

CtrSVDD: A Benchmark Dataset and Baseline Analysis for Controlled Singing Voice Deepfake Detection

Yongyi Zang¹, Jiatong Shi², You Zhang¹, Ryuichi Yamamoto³, Jionghao Han²,
Yuxun Tang⁴, Shengyuan Xu⁵, Wenxiao Zhao⁵, Jing Guo⁵, Tomoki Toda³, Zhiyao Duan¹

¹University of Rochester, Rochester, NY, USA ²Carnegie Mellon University, Pittsburgh, PA, USA
³Nagoya University, Nagoya, Japan ⁴Renmin University of China ⁵Timedomain.ai, Beijing, China

svddchallenge@gmail.com, you.zhang@rochester.edu

Abstract

Recent singing voice synthesis and conversion advancements necessitate robust singing voice deepfake detection (SVDD) models. Current SVDD datasets face challenges due to limited controllability, diversity in deepfake methods, and licensing restrictions. Addressing these gaps, we introduce CtrSVDD, a large-scale, diverse collection of bonafide and deepfake singing vocals. These vocals are synthesized using state-of-the-art methods from publicly accessible singing voice datasets. CtrSVDD includes 47.64 hours of bonafide and 260.34 hours of deepfake singing vocals, spanning 14 deepfake methods and involving 164 singer identities. We also present a baseline system with flexible front-end features, evaluated against a structured train/dev/eval split. The experiments show the importance of feature selection and highlight a need for generalization towards deepfake methods that deviate further from training distribution. The CtrSVDD dataset^{1,2} and baselines³ are publicly accessible. **Index Terms:** singing voice deepfake detection, anti-spoofing, benchmark, dataset

1. Introduction

The rapid advancement of generative artificial intelligence (AI) technologies has initiated a new era in audio deepfakes, drastically improving the quality of synthesized singing voice. Singing voice synthesis (SVS) [1, 2], analogous to text-to-speech (TTS), transforms lyrics and musical scores into singing vocals. Singing voice conversion (SVC) [3], analogous to voice conversion (VC), transforms one singer to sound like another singer’s voice without changing the lyrics and musical score. These advancements also give rise to significant concerns within the music industry. Artists, record labels, and publishing houses are increasingly alarmed by the potential for unauthorized deepfake reproductions that closely mimic well-known singers [4], posing a direct threat to original artists’ commercial value and intellectual property rights. The situation urgently calls for robust methods to protect against the unauthorized use of singing deepfake technologies.

Research has emerged towards the singing voice deepfake detection (SVDD) task as a response to synthesized singing voices. Our previous work introduced a multilingual in-the-wild dataset, SingFake [5], by collecting deepfake song clips from user-generated content websites. The label of bonafide or deepfake was identified by the uploaders and manually verified by the annotators. The synthesis methods utilized rely on uploaders to disclose and are often lacking: 60.6% of deepfake

songs in SingFake are reported with the “Unknown” generation method. Furthermore, out of all song clips that reported the generation method in SingFake, 92.2% reported variants of So-VITS⁴, indicating a potential lack of diversity. Xie et al. [6] curated a controlled Chinese song dataset for fake song detection (FSD). The deepfake songs are generated using one SVS and four SVC methods applied to the bonafide songs they collected. Due to licensing restrictions, its bonafide set is not publicly available. Both works found that speech-trained deepfake detection models cannot directly work on the SVDD task, highlighting the need for a singing voice deepfake detection dataset.

In this work, we present CtrSVDD, a benchmark dataset curated for controlled SVDD with enhanced **controllability, diversity, and data openness** that we believe could further accelerate the research towards SVDD. Towards controllability, we manage the entire synthesis pipeline end-to-end, including specific details about the source and target datasets and the exact deepfake generation method. Towards diversity, we include 7 SVS and 7 SVC methods to generate 188,486 (260.34 hours) deepfake song clips against 32,312 (47.64 hours) bonafide song clips for 164 singers, with an average length of 5.02 seconds. Towards data openness, our dataset is fully accessible under a CC BY-NC-ND 4.0 license. The bonafide song clips are from open-source singing datasets, while the deepfake clips include generation results from open-sourced SVS and SVC methods and those from a collaborating company, which allows us to distribute the data under the abovementioned license.

