on GTsinger. While objective metrics on the *fullsong* set are consistently high, subjective evaluations provide the definitive validation: *UniVocal* surpasses Vevo 1.5 in both naturalness (2.23 vs 2.17 N-MOS) and musicality (2.18 vs 2.08 M-MOS), successfully balancing vocal fidelity with melodic constraints.

Table 1: Mode-switching accuracy on SCSBench. F1(O) and F1(S) represent metrics evaluated by Gemini 2.5 Pro and human annotators, respectively, assessing the accuracy of speech-singing transition timing. The **bold** and underlined numbers indicate the optimal and sub-optimal results, respectively.

Model	SCSBench-Implicit		SCSBench-Explicit		SCSBench-Mixed	
	F1(O)	F1(S)	F1(O)	F1(S)	F1(O)	F1(S)
Gemini + Bark	0.414	0.142	0.533	0.250	0.465	0.199
Gemini + Cosy2 + LeVo	**0.752**	**0.685**	0.572	0.489	0.607	0.566
UniVocal	0.626	0.595	**0.714**	**0.635**	**0.871**	**0.810**

Table 2: Speech quality metrics on SCSBench. The **bold** and underlined numbers indicate the optimal and sub-optimal results, respectively.

Model	SCSBench-Implicit			SCSBench-Explicit			SCSBench-Mixed		
	WER↓	SIM↑	UTMOS↑	WER↓	SIM↑	UTMOS↑	WER↓	SIM↑	UTMOS↑
Gemini + Bark[1]	21.83	—	3.41	29.47	—	3.31	29.60	—	3.31
Gemini + Cosy2 + LeVo	17.97	**0.758**	3.42	8.18	**0.763**	3.62	12.43	**0.773**	3.54
UniVocal	**5.83**	0.650	**4.36**	**8.80**	0.643	**4.41**	**10.90**	0.652	**4.36**

[1] Since Bark is limited to using fixed registered voices, we did not calculate SIM metrics for this baseline.

Table 3: Performance on zero-shot TTS and textual empathy tasks. The **bold** and underlined numbers indicate the optimal and sub-optimal results.

(a) SeedTTS-EN

Model	WER↓	SIM↑	UTMOS↑
F5-TTS	**2.15**	**0.755**	3.68
CosyVoice 2	2.96	0.744	4.18
Vevo 1.5	12.58	0.718	3.68
UniVocal	2.69	0.703	**4.21**

(b) Textual Empathy Test set

Model	E-MOS↑	P-MOS↑	WER↓
CosyVoice 2	1.78	1.74	0.53
Multilingual-v2[1]	**2.30**	**2.47**	**0.24**
UniVocal	2.26	2.22	0.32

[1] Refers to ElevenLabs' multilingual-v2.

5.2 Ablation Studies

To comprehensively evaluate our framework, we examine the contributions of both the modular refined cent token and the curriculum learning strategy. We compare the expressive configuration (*UniVocal*) against the standard configuration (denoted as *w/o CoT*) and *UniVocal* trained without the two-stage curriculum (denoted as *w/o CL*). Table 5 presents the results of these variants.

Impact of CoT. Comparing the expressive configuration with the standard configuration (*w/o CoT*) reveals a clear trade-off. While removing CoT yields marginal gains in switching stability (SCS F1 increases to 0.810), it comes at the cost of intelligibility (WER increases) and aesthetic performance (with E-MOS and P-MOS decreasing by 0.23 and 0.38, and N-MOS and M-MOS dropping by 0.03 and 0.32, respectively.). Without the pitch planning from CoT, the model suffers a substantial drop in subjective MOS scores, confirming that CoT is essential for highly expressive generation.

Validation of Prosodic Planning. To further verify whether the refined cent token truly acts as a structural planner rather than merely providing supplemental acoustic details, we extracted the generated refined cent tokens during inference and calculated their correlation against the Ground Truth (GT) cent tokens—extracted directly from the final synthesized audios. As shown in Table 6, the positive SRCC and LCC demonstrate that the generated cent tokens effectively outline the primary pitch contour, upon which subsequent semantic tokens supplement finer details. This confirms that *UniVocal* actively drafts a structural pitch framework prior to content generation, fully realizing a prosodic "planning" mechanism. Selected qualitative cases are available on our demo website.

Necessity of Curriculum Learning Strategy. To validate the effectiveness of our progressive adaptation strategy, we trained a variant by jointly optimizing on the full mixture of speech, singing, and synthetic code-switching data (denoted as *w/o CL*), effectively bypassing the stage-1 alignment. As shown in Table 5, this one-stage approach leads to suboptimal convergence. Although the model retains reasonable TTS capabilities, it struggles to capture the subtle semantic triggers for mode-switching (F1 drops to 0.496). This indicates that the latent space alignment established in stage-1 is a prerequisite for effectively mastering the complex

Table 4: Results on the singing generation task. The **bold** and underlined numbers indicate the optimal and sub-optimal results, respectively.

Model	GTsinger				Fullsong					
	AES↑	WER↓	SIM↑	QUA↑	AES↑	WER↓	SIM↑	QUA↑	N-MOS↑	M-MOS↑
Vevo 1.5	5.38	22.79	**0.709**	8.71	5.46	49.55	0.66	6.97	2.17	2.08
YuE[1]	5.32	40.32	0.352	9.33	5.14	77.60	0.46	6.34	2.22	2.24
LeVo[2]	5.25	23.44	0.603	9.30	5.37	69.41	0.54	7.51	**2.41**	**2.34**
UniVocal	**5.44**	**18.07**	0.703	**10.70**	**5.58**	**35.88**	**0.72**	**7.75**	2.23	2.18

[1, 2] For LeVo and YuE, metrics are computed on vocal tracks.

