Adapting Speech Language Model to Singing Voice Synthesis

Yiwen Zhao[1] **Jiatong Shi**[1] **Jinchuan Tian**[1] **Yuxun Tang**[2]
Jiarui Hai[3] **Jionghao Han**[1] **Shinji Watanabe**[1]
[1] Carnegie Mellon University [2] Renmin University of China [3] John Hopkins University

Abstract

Speech Language Models (SLMs) have recently emerged as a unified paradigm for addressing a wide range of speech-related tasks, including text-to-speech (TTS), speech enhancement (SE), and automatic speech recognition (ASR). However, the generalization capability of large-scale pre-trained SLMs remains underexplored. In this work, we adapt a 1.7B parameter TTS pretrained SLM for singing voice synthesis (SVS), using only a 135-hour synthetic singing corpus, ACE-Opencpop. Building upon the ESPNet-SpeechLM, our recipe involves the following procedure: (1) tokenization of music score conditions and singing waveforms, (2) multi-stream language model token prediction, (3) conditional flow matching-based mel-spectrogram generation. (4) a mel-to-wave vocoder. Experimental results demonstrate that our adapted SLM generalizes well to SVS and achieves performance comparable to leading discrete token-based SVS models. Project page with sample and code is available[1].

1 Introduction

Large language models (LLMs) have attracted considerable attention in recent years due to their ability to unify representations across diverse data modalities. This unifying capability enables a single model architecture to scale effectively and generalize across a broad range of tasks. By adopting consistent paradigms for data processing and prediction, LLMs can be efficiently adapted to downstream applications, even in low-resource scenarios.

In the speech domain, prior research has primarily pursued two approaches: (1) fine-tuning language models pretrained on text for speech-related tasks, or (2) training language models directly on speech data. The latter approach, often referred to as Speech Language Models (SLMs), tends to capture fine-grained acoustic characteristics more effectively due to its native exposure to audio signals. However, SLMs are inherently data-intensive, requiring large-scale paired datasets. Consequently, most existing pre-training efforts focus on well-resourced tasks such as text-to-speech (TTS) and automatic speech recognition (ASR).

In contrast, singing voice synthesis (SVS) presents additional challenges. The input consists of richly structured musical scores, including phoneme-level lyrics, precise duration annotations, and MIDI notes. The output is vocal singing that must be both musically and phonetically faithful to these conditions. Compared to TTS, publicly available SVS datasets are far more limited due to restrictive licensing and the labor-intensive nature of score annotation.

To explore the generalization capability of SLMs, we propose adapting a TTS-pretrained SLM to the SVS task. We first tokenize the input music score and target singing waveforms as shown in Fig. 1, formulating a multi-stream token prediction task to fine-tune the LM. The predicted tokens, including SSL and multi-layer codec tokens, are able to be separated and decoded to a waveform using the pretrained codec decoder. However, our primary experiment shows that the raw predicted tokens are noisy, as shown in Tab. 2, with resulting waveforms exhibiting temporal discontinuities, particularly

[1]https://tsukasane.github.io/SLMSVS/

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Music.

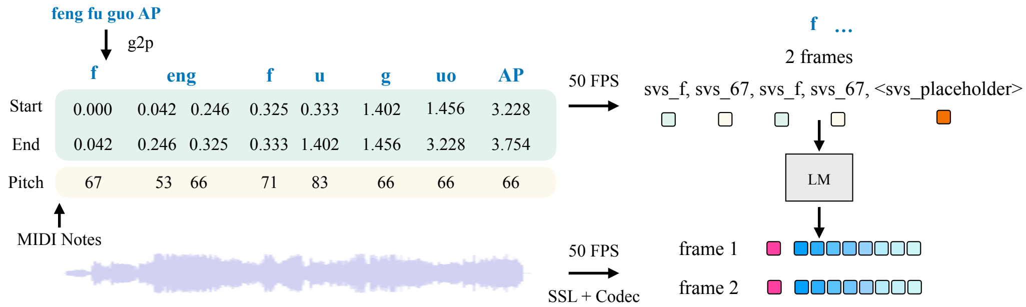

Figure 1: Illustration of music notes tokenization and waveform tokenization. The phoneme, duration, and MIDI are quantized to 50FPS discrete tokens, appended to the TTS vocabulary. The audio tokens are obtained by a pretrained codec encoder and SSL model, with each frame represented by a concatenation of one SSL token and eight codec tokens.

at token boundaries, leading to perceptual glitches and unnatural transitions [1]. Moreover, as the codec model Shi et al. [2024] is pretrained on speech data, it lacks the ability to faithfully resynthesize singing, resulting in a performance upper bound set by the decoder side.