With the CtrSVDD dataset, we also present baseline systems for CtrSVDD with flexible front-end modules (encoding waveforms into feature representations) and a fixed, robust back-end module (making predictions). Using this baseline, we explored the impact of front-end features by comparing raw waveform, spectrogram-based, and cepstral coefficients (CC)-based features. The CtrSVDD dataset, baseline system implementations, and trained model weights are publicly accessible.

2. CtrSVDD dataset design

The CtrSVDD dataset consists of 220,798 mono vocal clips in a total of 307.98 hours at a sample rate of 16 kHz. This section introduces the process of collecting bonafide vocal clips and generating deepfake clips. We also analyze the statistics of the resulting CtrSVDD dataset.

2.1. Details of bonafide vocals

Our bonafide singing vocals are sourced from existing open singing datasets, including Mandarin singing datasets:

¹<https://zenodo.org/records/10467648> (Train/Dev)

²<https://zenodo.org/records/10742049> (Eval)

³<https://github.com/SVDDChallenge/>

CtrSVDD2024_Baseline

⁴<https://github.com/svc-develop-team/so-vits-svc>

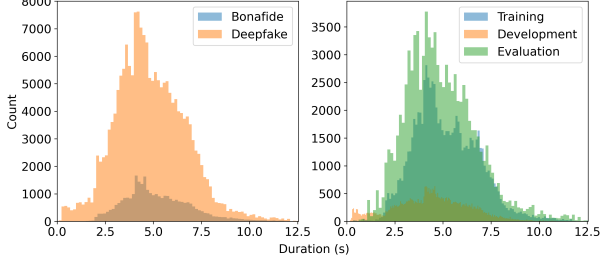


Figure 1: Histogram of audio duration. The left subfigure shows the distribution across two classes (bonafide and deepfake), whereas the right one differentiates among the train/dev/eval splits. We exclude data exceeding three standard deviations from the mean (0.6% of all data) for better visualization. All distributions are visualized with a 50% opacity then overlapped for a direct comparison between them.

Table 1: Summary of our CtrSVDD dataset.

Partition	# Speakers	Bonafide	Deepfake	
		# Utts	# Utts	Attack Types
Train	59	12,169	72,235	A01~A08
Dev	55	6,547	37,078	A01~A08
Eval	48	13,596	79,173	A09~A14

Opencpop [7], M4Singer [8], Kising [9], official ACE-Studio release [10], and Japanese singing datasets: Ofuton-P⁵, Oniku Kurumi⁶, Kiritan [11], and JVS-MuSiC [12]. We use the official temporal segmentation in their original papers for all Mandarin datasets. For the Japanese datasets, we performed automatic segmentation at long rests when the musical score is available or based on voice activity detection⁷ otherwise. Following segmentation, the bona fide vocal clips have an average duration of 5.31 seconds, amounting to a total of 32,312 clips which together span 47.64 hours.

2.2. Details of deepfake generation methods

We incorporate 14 deepfake systems across both SVS and SVC to cover many existing architectures, offering a comprehensive evaluation landscape. To ensure reproducibility in evaluation, we predominantly selected models from open-source toolkits, trained them on publicly available singing benchmarks, and then applied them to the bonafide singing vocals, with one exception of **A14** using a commercial system.

