Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/XXXX.2022.1234567

SpoofCeleb: Speech Deepfake Detection and SASV In The Wild

Jee-weon Jung[1], Member, IEEE, Yihan Wu[2], Member, IEEE, Xin Wang[3], Member, IEEE, Ji-Hoon Kim[4], Member, IEEE, Soumi Maiti[5], Member, IEEE, Yuta Matsunaga[6], Hye-jin Shim[1], Member, IEEE, Jinchuan Tian[1], Nicholas Evans[7], Member, IEEE, Joon Son Chung[4], Member, IEEE, Wangyou Zhang[8], Seyun Um[9], Member, IEEE, Shinnosuke Takamichi[10], Member, IEEE, Shinji Watanabe[1], Fellow, IEEE,

[1]Carnegie Mellon University, PA 15213 USA
[2]Renmin University of China, Beijing 100872 China
[3]National Institute of Informatics, Tokyo Japan
[4]Korea Advanced Institute of Science and Technology, Daejeon 34141 South Korea
[5]Meta, CA 94025 USA
[6]University of Tokyo, Tokyo 1130033 Japan
[7]EURECOM, Biot 06410 France
[8]Shanghai Jiao Tong University, Shanghai 200240 China
[9]Yonsei University, Seoul 03722 South Korea
[10]Keio University, Kanagawa 2238521 Japan

Corresponding author: Jee-weon Jung (email: jeeweonj@ieee.org).

Jee-weon Jung, Hye-jin Shim, Jinchuan Tian, and Shinji Watanabe utilized the Bridges2 system at PSC and the Delta system at NCSA through an allocation (CIS210014) from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program. This program is funded by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296. Xin Wang is supported by JST, PRESTO Grant Number JPMJPR23P9, Japan. This work was partly supported by ONR N00014-231-2086.

ABSTRACT This paper introduces SpoofCeleb, a dataset designed for Speech Deepfake Detection (SDD) and Spoofing-robust Automatic Speaker Verification (SASV), utilizing source data from real-world conditions and spoofing attacks generated by Text-To-Speech (TTS) systems also trained on the same real-world data. Robust recognition systems require speech data recorded in varied acoustic environments with different levels of noise to be trained. However, current datasets typically include clean, high-quality recordings (bona fide data) due to the requirements for TTS training; studio-quality or well-recorded read speech is typically necessary to train TTS models. Current SDD datasets also have limited usefulness for training SASV models due to insufficient speaker diversity. SpoofCeleb leverages a fully automated pipeline we developed that processes the VoxCeleb1 dataset, transforming it into a suitable form for TTS training. We subsequently train 23 contemporary TTS systems. SpoofCeleb comprises over 2.5 million utterances from 1,251 unique speakers, collected under natural, real-world conditions. The dataset includes carefully partitioned training, validation, and evaluation sets with well-controlled experimental protocols. We present the baseline results for both SDD and SASV tasks. All data, protocols, and baselines are publicly available at https://jungjee.github.io/spoofceleb.

INDEX TERMS Speech deepfake detection, spoofing-robust automatic speaker verification, in the wild

I. INTRODUCTION

THE quality of synthetic speech has improved rapidly, driven by advancements in technologies such as flow matching, neural codecs, and speech-language modeling [1]–[3]. These innovations have significantly enhanced the naturalness and intelligibility of generated speech. The increasing availability of open sources and APIs for Text-To-Speech (TTS) systems has made high-quality synthetic speech more accessible to the general public [4], [5].

Although originally developed for positive applications, this technology is increasingly being exploited for malicious purposes [6], [7]. Synthetic speech generated with harmful intent, often referred to as spoofing, is being used to deceive individuals in scenarios such as voice phishing (or vishing). Spoofing also undermines the reliability of speech biometric

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

systems, including Automatic Speaker Verification (ASV), many of which remain highly vulnerable to such attacks [8], [9].

In response to these challenges, several datasets have been developed to advance research in Speech Deepfake Detection (SDD) [10]–[12]. For robust recognition systems, it is essential to have training data that cover a wide range of real-world acoustic environments and speaker diversity. However, speech generation systems, such as TTS and Voice Conversion (VC), typically require studio-quality or clean, read speech for training. Therefore, current datasets tend to feature clean, monotonic bona fide speech, with spoofed samples also being clean, as they are synthesized using TTS and VC systems trained on such data. The emerging task of Spoofing-robust Automatic Speaker Verification (SASV) [13] lacks dedicated datasets. Many SDD datasets also suffer from limited speaker diversity, which hinders research on SASV systems that require training with data from hundreds or even thousands of speakers.