To alleviate the unsatisfactory performance introduced by the codec decoder and the noisy tokens, we use a conditional flow matching model, converting the source Gaussian noise to the target mel spectrogram conditioned on the codec, and additionally train a vocoder Kong et al. [2020] that is consistent with the codec STFT parameters. This optimization enables high-quality singing synthesis while addressing the limitations posed by data scarcity. Considering the expressiveness of synthesized singing, we strengthen the condition of pitch information again through the flow matching process, which improves the melodious fidelity.

Empirical results demonstrate that this second-stage refinement improves synthesis quality, yielding smoother transitions and enhanced pitch accuracy. Overall, our framework enables SLM-based SVS to achieve performance comparable to leading discrete SVS systems. A detailed introduction of related works is attached in appendix A.

2 Methodology

Given the music score $M := (M^{\text{ph}}, M^{\text{pi}}, M^{\text{du}})$, SVS targets to generate a human singing phrase $Y \in \mathbb{R}^{T}$ that align with M, where T is the number of samples in the waveform, $M^{\text{ph}} \in \mathbb{R}^{T'}$, $M^{\text{pi}} \in \mathbb{R}^{T'}$, $M^{\text{du}} \in \mathbb{R}^{T'}$ represents information about phoneme, pitch, and duration over the sequence of the same length T'. We first introduce the SVS data tokenization, then formulate the SVS fine-tuning on SLM, and lastly demonstrate the conditional flow refinement pipeline.

2.1 SVS Data Tokenization

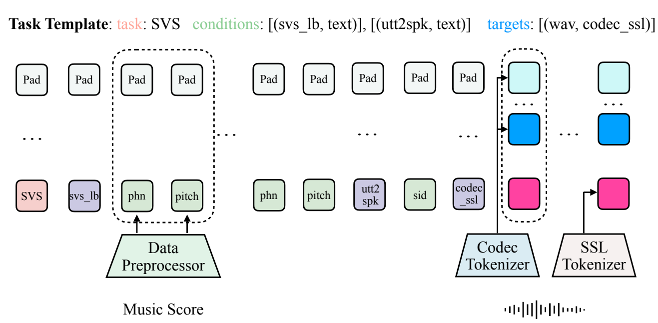

Figure 2: SVS fine-tuning task template. The audio codec tokens and SSL tokens are concatenated along the RVQ stream axis.

Our pipeline is built upon Espnet-SpeechLM Tian et al. [2025b,a]. For SVS, we introduce a new modality called svs_lb, which consists of frame-level pitch, duration, and phoneme conditions. Each unit in svs_lb is a two-element tuple, including a phoneme token and a pitch token with svs prefix. We use the repeat time of (phn, pitch) tuple to implicitly represent duration, which is calculated as: repeat $= (m_{\text{ed}} - m_{\text{st}}) \times \text{fps}$, where m_{ed} and m_{st} represent the end and start time of an element in M^{du}. This process aligns the annotation with the audio codec sample rate. We show our SVS data tokenization pipeline in Figure 1. Our prompt includes the svs_lb and spk_prompt; the target tokens are set to be the concatenation of codec and SSL tokens of singing waveforms.