We generate 188,486 deepfake vocal clips totaling 260.34 hours from the bonafide vocal clips, with an average length of 4.97 seconds. Following the speech anti-spoofing benchmark dataset ASVspoof2019 [13], we use the same set of synthesis methods but different singers between training and validation sets, and hold-out singers and synthesis methods for the evaluation set. Table 1 shows the detailed summary for subsets. The audio duration distribution for song clips is shown in Figure 1. An overview of source datasets and deepfake methods distribution is illustrated in Figure 2. The details of each deepfake method are described as follows:

⁵<https://sites.google.com/view/oftn-utagoedb/%E3%83%9B%E3%83%BC%E3%83%A0>

⁶<https://onikuru.info/db-download/>

⁷<https://github.com/wiseman/py-webrtcvad>

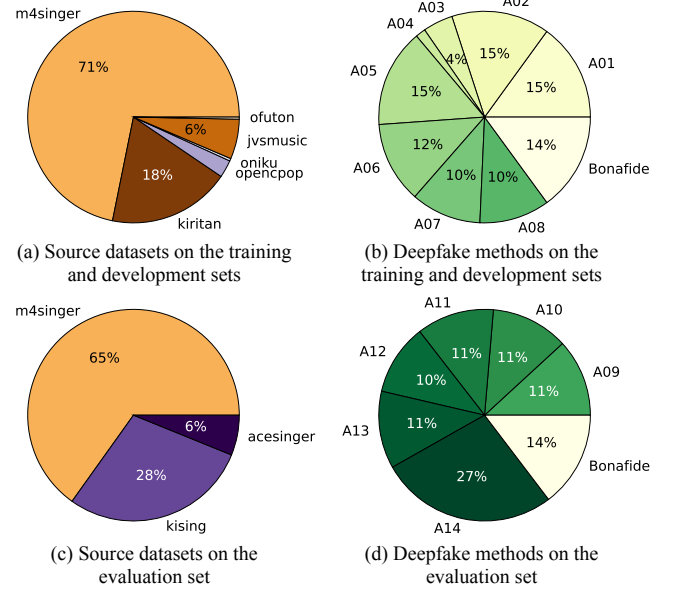


Figure 2: Overview of source datasets and deepfake methods distribution on the train/dev/eval splits of our CtrSVDD data.

2.2.1. Singing voice synthesis (SVS) systems

A01 is the non-autoregressive SVS acoustic model XiaoiceSing [1]. It employs a Transformer-based encoder-decoder architecture, similar to the FastSpeech series in the TTS domain [14, 15]. The encoder and decoder are connected through a length regulator that repeats the encoder states, considering the predicted duration information. An additional HiFi-GAN vocoder [16] is required to synthesize the waveform output.

A02 is the end-to-end SVS model VISinger that directly maps music scores to singing voices [2]. Inspired by VITS [17] in TTS, VISinger employs a variational auto-encoder (VAE) architecture combined with adversarial training.

A03 is VISinger2, an enhanced version of VISinger (A02), replacing the HiFi-GAN architecture with a differentiable digital signal processing (DDSP) vocoder [18].

A04 is NNSVS [19], an open-source software that supports various neural network (NN)-based SVS systems. We select a best-performing model that combines a diffusion-based acoustic model [20] with a source-filter HiFi-GAN vocoder [21].

A05 is Naive RNN, a non-autoregressive SVS acoustic model [22]. It utilizes bidirectional long-short-term memory (LSTM) layers to conduct a FastSpeech-like encoder-decoder modeling to convert music scores to spectral features. Similar to A01, an additional HiFi-GAN vocoder converts predicted spectral features to final singing voices.

A12 is DiffSinger, which utilizes a FastSpeech backbone by adding a diffusion-based decoder (i.e., denoiser) to generate spectral features [20]. The model training is conducted in two stages, initially optimizing the FastSpeech-based SVS and then utilizing the pre-trained encoder to continue training with the diffusion-based decoder. The output spectral features from the decoder are fed into a HiFi-GAN vocoder for decoding.

A14 is ACESinger, the singing synthesizer behind ACE-Studio [10]. The synthesized voices are further tuned manually to remove unnatural voices as detailed in [9].