Table 5: Ablation studies on the proposed framework. We report results for the expressive configuration (*UniVocal*), and variants removing the refined cent token (*w/o CoT*) or the curriculum learning strategy (*w/o CL*). The **bold** and underlined numbers indicate the optimal and sub-optimal results, respectively.

Model	Textual Empathy Test Set			Fullsong			SCSBench-Mixed	
	E-MOS↑	P-MOS↑	WER↓	N-MOS↑	M-MOS↑	WER↓	F1↑	WER↓
UniVocal	**2.26**	2.22	**0.32**	2.23	**2.18**	**35.30**	0.716	**5.99**
w/o CoT	2.03	1.84	0.51	2.20	1.86	35.88	**0.810**	10.90
w/o CL	2.24	**2.23**	0.52	**2.29**	2.17	37.21	0.496	14.46

Table 6: Correlation between predicted and GT cent tokens.

Dataset	SRCC	LCC
Textual Empathy Test Set	0.633	0.604
Fullsong Test Set	0.679	0.628

Table 7: Case study on mode-switching under different cue types. **Bold** and *italics* denote expected and successfully generated singing parts, respectively. Implicit Only features semantic distinctions between modes, while Explicit Only uses red trigger phrases with minimized semantic disparity.

Cue Type	Example	Outcome
Explicit + Implicit	He begin to sing softly, always the same tune. ***Oh, the river flows, and the wild wind blows...*** He'd trail off...	Accurate
Explicit Only	There's a lyric that really fits this mood. It goes... ***the streetlights look the same from every window.*** And they really do.	Accurate
Implicit Only	I wasn't just in my kitchen making coffee. ***Mmm-hmm, mmmm...*** It's amazing how a few notes can do that. **We were young and free, just you and me.** And for a second...	Partial Failure

SCS task.

The Source of Empathy Capabilities. CoT is not the sole driver of empathy. The *w/o CoT* variant already outperforms the baseline (E-MOS 2.03 vs 1.78). To isolate the source of this gain, we trained a supplementary variant using only stage-1 data without CoT, which yielded similar E-MOS scores to *w/o CoT*. This confirms that stage-2 data has negligible impact. We therefore attribute empathy to a synergy: emotionally diverse singing data in stage-1 unlocks latent expressive representations, which are further amplified by the refined cent token.

5.3 Qualitative Analysis on Switching Cues

To investigate the model's sensitivity to semantic triggers, we conduct a case study across different types of cues, as defined in Appendix A.1.2: implicit cues refer to the inherent semantic differences between speech and singing, while explicit cues are transitional trigger phrases inserted before singing parts. Table 7 presents the inference outcomes.

Impact of Explicit Anchors. As observed in the first two rows, the presence of explicit cues—whether combined with implicit semantic difference or used alone—significantly improves the switching success. Phrases like *"always the same tune"* and *"It goes..."* serve as strong anchors, effectively priming the model to switch modes at the correct boundary.

Challenges in Implicit-Only Scenarios. In the absence of explicit triggers, the model relies solely on semantic inference, which proves challenging. As shown in the "Implicit Only" case, the model fails to switch for the lyrical line *"We were young and free..."*, likely misinterpreting it as narrative prose due to the lack of structural semantic difference

from speech parts.

The "Humming" Exception. Interestingly, the same implicit-only sample demonstrates a successful switch on the humming segment (*"Mmm-hmm, mm-mm"*). Although classified as an implicit cue, humming possesses a unique non-lexical textual form that contrasts sharply with speech. This distinct feature acts as a "strong" implicit cue, allowing the model to robustly generate singing prosody even without explicit triggers.

Table 8: Mode-switching F1 scores on real-world SCS scenarios.

Model	Real SCS	Enhanced SCS
Gemini + Cosy2 + LeVo	0.452	0.691
UniVocal	0.201	0.730

5.4 Generalization to Real-World Scenarios

To validate *UniVocal*'s generalization capabilities beyond synthetic texts, we collected approximately 30 minutes of real-world human SCS recordings from the internet. Since our training SCS data predominantly consists of speech segments, we trimmed the singing portions of the real-world data to maintain a similar distribution, forming the *Real SCS* test set. Additionally, we created an *Enhanced SCS* test set by manually inserting one explicit semantic cue into the text context of each sample.

As shown in Table 8, *UniVocal* initially faces challenges with the domain gap inherent in real-world SCS scenery (F1 of 0.201). However, its performance surges to 0.730 on the Enhanced set. This demonstrates that with the addition of minor explicit semantic triggers acting as anchors, *UniVocal* generalizes well to real-world scenarios, achieving a mode-switching accuracy close to its in-domain performance.

6 Conclusion

We introduce **UniVocal**, a unified framework that pioneers **Speech-Singing Code-Switching (SCS) Synthesis** by autonomously inferring vocal modes from text. We address data scarcity via a scalable synthesis pipeline and enhance acoustic modeling through a refined cent token with CoT planning. *UniVocal* achieves state-of-the-art performance on **SCSBench** and maintaining competitive performance on regular generation benchmarks.

Limitations

Data Quality Constraints. Our singing training data relies on synthetic songs generated by Suno, which are processed via source separation and ASR tools. Due to the inherent limitations of the source audio and these processing pipelines, a significant portion of the data suffers from artifacts (typified by electronic tones) and lyric misalignment. These imperfections inevitably impose an upper bound on the acoustic fidelity and semantic consistency of the singing segments generated by *UniVocal*.

Gap in Realistic Scenarios. A distributional gap remains between our synthetic training data and complex, real-world SCS scenarios. Consequently, *UniVocal* currently relies on minor explicit semantic triggers to achieve robust generalization in natural settings, highlighting an area for future improvement in handling purely implicit transitions.