2.2 Language Model Formulation

For audio representation, we use two types of tokens: the high-level semantics tokens obtained from SSL, and the low-level acoustic tokens from the audio codec. We follow the pre-trained TTS model

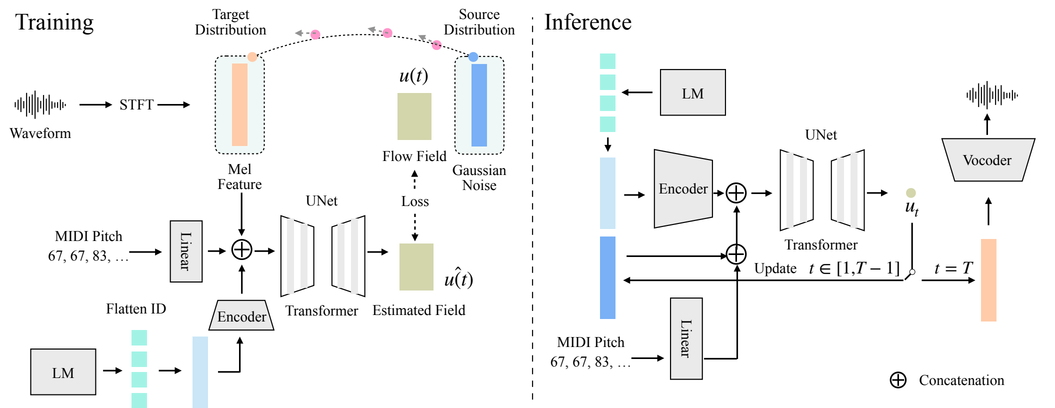

Figure 3: Training and inference process of flow matching.

to use a multi-stream discrete audio representation $\mathbf{s}_f \in \mathbb{N}^{F \times n_q}$ for each frame, where n_q stands for the number of streams, F is the total frame number. Specifically, we concatenate the audio codec tokens and the SSL tokens along the stream dimension, intending to balance their advantages for prediction and acoustic reconstruction. Then, we only keep the codec tokens for decoding. Task and template definitions are shown in Figure 2.

In SLM, tasks can be uniformly formulated as predicting the target sequence based on the input condition sequence. Adapting this template to SVS, we have the inputs $\mathbf{m} = [\mathbf{m_1}, \mathbf{m_2}, ...\mathbf{m_F}]$, which is the frame-level music score unit, speaker prompt $\mathbf{p} = [\mathbf{p_1}, \mathbf{p_2}, ...\mathbf{p_F}]$; and the target frame-level singing feature $\mathbf{s} = [\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_F}]$. As a token classification task, the fine-tuning objective is to maximize the posterior likelihood $P(\mathbf{s}|\mathbf{m}, \mathbf{p})$ using cross-entropy loss.

2.3 Flow-Based Refinement

Flow models are a class of generative models that learn a velocity field capable of transporting samples from a source distribution to a target distribution. In this work, we adopt flow matching to train the flow model efficiently and scalably. During training, the model regresses the velocity field along a probabilistic interpolation path by sampling intermediate timesteps $\mathbf{t} \in [0, 1]$. At inference, a sample from the source distribution is provided, and the trained velocity field is used to transport it towards the corresponding sample in the target distribution.

Let the source distribution be denoted as $X_0 \sim p$, where p is a standard Gaussian distribution. The corresponding target samples are drawn from $X_1 \sim q$, where q is the distribution of mel features obtained from the target waveforms from STFT. The goal of the flow model Ψ is to learn a smooth mapping from X_0 to X_1 through a continuous-time velocity field, which follows a continuous-time Markov process. The evolution of a sample over time is governed by:

$$X_{t+h} = \psi_{t+h|t}(X_t), \quad t \in [0, 1], \tag{1}$$

where $\psi_{t+h|t}$ denotes the transition function over a small time increment h. The instantaneous velocity of a point along its trajectory is defined as:

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)). \tag{2}$$

where u_t is the velocity field at time t. For flow matching, we adopt a linear interpolation path between source and target samples. This formulation corresponds to the optimal transport path under a kinetic energy minimization constraint,

$$\psi_t(x \mid x_1) = (1 - t)x + tx_1. \tag{3}$$

We fuse the LM-predicted codec and pitch signal as additional conditions to make the flow controllable. Let C denote the conditioning input, which contains $\mathbf{s}$ and other optional choices. Let θ represent the parameters of the conditional flow model. The training objective of Conditional Flow Matching (CFM) is to minimize the squared error between the ground-truth velocity and the model-predicted velocity at a set of points sampled from the path from source distribution to target distribution:

$$\mathcal{L}_{\text{CFM}(\theta)} = \mathbb{E}_{t,(X_0,X_1,C)\sim\pi_{0,1,C}} |u_t(X_t \mid X_1) - u_t^\theta(X_t \mid C)|^2, \tag{4}$$

$$u_t^\theta(x \mid c) : [0, 1] \times \mathbb{R}^d \times \mathbb{R}^k \to \mathbb{R}^d. \tag{5}$$

Table 1: Comparison of discrete SVS systems.