We utilize the training recipes in ESPnet-Muskits [23] to train deepfake systems of **A01**, **A02**, **A03**, **A05**, and **A12** for

each database. For system **A04**, we follow the NNSVS [19] corresponding recipes to optimize the system on different datasets.

2.2.2. Singing voice conversion (SVC) systems

A06 is the Nagoya University (NU) SVC system [24], which demonstrated strong generalization capabilities and high naturalness in the SVC challenge 2023 [3]. This model utilizes a diffusion-based acoustic model [20] and a source-filter HiFi-GAN [21]. The ContentVec features [25] are employed to extract the linguistic content. The model has been trained on a large-scale dataset comprising 750 hours of publicly available speech and singing data.

A07-A11, A13 are variations of Soft-VITS-SVC, one of the major frameworks adopted in the SVC challenge 2023 [3] by utilizing the VITS framework. The approach replaces the VITS text encoder and corresponding length regulator with pre-trained acoustic features and fundamental frequency. For **A09**, a source-filter HiFi-GAN model [21] is used instead of the original HiFi-GAN model, while the source speech encoder remains the same as the original. For deepfake systems **A07, A08, A10, A11**, and **A13**, we employ different pre-trained acoustic representations as the prior input to the Soft-VITS-SVC system. Correspondingly, we utilize:

- WavLM [26] for its superior performance in SUPERB benchmark across various speech processing tasks [27, 28],
- ContentVec [25] as it is designed to reduce encoding of speaker information, which fits the SVC objective,
- MR-HuBERT [29], which is the first self-supervised learning framework considering multi-resolution information,
- WavLabLM [30] because it has considered both noise-robustness from WavLM and multilingualism, and
- Chinese HuBERT [31] as it is consistent with the major language (i.e., Mandarin) in the CtrSVDD bonafide singing.

For the training of Soft-VITS-SVC models, we use the Mandarin datasets (i.e., Opencpop [7], KiSing [9, 23], and M4Singer [8]) and then inference each singing clip by randomly selecting another singer in the subset.

2.3. Comparison with the FSD dataset

The FSD dataset [6] is the work most similar to ours, where they utilized five deepfake systems (F01-F05) in their research. We incorporate their major systems in our CtrSVDD dataset, with A09 and A12 corresponding to their F01 and F04, respectively. Our A06 is akin to their F03, employing a diffusion-based SVC system. We opted not to include equivalents to their F02 and F05 in our dataset due to their significant resemblance to other SVC systems we have integrated, specifically A08 and A09. This selection process ensures a comprehensive yet distinct representation of various methods within our dataset, avoiding redundancy while covering a broad spectrum of techniques.

3. Baseline systems

Conventional hand-crafted features, such as linear frequency cepstral coefficients (LFCC), have shown promising results in speech deepfake detection. Moreover, recent advancements in end-to-end learning approaches, such as raw waveform-based models [32], have demonstrated competitive performance. However, the effectiveness of these features in the context of singing voice deepfake detection remains largely unexplored. Therefore, we propose to systematically evaluate a diverse range of representations to gain insights into their effective-

ness and robustness in detecting singing voice deepfakes.

To this end, we design a versatile baseline framework to facilitate a fair evaluation of diverse front-end representations. The system first extracts features from interchangeable front-end modules (Section 3.1), then employs downsampling residual blocks, followed by a graph attention module (Section 3.2) to aggregate spatial and temporal information. Finally, an output layer produces a probability score reflecting the deepfake likelihood of each song clip.

3.1. Frontends

The front end refers to the pre-processing part of the network that converts raw audio samples into features, which the back-end neural network can use to make predictions.

Spectrogram. We employ a normalized power spectrogram with a 512-sample window and hop size of 160 samples.

Mel-Spectrogram. We employ a mel-filterbank with 80 bands on the spectrogram.

Mel-Frequency Cepstral Coefficients (MFCC). We extract 40 MFCC bands with spectral processing parameters similar to those used in LFCC.