To this end, we introduce SpoofCeleb, a dataset built upon VoxCeleb1 [14], a widely used ASV dataset consisting of the voices of 1,251 celebrities recorded under real-world conditions. We also develop a fully automated pipeline that processes VoxCeleb1 to produce in-the-wild bona fide speech samples that can be used for training TTS systems.[1] From the two available TTS training sets in TITW, we use TITW-Easy as the source dataset to generate 23 spoofing attacks. SpoofCeleb is the first dataset explicitly designed for both SDD and SASV, where the bona fide speech is real-world, noisy speech. The dataset is divided into three subsets for training, validation, and evaluation, accompanied by evaluation protocols. Baseline systems trained on SpoofCeleb's training set are also presented, demonstrating SpoofCeleb's effectiveness in and potential for future research in SDD and SASV.

II. RELATED WORKS

Datasets for SDD and the generation-recognition trade-off. To safeguard the authenticity of speech, several datasets have been published to support research in SDD [9]–[12], [18], [21], [23], [26]. One of the most critical decisions when creating these datasets is the selection of the source data (i.e., bona fide speech). This decision involves a trade-off, which we refer to as the "*generation-recognition trade-off*."

For both SDD and SASV on the recognition side, incorporating data with diverse noise, reverberation, and varied domains is essential for training robust models. It is well known that recognition models trained solely on clean speech often struggle to effectively generalize to noisy environments during inference [18]. While data augmentation techniques can help mitigate this issue [31], the most effective solution is to use training data drawn from a wide range of real-world sources.

TABLE 1. List of datasets in Speech Deepfake Detection (SDD) and Spoofing-robust Automatic Speaker Verification (SASV). FakeAVCeleb [16] and "In The Wild [17]" also have in-the-wild data. However, they have either only an evaluation set or the number of speakers or spoofing attacks is limited.

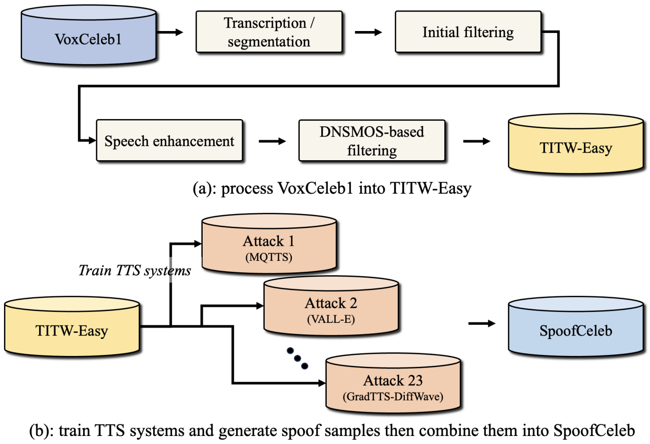
Dataset	# Spk	# Utt	# Attacks	Domain
SAS [10]	106	652,615	9	studio-recorded
ASVspoof2015 [11]	106	263,151	10	studio-recorded
Noisy Datsbase [18]	106	263,151	10	studio-recorded
ASVspoof2019 LA [9]	107	121,461	19	studio-recorded
ASVspoof2021 LA [19]	67	164,612	13	studio-recorded
ASVspoof2021 DF [19]	93	593,253	100+	studio-recorded
Voc.v [20]	21	82,048	8	studio-recorded
PartialSpoof [21]	107	121,461	19	studio-recorded
WaveFake [22]	2	136,085	9	studio-recorded
ADD 2022 [12]	N/R	493,123	N/R	studio-recorded
ADD 2023 [23]	N/R	517,068	N/R	studio-recorded
HAD [24]	218	160,836	2	studio-recorded
CFAD [25]	1302	347,400	12	studio-recorded
ASVspoof5 [26]	1,922	1,004,081	32	audiobook
MLAAD [27]	N/R	76,000	54	mixed
FMFCC-A [28]	131	50,000	13	N/R
FoR [29]	N/R	195,541	7	in the wild
FakeAVCeleb [16]	500	11,857	1	in the wild
In-The-Wild [17]	58	31,779	N/R	in the wild
VSASV [30]	1,382	338,000	3	mixed
SpoofCeleb	**1,251**	**2,687,292**	**23**	**in the wild**