Evaluation Precision. While ICL strategy aligns Gemini 2.5 Pro with human preference at the system level (achieving perfect rank consistency), we acknowledge limitations in statistical resolution at the sample level. As detailed in Appendix C.2.4, the magnitude of sample-level correlation coefficients is inherently dampened by the discrete nature of F1 scores on short samples. Since the generated audio segments are often brief, the F1 metric frequently collapses into binary outcomes (0.0 or 1.0), lacking the continuous variance required for high linear correlation metrics. Consequently, while the automated metric serves as a reliable and scalable proxy for system-level benchmarking, its sensitivity to subtle, non-binary quality variations in individual short samples remains limited compared to fine-grained human perception.

Ethical considerations

We acknowledge that the generation capabilities of *UniVocal* could potentially be misused for deepfakes. We emphasize that this research is intended solely for academic purposes. Regarding data compliance, we utilized the open-sourced LibriTTS and Suno datasets, avoiding copyright disputes and privacy concerns associated with real-world vocal recordings. To mitigate risks, we release our models under a restrictive license that strictly prohibits commercial misuse and illegal impersonation.

References

Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, and 1 others. 2024. Seed-tts: A family of high-quality versatile speech generation models. *arXiv preprint arXiv:2406.02430*.

Yafeng Chen, Siqi Zheng, Hui Wang, Luyao Cheng, Qian Chen, and Jiajun Qi. 2023. An enhanced res2net with local and global feature fusion for speaker verification. *arXiv preprint arXiv:2305.12838*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2023. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36:47704–47720.

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

Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, and 1 others. 2024a. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. *arXiv preprint arXiv:2407.05407*.

Zhihao Du, Yuxuan Wang, Qian Chen, Xian Shi, Xiang Lv, Tianyu Zhao, Zhifu Gao, Yexin Yang, Changfeng Gao, Hui Wang, and 1 others. 2024b. Cosyvoice 2: Scalable streaming speech synthesis with large language models. *arXiv preprint arXiv:2412.10117*.

Chitralekha Gupta, Haizhou Li, and Ye Wang. 2017. Perceptual evaluation of singing quality. In *Proc. APSIPA*, pages 577–586. IEEE.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Lin Huang, Chitralekha Gupta, and Haizhou Li. 2020. Spectral features and pitch histogram for automatic singing quality evaluation with CRNN. In *Proc. APSIPA*, pages 492–499.

Shengpeng Ji, Yifu Chen, Minghui Fang, Jialong Zuo, Jingyu Lu, Hanting Wang, Ziyue Jiang, Long Zhou, Shujie Liu, Xize Cheng, and 1 others. 2024. Wavchat: A survey of spoken dialogue models. *arXiv preprint arXiv:2411.13577*.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu-Anh Nguyen, Morgane Rivière, Abdelrahman Mohamed, Emmanuel Dupoux, and 1 others. 2021. Text-free prosody-aware generative spoken language modeling. *arXiv preprint arXiv:2109.03264*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Shun Lei, Yaoxun Xu, Zhiwei Lin, Huaicheng Zhang, Wei Tan, Hangting Chen, Jianwei Yu, Yixuan Zhang, Chenyu Yang, Haina Zhu, and 1 others. 2025. Levo: High-quality song generation with multi-preference alignment. *arXiv preprint arXiv:2506.07520*.

Yi Lei, Shan Yang, Xinsheng Wang, Qicong Xie, Jixun Yao, Lei Xie, and Dan Su. 2023. Unisyn: an end-to-end unified model for text-to-speech and singing voice synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13025–13033.

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. *arXiv preprint arXiv:2502.13128*.

Tu Anh Nguyen, Wei-Ning Hsu, Antony d'Avirro, Bowen Shi, Itai Gat, Maryam Fazel-Zarani, Tal Remez, Jade Copet, Gabriel Synnaeve, Michael Hassid, and 1 others. 2023. Expresso: A benchmark and analysis of discrete expressive speech resynthesis. *arXiv preprint arXiv:2308.05725*.

Ziqian Ning, Shuai Wang, Yuepeng Jiang, Jixun Yao, Lei He, Shifeng Pan, Jie Ding, and Lei Xie. 2025. Drop the beat! freestyler for accompaniment conditioned rapping voice generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24966–24974.

Changhao Pan, Dongyu Yao, Yu Zhang, Wenxiang Guo, Jingyu Lu, Zhiyuan Zhu, and Zhou Zhao. 2025. Synthetic singers: A review of deep-learning-based singing voice synthesis approaches. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 396–416.

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

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2022. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, and 1 others. 2025. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*.

George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. 2003. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, pages 143–152.

Yongqi Wang, Ruofan Hu, Rongjie Huang, Zhiqing Hong, Ruiqi Li, Wenrui Liu, Fuming You, Tao Jin, and Zhou Zhao. 2024. Prompt-singer: Controllable singing-voice-synthesis with natural language prompt. *arXiv preprint arXiv:2403.11780*.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, and 1 others. 2023. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*.

Guanrou Yang, Chen Yang, Qian Chen, Ziyang Ma, Wenxi Chen, Wen Wang, Tianrui Wang, Yifan Yang, Zhikang Niu, Wenrui Liu, and 1 others. 2025. Emovoice: Llm-based emotional text-to-speech model with freestyle text prompting. *arXiv preprint arXiv:2504.12867*.

Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, and 1 others. 2025. Yue: Scaling open foundation models for long-form music generation. *arXiv preprint arXiv:2503.08638*.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*.

Leying Zhang, Yao Qian, Xiaofei Wang, Manthan Thakker, Dongmei Wang, Jianwei Yu, Haibin Wu, Yuxuan Hu, Jinyu Li, Yanmin Qian, and 1 others. 2025a. Covomix2: Advancing zero-shot dialogue generation with fully non-autoregressive flow matching. *arXiv preprint arXiv:2506.00885*.

Xueyao Zhang, Junan Zhang, Yuancheng Wang, Chaoren Wang, Yuanzhe Chen, Dongya Jia, Zhuo Chen, and Zhizheng Wu. 2025b. Vevo2: Bridging controllable speech and singing voice generation via unified prosody learning. *arXiv preprint arXiv:2508.16332*.