Linear Frequency Cepstral Coefficients (LFCC). We employ 20 filters from 0 to 8 kHz to extract 60 coefficients from the audio signal, with type-II discrete Fourier transform (DCT) and ortho-norm for normalization, with a window length of 512 samples, and a hop length of 160 samples.

Raw waveform. We follow [32] to employ a RawNet2-style [33] learnable SincConv layer with 70 filters.

The residual blocks following front-end modules are implemented as two sequential batch normalization, SELU activation, and convolution blocks, with a residual connection between each block’s input and output; max pooling is applied before outputting. The first set of batch normalization and activation is dropped for the first residual block. We employ four residual blocks for all spectral and cepstral features; for the raw waveform feature, we employ six residual blocks to match [32]. The first two residual blocks have 32 filters, and the remaining ones have 64 filters. A linear layer is then used to connect it to the backend.

3.2. Backend

We follow [32] in our backend implementation, which consists of fully connected graph attention networks for spectral and temporal domains, then combined into a spatial-temporal graph and processed using heterogeneous stacking graph attention layers and four graph pooling layers. We connect the readout from graph pooling to a single neuron output with a linear layer. The logits output by the network (before activation) indicates the likelihood that a given song clip is bonafide.

4. Experiments and results

4.1. Experimental setup

We formulate the SVDD task as a binary classification task in alignment with the methodologies proposed by [5, 6]. SVDD models assign a continuous score to each vocal clip, with higher values indicating authentic singing and lower values suggesting deepfake ones. While a threshold is needed for model deployment in practice, using such a threshold may introduce unnecessary bias for model comparison. Instead, we employ the Equal Error Rate (EER) as the evaluation metric, which denotes the point at which the rates of false acceptances and false rejections

Table 2: Evaluation results of baseline systems on the evaluation set. Best performing results for each category are illustrated in bold.

Frontend	EER (%)	Per-method EER (%)					
		A09	A10	A11	A12	A13	A14
Spectrogram	25.50 $\pm$ 0.09	32.02 $\pm$ 0.09	14.03 $\pm$ 0.10	14.67 $\pm$ 0.08	35.18 $\pm$ 0.31	18.10 $\pm$ 0.10	28.55 $\pm$ 0.17
Mel-Spectrogram	25.19 $\pm$ 0.10	25.29 $\pm$ 0.14	15.95 $\pm$ 0.15	37.31 $\pm$ 0.09	29.28 $\pm$ 0.25	12.86 $\pm$ 0.08	27.54 $\pm$ 0.11
MFCC	26.67 $\pm$ 0.07	6.87 $\pm$ 0.10	2.50 $\pm$ 0.07	4.18 $\pm$ 0.06	45.57 $\pm$ 0.11	3.28 $\pm$ 0.04	42.98 $\pm$ 0.08
LFCC	16.15 $\pm$ 0.06	5.35 $\pm$ 0.07	2.92 $\pm$ 0.04	5.84 $\pm$ 0.07	29.47 $\pm$ 0.06	3.65 $\pm$ 0.05	24.00 $\pm$ 0.10
Raw Waveform	13.75 $\pm$ 0.11	6.72 $\pm$ 0.06	0.96 $\pm$ 0.05	3.59 $\pm$ 0.06	26.83 $\pm$ 0.10	0.95 $\pm$ 0.04	19.03 $\pm$ 0.12

are equal. This metric, distinct from accuracy, is not influenced by the choice of threshold, making it particularly apt for evaluating the performance of SVDD systems. A lower EER is indicative of a system’s superior performance.

We consistently apply a fixed random seed across all systems, utilizing the Adam optimizer, a batch size of 24, a learning rate of $1e-3$, and a weight decay of $1e-9$. Additionally, we employ a cosine annealing learning rate schedule that cycles to $1e-6$ every 10 epochs. We use binary focal loss [34], a generalized version of the binary cross-entropy loss, with focusing parameter (γ) as 2 and positive example weight (α) as 0.25. To ensure uniformity in input length, each song clip is either randomly cropped or extended to 4 seconds for batch formation during training, validation, and evaluation phases. Every system is trained for 100 epochs, after which the model checkpoint with the lowest validation EER is selected for evaluation.