Conversely, traditional TTS training requires a carefully curated and recorded dataset. Sentence prompts must be selected to ensure comprehensive phonetic coverage [32], and recordings are typically made by voice professionals in clean environments, ideally in a single anechoic studio. These recordings are of high studio quality and carefully articulated but are not scalable. For instance, the well-known CMU Arctic database includes recordings from fewer than 10 voice professionals, each reading approximately 1,000 speech prompts [32]. Modern TTS systems, however, often require significantly more training data. Instead of relying on these small-scale, TTS-specific databases, contemporary models frequently use audiobook datasets (e.g., MLS [33]), which, while not studio-grade, consist of relatively clean audiobook recordings made by numerous readers in their homes or offices.

Current SDD datasets tend to lean towards the generation side of the generation-recognition trade-off. They use source datasets that consist of either studio-quality or high-quality speech, facilitating the training of TTS and VC systems and the successful generation of spoofed speech samples. However, both the bona fide and spoofed speech in these datasets are exceedingly clean, making them far from real-world, noisy speech data.

SpoofCeleb is the first dataset to use real-world, noisy, and reverberant data originating from TITW, which originates from VoxCeleb1, as the source for training and synthesizing

[1] The development of this pipeline is extensive, and the resulting bona fide speech data can serve other purposes, such as advancing research on TTS systems trained on noisy, in-the-wild data. We detail this aspect in a separate work, referring to the dataset as TTS In The Wild (TITW) [15].



(a): process VoxCeleb1 into TITW-Easy

(b): train TTS systems and generate spoof samples then combine them into SpoofCeleb

FIGURE 1. Overall process pipeline of SpoofCeleb dataset collection. (a): our proposed fully automated pipeline transcribes, segments, filters, enhances, and again filters with DNSMOS to derive TITW-Easy [15] from VoxCeleb1 [14], which is adequate for TTS training. (b): 23 different TTS systems are trained using TITW-Easy and spoof speech samples are generated. All generated spoofing samples are combined with TITW-Easy to constitute SpoofCeleb.

spoofed speech. We tackle the generation-recognition trade-off by using our carefully curated, fully automated pre-processing pipeline that enables TTS models to be trained on data that more closely mirrors real-world conditions.

Datasets for SASV. As SASV is an emerging task extending the scope of ASV systems with spoofing robustness, there is a lack of dedicated datasets for SASV. Earlier studies on SASV have relied on SDD datasets [34], [35]. However, current SDD datasets do not prioritize speaker diversity and balance, both of which are critical for SASV. Most datasets also lack a sufficient number of speakers.

To the best of our knowledge, VSASV [30], a parallel data collection effort to SpoofCeleb, is the only attempt at addressing these limitations by creating a dataset specifically for SASV. SpoofCeleb complements VSASV while also having several distinctions. While VSASV includes three spoofing attacks, SpoofCeleb contains 23. Although VSASV uses in-the-wild bona fide data, its spoofed data are derived from high-quality sources due to the challenges in developing TTS systems with in-the-wild data. In contrast, SpoofCeleb adopts TITW which originates from VoxCeleb1, a widely-used ASV dataset recorded in the wild, as its bona fide source. Additionally, VSASV includes approximately 300 k samples, whereas SpoofCeleb offers over 2.5 M samples. Table 1 compares SpoofCeleb with other SDD and SASV datasets.

III. SOURCE DATASET: TITW

Our goal is to create a dataset for SDD and SASV using VoxCeleb1 as the source so that both bona fide and spoofed samples would reflect real-world scenarios. However, VoxCeleb1 is not suitable for direct use in TTS training.[2] The challenges with VoxCeleb1 are multifaceted. For example, the speech samples often (i) contain overly emotional expressions, (ii) include extended non-speech segments, or (iii) have excessively long durations. To address these issues, our developed fully automated pipeline processes VoxCeleb1 into the TITW dataset, which can be used for TTS training.