Strategies	ACE-Opencpop					
	F0_RMSE↓	F0_CORR↑	MCD↓	PER↓	SingMOS↑	Sheet-SSQA↑
XiaoiceSing	71.67	0.62	11.47	0.09	3.88	3.62
TokSing	55.83	0.67	6.77	0.19	4.08	3.89
LM + Flow1 + Voc	62.79	0.60	7.86	0.36	4.09	3.79

Table 2: Ablation on recipe designs.

Conditions	ACE-Opencpop						
	F0_RMSE↓	F0_CORR↑	MCD↓	PER↓	Singer-Sim↑	SingMOS↑	Sheet-SSQA↑
CD Resynthesis	51.38	0.73	5.84	0.19	0.67	3.95	3.78
LM + CD	62.90	0.60	8.26	0.56	0.49	3.65	3.08
LM + Flow1 + CD	61.51	0.60	8.44	0.45	0.49	3.64	3.08
LM + Flow1 + Voc	62.79	0.60	7.86	0.36	0.61	4.09	3.79
LM + Flow2 + Voc	62.52	0.62	7.66	0.42	0.56	3.95	3.55

where $u_t^\theta(x \mid c)$ is the predicted velocity field function. During inference, we solve the corresponding ordinary differential equation (ODE) using a numerical ODE solver, which starts from $X_0 \sim \mathcal{N}(0, I)$, and integrates the velocity field forward in time to obtain the target sample in X_1.

$$\frac{d}{dt} X_t = u_t^\theta(X_t), \tag{6}$$

$$Y = \text{Voc}(x_1), \quad x_1 \in X_1. \tag{7}$$

The ultimate sample $x_1 \in X_1$ is further converted to a waveform Y using a vocoder,

3 Experiment

3.1 Corpus and Parameters Setups

ACE-Opencpop is a synthetic corpus that inherits the song list from 5.2-hour Mandarin female singing corpus Opencpop, but curates the singing of 30 additional singers using ACE Studio with manual tuning, resulting in the largest, 135-hour opensourced SVS corpus. Detailed parameters setup for SLM fine-tuning and flow matching are shown in appendix B.

3.2 Results Analysis and Ablations

Explanation of the abbreviations used in Tab. 1 and Tab. 2. XiaoiceSing uses music score information to predict discrete tokens and train a vocoder on discrete tokens to waveform. **TokSing** also builds on a discrete NAR architecture, but further introduces a melody predictor and a music enhancer to improve pitch precision. **CD Resynthesis** denotes directly encoding and decoding singing waveforms using a speech-pretrained codec model. **Flow1** refers to flow matching conditioned on the LM-predicted codec features, while **Flow2** conditions on both the LM-predicted codec features and the pitch tensor. **+CD** indicates that the flow output is in the codec embedding space and is converted to waveforms via a pretrained codec decoder. **+Voc** indicates that the flow output is in the mel-spectrogram space and is converted to waveforms via a vocoder.

Metrics used in Tab. 1 and Tab. 2. F0_RMSE and **F0_CORR** Hayashi et al. [2020] measure the root mean squared error and correlation between the fundamental frequency (F0) of the synthesized and reference singing signals, focusing on pitch accuracy. **MCD** Kubichek [1993] quantifies the spectral distance between the generated and ground-truth audio using mel-cepstral coefficients. **PER** denotes phoneme error rate. **SingMOS** Tang et al. [2024] and **Sheet-SSQA** Huang et al. [2024] is a pseudo MOS predictor based on a 5-point mean opinion score scale.