In evaluation, we apply 5 different random seeds to trim vocal clips, creating 5 variations of the test set. We report the mean and standard deviation of the EER across these versions to assess model robustness against random time shifts. All experiments are performed on a single RTX 4090 GPU. The training time for the raw-waveform-based model is slightly longer than 24 hours, while the spectrogram-based model trains for around 7 hours and all other frontend features for about 5 hours.

4.2. Results and discussions

Table 2 presents the system performance results. The small standard deviation observed across all EERs suggests consistent and stable predictions across random window shifts within each song clip, lending statistical significance to comparisons among the baseline systems.

Overall EERs. Amongst all frontends for our baseline systems, the raw-waveform-based system achieves the lowest overall EER, closely succeeded by the LFCC-based system. Systems based on spectrograms, mel-spectrograms, and MFCCs exhibit comparable overall performances, trailing behind raw-waveform-based and LFCC-based systems by a large margin.

Per-method EERs. We observe that the performance gap between the top two performing systems and the remaining methods is notably large in **A09-A11** and **A13**, which are variations of the Soft-VITS-SVC with different text encoders. This suggests that the top-performing frontends generalize against unseen content encoding methods better. Conversely, all systems perform much weaker for **A12** and **A14**. **A14**, as a commercial black-box system, has an undisclosed architecture, whereas **A12** employs a diffusion-based decoder on top of the FastSpeech backbone, which is distinct from other deepfake generation approaches. We speculate that the divergence of these methods from the training distribution might prevent SVDD systems from effectively distinguishing them from bonafide singing, indicating a challenge in learning discriminative representations for these unique deepfake techniques.

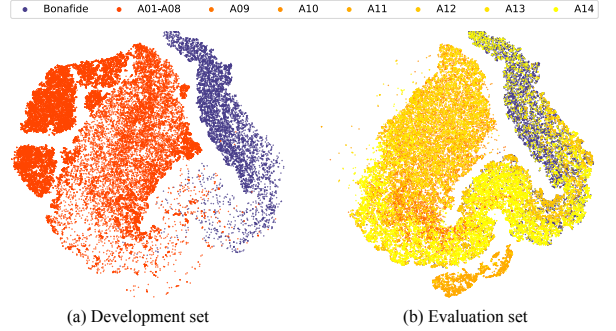


Figure 3: *t*-SNE visualization of the learned representation for the raw-waveform-based baseline system on both development and evaluation sets. Best viewed in color.

To test this hypothesis, we visualize the learned representation before the final linear layer of the raw-waveform frontend using *t*-SNE [35] in Figure 3. The visualizations on the development and evaluation sets use the same coordinate system.

As depicted in Figure 3, the distribution of bonafide singing remains consistent across both datasets. However, the distributions of deepfake singing, especially for methods **A12** and **A14**, exhibit significant overlap with that of bonafide singing. This overlap explains the reduced performance observed with these methods. It suggests that although top-performing systems can differentiate between deepfake and bonafide singing when the deepfake characteristics are similar to those encountered during training (**A09-A11** and **A13**), they struggle to accurately represent deepfake methods that deviate further from the training distribution. This underscores the need for research on SVDD systems with improved generalization ability to unseen deepfake techniques, as these techniques are rapidly advancing.

5. Conclusion

We present CtrSVDD, a dataset for controlled singing voice deepfake detection. CtrSVDD addresses key limitations in existing SVDD datasets by providing enhanced controllability, diversity, and data openness, comprising a large-scale collection of 220,798 vocal clips totaling 307.98 hours. To facilitate SVDD research using CtrSVDD, we also presented a versatile baseline system that allows for interchangeable front-end feature extraction modules. Our experiments demonstrated the importance of feature selection, with raw waveform and LFCC front-ends exhibiting the most robust performance. However, the results also highlighted a lack of generalization towards unseen deepfake methods, underscoring the need for more generalizable SVDD systems. By releasing CtrSVDD, baseline implementations, and pre-trained model weights, we aim to accelerate research for the SVDD task.