Yongmao Zhang, Heyang Xue, Hanzhao Li, Lei Xie, Tingwei Guo, Ruixiong Zhang, and Caixia Gong. 2022. Visinger 2: High-fidelity end-to-end singing voice synthesis enhanced by digital signal processing synthesizer. *arXiv preprint arXiv:2211.02903*.

Yu Zhang, Wenxiang Guo, Changhao Pan, Dongyu Yao, Zhiyuan Zhu, Ziyue Jiang, Yuhan Wang, Tao Jin, and Zhou Zhao. 2025c. Tcsinger 2: Customizable multilingual zero-shot singing voice synthesis. *arXiv preprint arXiv:2505.14910*.

Yu Zhang, Ziyue Jiang, Ruiqi Li, Changhao Pan, Jinzheng He, Rongjie Huang, Chuxin Wang, and Zhou Zhao. 2024a. Tcsinger: Zero-shot singing voice synthesis with style transfer and multi-level style control. *arXiv preprint arXiv:2409.15977*.

Yu Zhang, Changhao Pan, Wenxiang Guo, Ruiqi Li, Zhiyuan Zhu, Jialei Wang, Wenhao Xu, Jingyu Lu, Zhiqing Hong, Chuxin Wang, and 1 others. 2024b. Gtsinger: A global multi-technique singing corpus with realistic music scores for all singing tasks. *Advances in Neural Information Processing Systems*, 37:1117–1140.

Yi Zhao, Haoyu Li, Cheng-I Lai, Jennifer Williams, Erica Cooper, and Junichi Yamagishi. 2020. Improved prosody from learned f0 codebook representations for vq-vae speech waveform reconstruction. *arXiv preprint arXiv:2005.07884*.

Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. 2025. IndexTTS2: A Breakthrough in Emotionally Expressive and Duration-Controlled Auto-Regressive Zero-Shot Text-to-Speech. *arXiv preprint arXiv:2506.21619*.

A Details of Dataset

A.1 Code-Switching Speech-Singing Dataset

A.1.1 Overview

To enable the model to learn autonomous mode switching, we constructed a synthetic code-switching speech-singing dataset comprising 11,769 samples (262 hours). The dataset covers three scenarios—monologue, personal podcast, and audiobook—spanning diverse emotions. Structurally, each sample consists of a spoken narrative naturally interwoven with sung or hummed phrases. We utilize 9 distinct speaker timbres selected from diverse voice characteristics to maximize acoustic diversity. Key statistics of the dataset are summarized in Table 9. A representative subset of 1,200 samples (approximately 10%) is reserved as **SCSBench** for evaluation, maintaining the roughly same scenario distribution as the training set.

A.1.2 Construction Pipeline

The dataset construction follows a three-step pipeline designed to simulate "boundary-blurring" switches:

Semantic Text Generation: To facilitate the construction of naturally "boundary-blurring" switching scripts, we employ Gemini 2.5 Pro (Comanici et al., 2025) to generate content where vocal mode switches are intrinsically driven by semantic context. Designed for first-person scenarios such as monologues, podcasts, and audiobooks, the scripts

are primarily spoken but feature naturally inserted singing or humming segments. To enable *UniVocal* to autonomously infer mode transitions solely from text, we enforce strict semantic distinctiveness between modes via a constrained prompting strategy. This establishes **implicit cues**: speech is maintained in a natural, prose-like conversational tone to drive narrative and logic, whereas singing is characterized by lyrical, repetitive, and emotionally heightened patterns. Humming segments (such as *"hmm hmm hmm"*) utilize inherently unique textual representations. To further enhance robustness, we insert **explicit cues** (transitional trigger phrases such as *"And that reminds me of a lyric..."* or *"Let me try to recall that jingle for you..."*) immediately preceding singing parts in approximately 50% of the samples. These phrases, expanded by Gemini 2.5 Pro from seed examples, serve as strong semantic anchors to stabilize the learning of speech-singing code-switching. All descriptive words containing special tags in scripts are excluded. All generated text content is then compiled into JSON files.

Audio Synthesis: We utilize our stage-1 aligned model to generate segments for both modes. We apply differential conditioning: speech segments are synthesized conditioned on emotion-specific reference audio (sourced from Expresso (Nguyen et al., 2023) and EmoVoice-DB (Yang et al., 2025) datasets) to align with the textual sentiment. This reference audio set comprises prompts from 9 speakers covering nine emotions: *confused, happy, sad, angry, surprised, fearful, disgusted, default, laughing*. In contrast, singing segments are conditioned solely on the target speaker embedding (injected during the Flow Matching inference stage). These segments are subsequently concatenated—with a 0.25s silence interval inserted between segments—to form the complete code-switching audio samples. This approach preserves the specific emotional tone in speech while maintaining a consistent speaker identity across the concatenated sample. The emotion distribution of the resulting samples is skewed towards "*happy*" (50%), "*sadness*" (16%), and "*default*" (11.6%), with other emotional states collectively accounting for the remaining $\sim$12.4%.

Quality Filtering: We filter samples based on Word Error Rate (WER) computed using Whisper-v3 (Radford et al., 2023). The 20% threshold is empirically determined to balance data quality and quantity: samples with WER $\geq$ 20% are discarded as they indicate severe misalignment, while those with moderate WER (10-20%) are retained with ASR-transcribed text to ensure precise alignment. This filtering process removes approximately 15% of initially generated samples. Finally, valid samples are formatted for training by prepending the scenario-specific instruction (such as *"Generate a monologue"*) and the *"<|endofprompt|>"* separator to the to-synthesize text.