Quantitative Analysis. As shown in Tab. 1, our SLM-based SVS pipeline achieves performance comparable to state-of-the-art discrete SVS models. Compare with XiaoiceSing Lu et al. [2020], our model performs better in pitch-related f0 metrics and overall singing quality, as indicated by pseudo MOS. While the pitch accuracy (reflected by the f0-related metrics) slightly lags behind TokSing Wu et al. [2024], the overall singing quality matches or even surpasses it, demonstrating the effectiveness of our architectural design and the strong downstream generalization capability of SLM. The ablation study highlights the impact of using mel versus codec features as the flow output space: mel features appear easier to model, as indicated by the greater improvement over the LM+CD baseline. Also, as the speech-pretrained codec model leads to information loss in resynthesis already, it sets an upper bound for using the codec feature to form the flow output space. Moreover, incorporating pitch information into flow matching yields a modest gain in pitch fidelity, which is shown by the comparison between LM + Flow1 + Voc and LM + Flow2 + Voc.

4 Conclusion

In this paper, we present a pipeline that adapts a TTS-pretrained SLM for the SVS task, achieving performance on par with state-of-the-art discrete SVS methods. This demonstrates the strong generalizability of SLMs in low-resource downstream settings and points to promising directions for future multi-task SLM research.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Ye Bai, Haonan Chen, Jitong Chen, Zhuo Chen, Yi Deng, Xiaohong Dong, Lamtharn Hantrakul, Weituo Hao, Qingqing Huang, Zhongyi Huang, Dongya Jia, Feihu La, Duc Le, Bochen Li, Chumin Li, Hui Li, Xingxing Li, Shouda Liu, Wei-Tsung Lu, Yiqing Lu, Andrew Shaw, Janne Spijkervet, Yakun Sun, Bo Wang, Ju-Chiang Wang, Yuping Wang, Yuxuan Wang, Ling Xu, Yifeng Yang, Chao Yao, Shuo Zhang, Yang Zhang, Yilin Zhang, Hang Zhao, Ziyi Zhao, Dejian Zhong, Shicen Zhou, and Pei Zou. Seed-music: A unified framework for high quality and controlled music generation, 2024. URL `https://arxiv.org/abs/2409.09214`.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan. Espnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 7654–7658. IEEE, 2020.

Wen-Chin Huang, Erica Cooper, and Tomoki Toda. Mos-bench: Benchmarking generalization abilities of subjective speech quality assessment models. *arXiv preprint arXiv:2411.03715*, 2024.

Eugene Kharitonov, Damien Vincent, Zalán Borsos, Raphaël Marinier, Sertan Girgin, Olivier Pietquin, Matt Sharifi, Marco Tagliasacchi, and Neil Zeghidour. Speak, read and prompt: High-fidelity text-to-speech with minimal supervision, 2023. URL `https://arxiv.org/abs/2302.03540`.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Robert Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, volume 1, pages 125–128. IEEE, 1993.

Ruiqi Li, Zhiqing Hong, Yongqi Wang, Lichao Zhang, Rongjie Huang, Siqi Zheng, and Zhou Zhao. Accompanied singing voice synthesis with fully text-controlled melody. *CoRR*, abs/2407.02049, 2024. URL `https://doi.org/10.48550/arXiv.2407.02049`.

Peiling Lu, Jie Wu, Jian Luan, Xu Tan, and Li Zhou. Xiaoicesing: A high-quality and integrated singing voice synthesis system. *arXiv preprint arXiv:2006.06261*, 2020.

Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International conference on machine learning*, pages 8599–8608. PMLR, 2021.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

Keijiro Saino, Heiga Zen, Yoshihiko Nankaku, Akinobu Lee, and Keiichi Tokuda. An hmm-based singing voice synthesis system. In *INTERSPEECH*, pages 2274–2277, 2006.

Jiatong Shi, Jinchuan Tian, Yihan Wu, Jee-Weon Jung, Jia Qi Yip, Yoshiki Masuyama, William Chen, Yuning Wu, Yuxun Tang, Massa Baali, Dareen Alharthi, Dong Zhang, Ruifan Deng, Tejes Srivastava, Haibin Wu, Alexander H. Liu, Bhiksha Raj, Qin Jin, Ruihua Song, and Shinji Watanabe. Espnet-codec: Comprehensive training and evaluation of neural codecs for audio, music, and speech. In *IEEE Spoken Language Technology Workshop, SLT 2024, Macao, December 2-5, 2024*, pages 562–569. IEEE, 2024.