6. Acknowledgments

This work is supported in part by a New York State Center of Excellence in Data Science award, National Institute of Justice (NIJ) Graduate Research Fellowship Award 15PNJ-23-GG-01933-RESS, National Science Foundation (NSF) grants 1846184 and 2222129, synergistic activities funded by NSF grant DGE-1922591, and JST CREST JPMJCR19A3, Japan.

7. References

- [1] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, “XiaoiceSing: A high-quality and integrated singing voice synthesis system,” in *Proc. Interspeech*, 2020, pp. 1306–1310.
- [2] Y. Zhang, J. Cong, H. Xue, L. Xie, P. Zhu, and M. Bi, “VISinger: Variational inference with adversarial learning for end-to-end singing voice synthesis,” in *Proc. IEEE ICASSP*, 2022, pp. 7237–7241.
- [3] W.-C. Huang, L. P. Violeta, S. Liu, J. Shi, and T. Toda, “The singing voice conversion challenge 2023,” in *Proc. IEEE ASRU*, 2023, pp. 1–8.
- [4] N. Collins and M. Grierson, “Avoiding an AI-imposed taylor’s version of all music history,” *arXiv preprint arXiv:2402.14589*, 2024.
- [5] Y. Zang, Y. Zhang, M. Heydari, and Z. Duan, “SingFake: Singing voice deepfake detection,” in *Proc. IEEE ICASSP*, 2024.
- [6] Y. Xie, J. Zhou, X. Lu, Z. Jiang, Y. Yang, H. Cheng, and L. Ye, “FSD: An initial chinese dataset for fake song detection,” in *Proc. IEEE ICASSP*, 2024.
- [7] Y. Wang, X. Wang, P. Zhu, J. Wu, H. Li, H. Xue, Y. Zhang, L. Xie, and M. Bi, “Opencpop: A high-quality open source chinese popular song corpus for singing voice synthesis,” in *Proc. Interspeech*, 2022, pp. 4242–4246.
- [8] L. Zhang, R. Li, S. Wang, L. Deng, J. Liu, Y. Ren, J. He, R. Huang, J. Zhu, X. Chen, and Z. Zhao, “M4singer: A multi-style, multi-singer and musical score provided mandarin singing corpus,” in *Proc. NeurIPS (Dataset and Benchmarks Track)*, 2022.
- [9] J. Shi, Y. Lin, X. Bai, K. Zhang, Y. Wu, Y. Tang, Y. Yu, Q. Jin, and S. Watanabe, “Singing voice data scaling-up: An introduction to ace-opencpop and ace-kising,” *arXiv preprint arXiv:2401.17619*, 2024.
- [10] Timedomain, “ACE Studio.” [Online]. Available: <https://acestudio.ai/>
- [11] I. Ogawa and M. Morise, “Tohoku kiritan singing database: A singing database for statistical parametric singing synthesis using japanese pop songs,” *Acoustical Science and Technology*, vol. 42, no. 3, pp. 140–145, 2021.
- [12] H. Tamaru, S. Takamichi, N. Tanji, and H. Saruwatari, “JVS-MuSiC: Japanese multispeaker singing-voice corpus,” *arXiv preprint arXiv:2001.07044*, 2020.
- [13] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee *et al.*, “ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech,” *Computer Speech & Language*, vol. 64, p. 101114, 2020.
- [14] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “FastSpeech: Fast, robust and controllable text to speech,” *Proc. NeurIPS*, vol. 32, 2019.
- [15] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “FastSpeech 2: Fast and high-quality end-to-end text to speech,” in *Proc. ICLR*, 2020.
- [16] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” *Proc. NeurIPS*, vol. 33, pp. 17 022–17 033, 2020.
- [17] J. Kim, J. Kong, and J. Son, “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech,” in *Proc. ICML*. PMLR, 2021, pp. 5530–5540.