Table 9: Key statistics of the code-switching speech-singing dataset

Scenario	Count	Total Dur. (h)	Avg. Dur. (s)
Monologue	6,247	84.3	48.6
Podcast	2,432	87.2	129.1
Audiobook	3,090	90.4	105.3
Sum	**11,769**	**261.9**	**80.1**

A.2 Singing Data Cleaning Pipeline

To enable singing generation, we created an approximately 3,700-hour English singing dataset from the 23,000-hour open-source Suno[5] music dataset. Our objective was to extract only the English vocal tracks to mitigate training complexity. The process began with source separation using MelBand Roformer (viperx edition)[6]. A significant portion (60%) of the separated audio was then filtered out using DNSMOS (OVRL) and SRMR metrics by ClearVoice-Studio[7] to remove tracks with strong background noise and heavy reverberation. We applied MelBand Roformer (anvuew edition)[8] for dereverberation to reduce the electric tone. Next, we segmented the audio using an energy-based voice activity detection (VAD) method, merging adjacent segments into clips up to 4 minutes long. Lyrics were transcribed using FastWhisper [9]. As FastWhisper's WER is high for singing, we followed RapBank's (Ning et al., 2025) methodology, calculating the phoneme-per-second (PPS) rate of the lyrics and discarding segments with rates that were too high or low to filter out hallucinations. Ultimately, we obtained approximately 3,700 hours of

[5] https://huggingface.co/datasets/nyuuzyou/suno

[6] https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/pretrained_models.md

[7] https://github.com/modelscope/ClearerVoice-Studio/tree/main/speechscore

[8] https://github.com/ZFTurbo/Music-Source-Separation-Training/blob/main/docs/pretrained_models.md

[9] https://huggingface.co/Systran/faster-whisper-large-v3

data. We then used Gemini 2.5 Pro to generate 100 natural language instruction templates (see Table 11 for examples), which we populated with style tags from the original metadata (such as *"A song featuring pop...<|endofprompt|>< to-synthesize text>"*). It is important to note that due to limitations in the metadata and separation tools, the quality of the final dataset remains suboptimal. This is primarily manifested as persistent, noticeable electric tone and a weak correlation between the style tags and the actual singing performance. We use the full singing data in the stage-1, whereas for the stage-2, we sample approximately 200 hours for training.

A.3 Instruction Templates

We design distinct instruction formats to condition the model for different generation tasks. The specific prompt structures and templates are detailed below.

Speech-Singing Code-Switching (SCS) Synthesis. As shown in Table 10, the instruction explicitly defines the narrative scenario for the SCS task. We employ three fixed instruction tokens corresponding to the dataset scenarios. During training and inference, one of these instructions prefixes the input text, separated by the special token "<|*endofprompt*|>".

Table 10: Instruction templates for the SCS task. *Note: These prompts activate the SCS task globally. No segment-level tags (e.g.,<sing>, <speech>) are used within the input text.*

Scenario	Instruction Prompt
Monologue	*Generate a monologue.* <|*endofprompt*|>
Podcast	*Generate a podcast.* <|*endofprompt*|>
Audiobook	*Generate an audiobook.* <|*endofprompt*|>

Speech Generation For regular speech synthesis tasks, no additional instruction prefix is required. The model takes the text content as input, following the default behavior of the baseline model.

Singing Generation. For singing tasks, the instruction is constructed to encapsulate the musical style. To ensure robustness and linguistic diversity, we utilized Gemini 2.5 Pro to generate 100 distinct natural language description templates containing placeholders for style tags.

Formally, given the comprehensive metadata for a song, we employ a dynamic sampling strategy to enhance robustness. We randomly sample a subset of style tags S (where $|S| = k$ and $1 \leq k \leq 5$) to fit the maximum capacity of our templates. Subsequently, we select a template T specifically designed with k placeholders to form the final instruction:

$$\text{Input\_text} = T(S) \oplus \text{<|}endofprompt\text{|>} \oplus \text{Lyrics}$$

Table 11 lists a subset of these templates. The `{style}` placeholder is replaced by the selected style tags from the singing metadata during training.

B Training Details

B.1 Hyperparameters

The Language Model (LM) is optimized using AdamW with $\beta_1 = 0.9, \beta_2 = 0.95$, weight decay 0.1, and gradient clipping at 1.0. The learning rate schedules differ by stage:

- **Stage-1:** Linear decay from 2×10^{-4} to zero over 70,000 steps (with 5,000 warmup steps).
- **Stage-2:** Constant learning rate of 1×10^{-4} for 30,000 steps.

Both stages utilize dynamic batching with a maximum capacity of 4.5 minutes per batch.

B.2 Compute Resources

All experiments were conducted on 4 NVIDIA A800 GPUs. Stage-1 training took approximately 5 days, while Stage-2 took 1 day. The Flow Matching module was fine-tuned on Stage-2 data for 2 days using a batch size of 2 minutes per GPU.

C Details of Evaluation

C.1 Evaluation Settings for Each Task

This section details the evaluation dataset statistics, model inference configurations, and baseline implementations that supplement the methodology described in the main text.

C.1.1 Speech-Singing Code-Switching (SCS)

SCSBench Statistics. The constructed **SCSBench** comprises approximately 1,210 samples. The dataset is strictly balanced across two dimensions:

1. **Cue Types:** The three subsets (**SCSBench-Implicit**, **SCSBench-Explicit**, and **SCSBench-Mixed**) each constitute approximately one-third of the total data.

Table 11: Selected examples of the 100 natural language templates used for singing instructions. The `{style}` slot is dynamically filled with metadata tags.

Singing Generation Instruction Templates (Selected)
Generate a song in the `{style}` *style.*<|endofprompt|>
Sing a song with a `{style}` *atmosphere.* <|endofprompt|>
I want a song that encapsulates `{style}`. <|endofprompt|>
I need a `{style}` *song with* `{style}`. <|endofprompt|>
A song that is `{style}`*, featuring* `{style}`. <|endofprompt|>
Generate a `{style}` *song, featuring* `{style}` *and* `{style}`. <|endofprompt|>
Create a track that's `{style}`*,* `{style}`*,* `{style}`*, and* `{style}` *in style.* <|endofprompt|>
... **(100 templates in total)**

2. **Scenarios:** Within each subset, samples are equally distributed across the three narrative scenarios: monologue, podcast, and audiobook.