Martin Strauss, Matteo Torcoli, and Bernd Edler. Improved normalizing flow-based speech enhancement using an all-pole gammatone filterbank for conditional input representation. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 444–450. IEEE, 2023.

Yuxun Tang, Jiatong Shi, Yuning Wu, and Qin Jin. Singmos: An extensive open-source singing voice dataset for MOS prediction. *CoRR*, abs/2406.10911, 2024. URL `https://doi.org/10.48550/arXiv.2406.10911`.

Jinchuan Tian, William Chen, Yifan Peng, Jiatong Shi, Siddhant Arora, Shikhar Bharadwaj, Takashi Maekaku, Yusuke Shinohara, Keita Goto, Xiang Yue, Huck Yang, and Shinji Watanabe. Opuslm: A family of open unified speech language models. *CoRR*, abs/2506.17611, 2025a.

Jinchuan Tian, Jiatong Shi, William Chen, Siddhant Arora, Yoshiki Masuyama, Takashi Maekaku, Yihan Wu, Junyi Peng, Shikhar Bharadwaj, Yiwen Zhao, Samuele Cornell, Yifan Peng, Xiang Yue, Chao-Han Huck Yang, Graham Neubig, and Shinji Watanabe. Espnet-speechlm: An open speech language model toolkit. *CoRR*, abs/2502.15218, 2025b. URL `https://doi.org/10.48550/arXiv.2502.15218`.

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2(3), 2023.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Yanze Wang, Xuming Han, Shuai Lv, Ting Zhou, and Yali Chu. Llfm-voice: Emotionally expressive speech and singing voice synthesis with large language models via flow matching. *Preprints*, April 2025. doi: 10.20944/preprints202504.1831.v1. URL `https://doi.org/10.20944/preprints202504.1831.v1`.

Yongqi Wang, Ruofan Hu, Rongjie Huang, Zhiqing Hong, Ruiqi Li, Wenrui Liu, Fuming You, Tao Jin, and Zhou Zhao. Prompt-singer: Controllable singing-voice-synthesis with natural language prompt. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4780–4794. Association for Computational Linguistics, 2024.

Yuning Wu, Chunlei Zhang, Jiatong Shi, Yuxun Tang, Shan Yang, and Qin Jin. Toksing: Singing voice synthesis based on discrete tokens. In Itshak Lapidot and Sharon Gannot, editors, *25th Annual Conference of the International Speech Communication Association, Interspeech 2024, Kos, Greece, September 1-5, 2024*. ISCA, 2024.

Chong Zhang, Yukun Ma, Qian Chen, Wen Wang, Shengkui Zhao, Zexu Pan, Hao Wang, Chongjia Ni, Trung Hieu Nguyen, Kun Zhou, Yidi Jiang, Chaohong Tan, Zhifu Gao, Zhihao Du, and Bin Ma. Inspiremusic: Integrating super resolution and large language model for high-fidelity long-form music generation, 2025. URL `https://arxiv.org/abs/2503.00084`.

Yongmao Zhang, Heyang Xue, Hanzhao Li, Lei Xie, Tingwei Guo, Ruixiong Zhang, and Caixia Gong. Visinger2: High-fidelity end-to-end singing voice synthesis enhanced by digital signal processing synthesizer. In Naomi Harte, Julie Carson-Berndsen, and Gareth Jones, editors, *24th Annual Conference of the International Speech Communication Association, Interspeech 2023, Dublin, Ireland, August 20-24, 2023*, pages 4444–4448. ISCA, 2023. URL `https://doi.org/10.21437/Interspeech.2023-391`.

A Related Works

A.1 Speech Language Models

Large language models have witnessed significant advancements in recent years, primarily driven by increased model scale, enhanced data availability, and improved training paradigms. Their ability to learn unified representations across modalities has enabled progress on a wide range of tasks.