- [18] Y. Zhang, H. Xue, H. Li, L. Xie, T. Guo, R. Zhang, and C. Gong, “VISinger2: High-fidelity end-to-end singing voice synthesis enhanced by digital signal processing synthesizer,” in *Proc. Interspeech*, 2023, pp. 4444–4448.
- [19] R. Yamamoto, R. Yoneyama, and T. Toda, “NNSVS: A neural network-based singing voice synthesis toolkit,” in *Proc. IEEE ICASSP*. IEEE, 2023, pp. 1–5.
- [20] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, “DiffSinger: Singing voice synthesis via shallow diffusion mechanism,” in *Proc. AAAI*, vol. 36, no. 10, 2022, pp. 11 020–11 028.
- [21] R. Yoneyama, Y.-C. Wu, and T. Toda, “Source-filter HiFi-GAN: Fast and pitch controllable high-fidelity neural vocoder,” in *Proc. IEEE ICASSP*. IEEE, 2023, pp. 1–5.
- [22] J. Shi, S. Guo, N. Huo, Y. Zhang, and Q. Jin, “Sequence-to-sequence singing voice synthesis with perceptual entropy loss,” in *Proc. IEEE ICASSP*, 2021, pp. 76–80.
- [23] J. Shi, S. Guo, T. Qian, T. Hayashi, Y. Wu, F. Xu, X. Chang, H. Li, P. Wu, S. Watanabe, and Q. Jin, “MusKits: an end-to-end music processing toolkit for singing voice synthesis,” in *Proc. Interspeech*, 2022, pp. 4277–4281.
- [24] R. Yamamoto, R. Yoneyama, L. P. Violeta, W.-C. Huang, and T. Toda, “A comparative study of voice conversion models with large-scale speech and singing data: The T13 systems for the singing voice conversion challenge 2023,” in *Proc. IEEE ASRU*, 2023, pp. 1–6.
- [25] K. Qian, Y. Zhang, H. Gao, J. Ni, C.-I. Lai, D. Cox, M. Hasegawa-Johnson, and S. Chang, “Contentvec: An improved self-supervised speech representation by disentangling speakers,” in *Proc. ICML*. PMLR, 2022, pp. 18 003–18 017.
- [26] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, “WavLM: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [27] S.-W. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, “SUPERB: Speech processing universal performance benchmark,” in *Proc. Interspeech*, 2021, pp. 1194–1198.
- [28] T.-h. Feng, A. Dong, C.-F. Yeh, S.-w. Yang, T.-Q. Lin, J. Shi, K.-W. Chang, Z. Huang, H. Wu, X. Chang *et al.*, “SUPERB@SLT 2022: Challenge on generalization and efficiency of self-supervised speech representation learning,” in *Proc. IEEE SLT*, 2023, pp. 1096–1103.
- [29] J. Shi, H. Inaguma, X. Ma, I. Kulikov, and A. Sun, “Multi-resolution HuBERT: Multi-resolution speech self-supervised learning with masked unit prediction,” in *Proc. ICLR*, 2024.
- [30] W. Chen, J. Shi, B. Yan, D. Berrebbi, W. Zhang, Y. Peng, X. Chang, S. Maiti, and S. Watanabe, “Joint prediction and denoising for large-scale multilingual self-supervised learning,” in *Proc. IEEE ASRU*, 2023, pp. 1–8.
- [31] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [32] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, “AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks,” in *Proc. IEEE ICASSP*, 2022, pp. 6367–6371.
- [33] J.-w. Jung, S.-b. Kim, H.-j. Shim, J.-h. Kim, and H.-J. Yu, “Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms,” *Proc. Interspeech*, pp. 3583–3587, 2020.
- [34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proc. ICCV*, 2017, pp. 2980–2988.
- [35] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.