Inference Configurations. For *UniVocal*, we generate audio using the task-specific instruction (as defined in Appendix A.3) and a randomly assigned speaker for each sample. For the cascade baselines, we implement a multi-stage pipeline to enable code-switching:

- *Gemini + Bark:* We first employ Gemini 2.5 Pro to segment the input text into speech and singing parts based on semantic content. Since Bark has a maximum duration limit (<20s), we further segment long text parts using the NLTK[10] library. After synthesizing each segment independently, we concatenate them with a 0.1s silence interval inserted between mode switches to form the final audio.

- *Gemini + CosyVoice 2 + LeVo:* Similar to the above, we use CosyVoice 2 for speech segments and LeVo for singing segments. Both models are conditioned on the same target speaker to maintain identity consistency. Segments are concatenated with a 0.1s silence interval.

Metric Calculation. To evaluate mode-switching performance, we treat the presence of singing as the positive class and adopt a two-step evaluation procedure. First, at the sentence level, we determine the ground truth and prediction labels. Gemini 2.5 Pro and human annotators are instructed to classify each generated audio segment as speech or singing based on its acoustic characteristics. Second, based on these segment-wise classifications, we compute the metrics:

- Macro-F1: Calculated by computing the F1-score for each individual audio sample (sample-level) and then averaging these scores across the entire test set.

- Micro-F1: Calculated by aggregating all segment-level predictions (TP, FP, FN) globally across the dataset before computing the metric.

C.1.2 Textual Empathetic Speech

The textual empathetic speech test set consists of 50 colloquial sentences designed to elicit emotional expressiveness. The dataset covers 10 distinct emotional scenarios, with 5 samples per category: *anger, disgust, contempt, fear, hope, pride, joy, nostalgia, surprised,* and *sadness*.

During inference, to strictly evaluate the model's ability to infer emotion from text content, all models are conditioned solely on speaker timbre without any emotion-specific reference audio. For the commercial baseline *ElevenLabs multilingual-v2 model*, we utilize the "Brian" voice from official website[11].

C.1.3 Singing Generation

We utilize two test sets to evaluate short-phrase and long-form singing generation capabilities.

- **GTSinger (Short-phrase):** This set contains 200 lyrics samples. Each sample is paired with a reference audio prompt of 5–10 seconds. During inference, all models are conditioned on the style instruction for "Pop" music and perform in-context learning using the provided audio prompt.

[10] https://pypi.org/project/nltk/

[11] https://elevenlabs.io/app/speech-synthesis/text-to-speech

- **Fullsong (Long-form):** This set comprises 27 tracks covering 9 musical styles: *blues, country, electro, emotional, folk, jazz, pop, rap,* and *rock*. Each sample is assigned a style-specific audio prompt (5–20 seconds) sampled from the training data. Models generate singing using the corresponding style instruction and the assigned prompt.

Metric Notes. Due to the limited style diversity in GTSinger, the MuQ-T metric is reported only for the *fullsong* test set. For the FAD calculation, the reference distribution is constructed using the audio prompts from the respective test sets.

C.2 Scoring Criteria and Protocols

This section details the human annotation setup, the specific logic for determining mode-switching accuracy (F1 score), and the granular scoring criteria used for both objective (Gemini-based) and subjective (human-based) evaluations.

C.2.1 Human Annotator Configuration

To ensure high-quality evaluation, we recruited paid participants to perform subjective assessments. The recruitment criteria were tailored to the specific tasks:

- **Singing-Specific Criteria:** For singing generation task, we recruited annotators with professional musical backgrounds or formal musical training to ensure reliable judgment.
- **Other Task's Criteria:** All participants were required to demonstrate high proficiency in English listening comprehension for reliable judgment.

Each audio sample was evaluated by at least three independent annotators. Each audio sample was evaluated by at least three independent annotators. The reported scores represent the mean of these independent ratings. To validate the consistency of human judgments, we computed the inter-annotator agreement using Fleiss' kappa, yielding a score of $\kappa = 0.684$, which indicates substantial agreement among annotators.

C.2.2 F1-score Calculation Protocols for SCS

As described in Appendix C.1, calculating the F1-score for Speech-Singing Code-Switching (SCS) requires determining whether the generated segments match the target mode (singing or speech). The determination logic for both objective (Gemini 2.5 Pro) and subjective (Human annotators) evaluations shares a common definition of singing mode but differs in the matching verification process.

Definition of Singing Mode. Both Gemini 2.5 Pro and human annotators are instructed to classify a segment as "singing" based purely on acoustic and melodic characteristics. The core criterion is the intent of the "virtual speaker". Even if the singing is imperfect (such as off-key), it is classified as the positive class as long as it exhibits a distinct melodic contour distinguishing it from speech.

Matching Logic: Objective (F1(O)). For the Gemini-based evaluation, we employ an In-Context Learning (ICL) strategy, providing the model with acoustic-grounded few-shot examples to mitigate semantic bias. The process follows these steps:

1. **Transcription:** Gemini transcribes the identified singing segments from the generated audio.
2. **Segmentation:** Both the transcribed text and the ground-truth target text are segmented by commas to isolate phrases.
3. **Fuzzy Matching:** We employ the Python *thefuzz*[12] library to calculate the similarity between the transcribed singing segments and the target singing lyrics. A match is declared if the similarity score exceeds 70%.
4. **Exception for Humming:** Due to the high hallucination rate in transcription for non-lexical humming, we bypass *thefuzzy* matching step for humming segments. Instead, success is determined solely by comparing the count of generated humming segments against the expected count in the target text.