Figure 1-(a) illustrates the automated processing pipeline that was used to generate the TITW dataset. The pipeline begins by transcribing and obtaining word-level alignment using the WhisperX toolkit [36]. This toolkit transcribes the speech using the pre-trained Whisper Large v2 Automatic Speech Recognition (ASR) model [37], while word-level segmentation is derived from another phoneme-based ASR model. For a small subset of randomly selected samples, we also transcribe the text using the OWSMv3.1 model [38] and cross-check the accuracy of the transcriptions. We then segment the utterances from VoxCeleb1 whenever a silence longer than 500 ms is detected, resulting in multiple segments from a single utterance. Next, we apply a series of heuristic-driven rules – developed through several iterations of TTS training – to filter the data. We discarded any samples that (i) were non-English, (ii) were shorter than 1 second or longer than 8 seconds, (iii) contained one or more words with a duration exceeding 500ms, or (iv) had empty transcriptions.

After completing the initial processing steps (referred to as TITW-Hard in [15]), we conducted multiple iterations of TTS training trials. Despite these efforts, training remained extremely challenging for most TTS systems, with only a few recent models showing success. The generated speech was still insufficient to deceive pre-trained ASV systems, as measured using the SPooF Equal Error Rate (SPF-EER) metric [13].[3] To address this, we applied speech enhancement using a pre-trained model named DEMUCS and excluded samples with DNSMOS "BAK" (background noisy quality) scores below 3.0. The final number of speech segments (TITW-Easy in [15]) is approximately 248 k, which serves as the bona fide portion of the SpoofCeleb dataset. For full details on the preparation of TITW from VoxCeleb1, refer to [15]. Nonetheless, we note that this choice of enhancing the bona fide speech may confuse the training of detection models because inevitable artifacts can be added with the enhancement process. Yet, we employ TITW-Easy as the bona fide, not TITW-Hard, because of the aforementioned practicality.

IV. SPOOFCELEB

Figure 1-(b) illustrates the composition of SpoofCeleb. The TITW dataset serves as the foundation for training multiple TTS systems. These systems are then used to synthesize spoofed speech samples, which are combined with the bona fide speech samples from TITW to form the complete

[2] Our preliminary attempts to train TTS systems using the raw VoxCeleb1 data without further processing were unsuccessful.

[3] The SPF-EER is calculated by assessing an ASV system's ability to correctly accept target trials while rejecting spoofed non-target trials. Bona fide non-target trials are excluded from this protocol, as the focus is solely on evaluating the ASV system's spoofing robustness.

SpoofCeleb. To achieve this, we use 4 acoustic models, 6 waveform models (i.e., vocoders), and 5 End-to-End (E2E) models. Unless mentioned otherwise, all models were trained from scratch using the TITW-Easy data. SpoofCeleb does not include voice conversion systems, as TTS systems pose more immediate and prevalent security threats with publicly available APIs. Incorporating voice conversion systems would also require more complex configurations, such as defining source and target speaker pairs. Hence we leave this part for future work.

A. Acoustic models

Training acoustic models using in-the-wild data was one of the most challenging aspects of SpoofCeleb creation. We applied several criteria to evaluate the success of the training, including (but not limited to) speech intelligibility, measured by the Word Error Rate (WER), noisiness, assessed using DNSMOS, and speaker identity, evaluated using SPF-EER. Among these metrics, SPF-EER was prioritized as the primary measure, since the most critical factor in a spoofing attack is whether it can deceive an ASV system. The final models that were successfully trained include TransformerTTS, GradTTS, Matcha-TTS, and BVAE-TTS.

TransformerTTS [39] is an autoregressive TTS model that generates mel-spectrograms from textual input using a transformer-based architecture. The model uses a sequence of transformer encoder and decoder blocks with multi-head self-attention. We trained TransformerTTS using the ESPnet toolkit [40].[4]

GradTTS. [41] is a TTS model with a score-based decoder that generates mel-spectrograms by gradually transforming noise predicted by the text encoder. During inference, we set the denoise step to 50 to ensure high-quality speech generation. We used the official implementation and followed the default settings.[5]

Matcha-TTS. [42] is an efficient non-autoregressive TTS model based on an optimal-transport conditional flow matching decoder [1]. Unlike score-based models, it constructs a more direct sampling trajectory, enabling high-quality generation with fewer sampling steps. We used the official implementation.[6]