Recent works have extended this paradigm to the audio domain, resulting in speech language models (SLMs) that treat speech as a sequence of discrete tokens, analogous to natural language Abouelenin et al. [2025], Chu et al. [2024], Ding et al. [2025], Wang et al. [2023], Kharitonov et al. [2023]. SLMs demonstrate strong transferability from general pretraining to specific downstream tasks, using techniques like fine-tuning Ding et al. [2025], Chu et al. [2024] and LORAs Abouelenin et al. [2025], while pretraining is typically conducted on large-scale, general-purpose corpora, domain adaptation is feasible through fine-tuning on curated task-specific datasets.

Previous works also explore using a language model to generate music Zhang et al. [2025], Bai et al. [2024]. However, training the language model on singing requires a large amount of data, which is always closed-source and hard to access by the community. Our work adapt TTS-pretrained SLM to a low-resource SVS setting.

A.2 Singing Voice Synthesis

Singing voice synthesis (SVS) aims to generate expressive and intelligible singing voices from structured musical inputs, typically including phoneme-level lyrics, MIDIs, and note durations. Compared to TTS, SVS requires a higher degree of temporal precision and pitch accuracy to maintain musicality. This renders SVS more sensitive to the alignment between input conditions and the generated acoustic output. Early approaches in SVS were based on concatenative and HMM-based methods Saino et al. [2006], while recent work has shifted towards neural vocoder-based systems Lu et al. [2020], including encoder-decoder architectures and end-to-end frameworks Zhang et al. [2023].

Recent studies have begun leveraging large language models (LLMs) for singing voice synthesis (SVS) and text-to-song generation, enabling more flexible and controllable singing beyond traditional alignment-based methods. Prompt-Singer Wang et al. [2024] introduces a prompt-based SVS system that controls vocal style (e.g., timbre, range) via natural language, using a decoder-only transformer with a range–melody decoupled pitch representation for intuitive style manipulation. MelodyLM / TTSong Li et al. [2024] advances toward fully text-driven song generation by predicting melody representations from lyrics and textual descriptions, integrating LLM-based melody modeling with diffusion-based accompaniment synthesis. LLFM-Voice Wang et al. [2025] unifies expressive speech and singing synthesis using an LLM front-end and flow-matching acoustic model, achieving smoother emotional expression and higher-fidelity vocal rendering.

Our work differs by leveraging a general-purpose speech language model, pre-trained on TTS data, and adapting it to SVS through token-level modeling and conditional refinement, enabling the complex SVS to be a subtask of a unified model.

A.3 Flow-Based Models

Flow-based generative models, such as RealNVP Dinh et al. [2016] and Glow Kingma and Dhariwal [2018], are a class of invertible neural networks that learn data distributions through a sequence of bijective transformations. These classical flow models enable exact likelihood estimation and efficient sampling, and have been successfully applied to high-fidelity speech and audio generation tasks, including vocoding Prenger et al. [2019], speech enhancement Strauss et al. [2023], and expressive speech synthesis Popov et al. [2021].

In contrast, flow-matching approaches Tong et al. [2023] are ODE-based methods conceptually closer to diffusion models (which are SDE-based). Instead of performing explicit density estimation like classical flows, flow-matching trains conditional flows via velocity field learning, providing a scalable and flexible framework for conditional generation.

In this work, we adopt a conditional flow-matching model to generate mel-spectrogram features from Gaussian noise, conditioned on LM-predicted codec embeddings as well as musical pitch labels. This approach refines the acoustic realism and pitch fidelity of generated singing voices while avoiding the computational overhead of traditional flow-based likelihood estimation.

B Model Parameter Setups

B.1 Language Model Finetuning.

We finetune the model using `DeepSpeed` with mixed-precision (`FP16`) training, Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, learning rate 5×10^{-6}), and ZeRO stage-2 optimization for memory efficiency. A Warmup–Cosine learning rate scheduler with 7×10^5 total steps (minimum LR ratio 0.3) is employed. Gradient clipping is set to 1.0. Contiguous memory and communication-overlap optimizations are enabled to ensure stable and scalable training.

B.2 Flow Matching.

We train the model for a total of 30 epochs with a batch size of 64. The learning rate is scheduled to decay linearly from 3×10^{-4} to 1×10^{-4} between steps 2×10^5 and 5×10^5.