Matching Logic: Subjective (F1(S)). For human evaluation, annotators directly compare the generated audio with the target text. The evaluation prioritizes timing and mode accuracy over lyrical perfection. If the model generates singing in the correct time slot but with minor lyrical deviations, it is considered a *True Positive* for the mode-switching task. Such errors are penalized in WER metrics but are treated as successful mode switches. Conversely, speaking the lyrics when singing was required is treated as a mode error (*False Negative*).

[12] https://www.piwheels.org/project/thefuzz/

Table 12: Configuration of the Contrastive Few-Shot Examples used in ICL. The "Hard Negative" forces the model to ignore lyrical text when the acoustics indicate speech.

Case Type	Audio Content Description	Textual Content (Lyrics)	Ground Truth Label
Hard Negative	**Spoken** recitation with natural speaking cadence	*"We were both young when I first saw you..."*	*speech*
True Positive	**Sung** performance with stable pitch and melody	*"We were both young when I first saw you..."*	*sing*

C.2.3 In-Context Learning Strategy for Metric Calibration

Without calibration, standard multimodal LLMs exhibit a strong "semantic bias." For instance, in our preliminary experiments (without ICL), we observed that baseline models like *Gemini + Bark* achieved inflated scores by generating correct lyrics without singing prosody, while valid singing with ASR errors (common in *Gemini + Cosy2 + LeVo*) was unfairly penalized. This discrepancy highlights that naive prompting leads the evaluator to rely on transcribed semantics rather than acoustic realization. To correct this and ensure F1(O) accurately reflects prosodic mode-switching, We implemented a rigorous In-Context Learning strategy comprising two key components: task-specific system instructions and contrastive acoustic demonstrations.

Task-Specific System Instructions. We constructed a structured system prompt that explicitly defines the segmentation rules and labeling conventions. Crucially, the prompt includes a *Negative Constraint* to counteract semantic bias:

> *"Focus exclusively on acoustic features... You must distinguish 'speech' from 'singing' based ONLY on prosody, pitch modulation, rhythm, and melody, NOT on the text content. Ignore semantic cues: Do not classify a segment as 'singing' just because the lyrics look like a song."*

The model is instructed to output a structured log containing timestamps and transcriptions, labeled strictly as *speech*, *sing*, or *hum*.

Contrastive Acoustic Demonstrations (Few-Shot). To further ground the model's understanding in acoustics, we provided two carefully curated 1-shot examples (demonstrations) representing a "Hard Negative" and a "True Positive" case. As illustrated in Table 12, these examples use identical or semantically similar lyrical content but differ significantly in vocal mode:

- **Hard Negative Example (Speech):** We selected an audio sample where a speaker recites the lyrics of a famous song (*Love Story*) in a spoken prosody. Despite the text being clearly lyrical, the ground-truth label provided to Gemini is *speech*. This forces the model to override its textual prior and attend to the flat pitch and lack of musical tempo.

- **True Positive Example (Singing):** We selected a genuine singing sample of the same song. The label provided is *sing*, reinforcing the association between the label and acoustic features like sustained pitch and melodic intervals.

By processing these contrastive examples in the context window before inference, Gemini 2.5 Pro learns to decouple semantic content from vocal mode, significantly improving its reliability as an automated evaluator for the SCS task.

C.2.4 Automated Metric Calibration Analysis

To validate the reliability of Gemini 2.5 Pro as an automated evaluator, we conducted a correlation analysis against human ratings on 243 generated audio samples. We observed a statistically positive correlation at the sample level (Pearson $r = 0.343$, Spearman $\rho = 0.346$, $p < 0.05$).

While the magnitude of these coefficients indicates moderate correlation, it is inherently limited by the discrete nature of F1 scores on short samples. Since sample-level F1 scores often result in binary outcomes (0.0 or 1.0) due to the ability of baselines, the lack of continuous variance dampens linear correlation metrics. However, when aggregating these scores to the system level, the noise averages out. As detailed in the main text, the automated metric exhibits perfect rank consistency with human evaluation across all **SCSBench** subsets, confirming its validity as a proxy for distinguishing relative system performance.

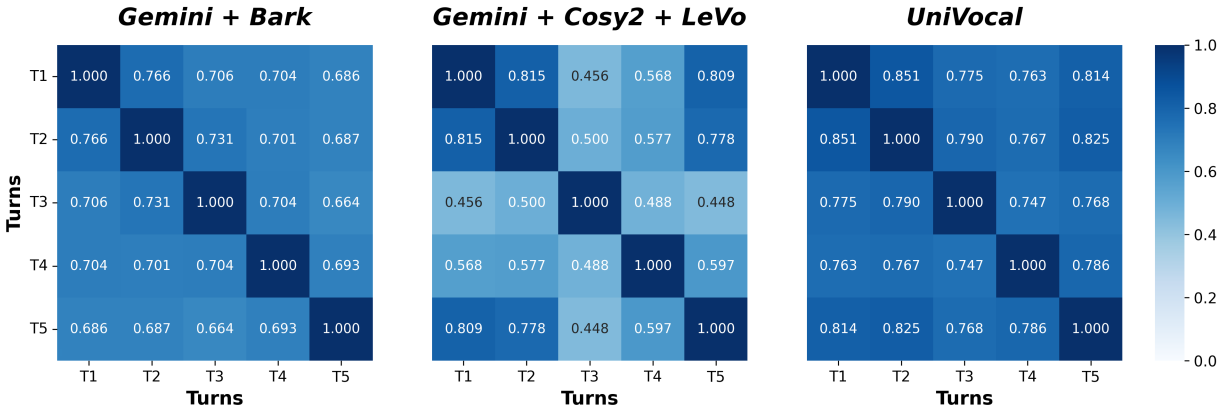

Figure 4: Intra-sample speaker consistency. Pairwise similarity heatmap between five temporal segments, averaged across all generated samples from each system. Darker colors indicate higher speaker stability.

Table 13: Results on the textual empathy task. Metrics suffixed with (O) and (S) denote evaluations by Gemini 2.5 Pro and human annotators, respectively. The **bold** and underlined numbers indicate the optimal and sub-optimal results, respectively.

Model	Textual Empathy Test Set				
	E-MOS(O)↑	P-MOS(O)↑	E-MOS(S)↑	P-MOS(S)↑	WER↓
CosyVoice 2	1.66	1.93	1.78	1.74	0.53
ElevenLabs' multilingual-v2	**2.26**	**2.63**	**2.30**	**2.47**	**0.24**
UniVocal	2.11	2.25	2.26	2.22	0.32

C.2.5 MOS Scoring Criteria for Other Tasks

The Mean Opinion Score (MOS) evaluations utilize a 3-point scale (1.0 to 3.0, in increments of 0.25). While E-MOS and P-MOS are applied identically in both Gemini-based objective and human subjective evaluations, N-MOS and M-MOS are assessed exclusively by human annotators.

E-MOS

- **1.0 (Poor):** The delivery is flat, monotonous, or emotionally neutral, showing no connection to the text's emotional content.
- **2.0 (Acceptable):** An attempt at emotion is audible, but it is either weak, inconsistent, or does not fully match the nuance of the text.
- **3.0 (Excellent):** The emotional delivery is clear, authentic, and perfectly aligns with the emotion and intensity implied by the text. The performance is convincing.

P-MOS

- **1.0 (Poor):** Delivery is unclear, monotonous, or robotic. Words may be slurred. Pausing, rhythm, and stress are unnatural and hinder comprehension.
- **2.0 (Acceptable):** Delivery is generally clear but may have minor flaws. Prosody is understandable but may lack naturalness.
- **3.0 (Excellent):** Delivery is exceptionally clear, fluid, and natural. Intonation, rhythm, and stress effectively enhance the meaning. Every word is crisply enunciated.

N-MOS

- **1.0 (Poor):** The voice sounds distinctly synthesized, metallic, or robotic. Contains significant acoustic artifacts, glitches, or unnatural phase issues.
- **2.0 (Acceptable):** The voice sounds generally human-like but exhibits occasional artifacts or slight unnaturalness in timbre that reveals its synthetic nature.
- **3.0 (Excellent):** The voice is indistinguishable from a real human recording. The timbre is rich, warm, and free of any audible processing artifacts.

M-MOS

- **1.0 (Poor):** Significant off-key notes, unstable pitch, or erratic rhythm. The singing is unpleasant to listen to or musically incoherent.

Table 14: Objective performance across different cent token resolutions.

	Textual Empathy Test Set			Fullsong Test Set	
Resolution	**E-MOS(O)↑**	**P-MOS(O)↑**	**WER↓**	**AES↑**	**WER↓**
12 bins	1.57	1.63	0.46	**3.61**	58.87%
480 bins	1.82	1.97	0.51	3.42	**49.72%**
1200 bins	**1.85**	**2.06**	**0.42**	3.45	56.13%

- **2.0 (Acceptable):** Pitch and rhythm are mostly correct, but the performance lacks expressiveness or "soul." It is technically adequate but musically flat or generic.

- **3.0 (Excellent):** The performance is musically engaging with stable pitch and precise rhythm. It demonstrates musical nuance that enhances aesthetic appeal.

D Additional Experimental Details and Results

D.1 Extended Analysis on Intra-Sample Speaker Consistency

To further investigate the stability of speaker identity during mode transitions, we extend the analysis presented in Section 5.1 by including the *Gemini + Bark* baseline in our intra-sample speaker consistency evaluation. We utilize the same metric methodology: partitioning generated samples into five temporal segments and computing the average pairwise speaker similarity between all segments to visualize identity drift. The comparative results are illustrated in Figure 4.

A critical divergence appears when contrasting intra-sample consistency with the global metrics. Although *Gemini + Cosy2 + LeVo* achieves the highest global speaker similarity (SIM in Table 2) , its intra-sample speaker stability is compromised by the use of distinct models for speech and singing, resulting in noticeable timbre mismatches between segments. Conversely, *UniVocal* and *Gemini + Bark* exhibits superior internal coherence due to their unified architecture. By effectively balancing precise switching timing (F1-scores in Table 1), competitive global similarity, and robust intra-sample consistency, *UniVocal* stands out as the optimal framework for high-quality speech-singing code-switching synthesis.

D.2 Additional Results on Textual Empathy

Table 13 presents the comprehensive evaluation results on the textual empathy test set, comparing objective scores from Gemini 2.5 Pro (suffixed with (O)) against the subjective human ratings (suffixed with (S)) reported in the main text. Notably, while slight deviations exist in absolute values, the system-level rankings remain consistent across both evaluators: *ElevenLabs* > *UniVocal* > *CosyVoice 2*. This strong alignment with human preference confirms that Gemini 2.5 Pro captures the nuances of emotional expression and prosody effectively, validating its potential as a reliable automated evaluator for expressive TTS.

D.3 Ablation on Cent Token Resolution

We conducted a supplementary ablation study to evaluate the impact of different cent token resolutions on generation quality. We trained model variants with 12 bins, 480 bins, and the default 1200 bins. Due to computational constraints, each variant was trained with the Text-to-Vocal LLM for 3 epochs and Flow Matching for 1 epoch.

The objective results in Table 14 indicate that a higher token resolution yields markedly better performance in expressive speech while maintaining competitive results in singing tasks. The 12-bin setting, while sufficient for basic melody, is too coarse to capture the micro-prosody required for empathetic speech, leading to significantly lower E-MOS and P-MOS scores. Thus, our choice of 1200 bins provides the optimal granularity for the unique demands of *UniVocal*.