BVAE-TTS. [43] uses a Bidirectional-inference Variational AutoEncoder (BVAE) to model the hierarchical relationships between text and speech. By leveraging the attention maps generated using BVAE-TTS, the model jointly trains a duration predictor, enabling robust and efficient non-autoregressive speech generation. We used the official implementation.[7]

B. Waveform models

The training of waveform models was comparatively straightforward. We employed a mix of both classic and recent waveform models, including DiffWave, HiFiGAN, Parallel WaveGAN, Neural source-filter model with HiFi-GAN discriminators (NSF-HiFiGAN), BigVGAN, and WaveGlows.

DiffWave. [44] is a diffusion probabilistic model designed for both conditional and unconditional waveform generation. We used the official implementation.[8]

HiFiGAN. [45] is a widely known GAN-based waveform model that uses multiple transposed convolution blocks to progressively upsample and transform input mel-spectrograms into speech waveforms. The generator is optimized using multiple discriminator losses, a feature matching loss, and L1 loss between the generated and ground truth mel-spectrograms. We used the HiFiGAN V1 architecture from the official implementation.[9]

Parallel WaveGAN. [46] is a lightweight vocoder model. It uses a non-autoregressive WaveNet [47] architecture combined with multi-resolution Short-Time Fourier Transform (STFT) loss and waveform adversarial loss. We used the official implementation.[10]

NSF HiFiGAN. [48] is similar to Parallel WaveGAN but explicitly incorporates a sine-based source signal as input to the generator. It also includes a noise branch that transforms random noise into an aperiodic signal. This aperiodic signal is combined with the generator's periodic output for harmonic-plus-noise speech waveform generation. We used the official implementation.[11]

BigVGAN. [49] is a universal GAN-based vocoder that generalizes effectively across diverse scenarios, including unseen speakers, languages, and recording environments. By using periodic activation functions and anti-aliased representations, BigVGAN introduces a beneficial inductive bias for speech synthesis. We used the official implementation[12]

WaveGlow. [50] generates waveforms through a series of neural network-based invertible affine transformations conditioned on input mel-spectrograms. During training, the model parameters are optimized to whiten the ground-truth waveform as much as possible. We used the same toolkit as with NSF HiFiGAN.

C. E2E and speech-language models with neural codecs

While two-stage TTS pipelines have proven effective for modeling speech from text, they often suffer from poor quality due to the mismatch between acoustic and waveform models. Waveform models are trained on predefined features

[4] https://github.com/ESPnet/ESPnet.

[5] https://github.com/huawei-noah/Speech-Backbones.

[6] https://github.com/shivammehta25/Matcha-TTS.

[7] https://github.com/LEEYOONHYUNG/BVAE-TTS.

[8] https://github.com/lmnt-com/diffwave.

[9] https://github.com/jik876/hifi-gan.

[10] https://github.com/kan-bayashi/ParallelWaveGAN.

[11] https://github.com/nii-yamagishilab/project-NN-Pytorch-scripts.

[12] https://github.com/NVIDIA/BigVGAN.

but must process the outputs generated by acoustic models during inference, leading to potential inconsistencies. To address this issue, several E2E models have been proposed, and we have successfully trained multiple E2E models using the TITW dataset.

Speech-Language Models (SpeechLMs) represent an emerging category of TTS models. Similar to language models in natural language processing, they are trained to predict tokens, in this case, tokens of neural codecs, which are then decoded via a neural codec system's decoder. Unlike acoustic models, which can be paired with any compatible waveform model, SpeechLMs rely on a predetermined decoder based on the neural codec used during training, limiting their ability to function with multiple decoders.

VALL-E, Multi-Scale Transformer, and Delay. VALL-E [2] predicts the first token of each frame using an autoregressive module, followed by a non-autoregressive prediction for the remaining tokens. Multi-Scale Transformer [51] uses a global Transformer for inter-frame modeling and a local Transformer for intra-frame modeling, maintaining full autoregression without approximation. In Delay [52], the multi-stream token sequences are processed using a "delay" interleave pattern, which enables approximate autoregressive prediction for both inter- and intra-frame modeling, achieving high efficiency. We used implementations of the three models in the ESPnet toolkit.[4]

MQTTS. [3] is designed to synthesize speech using real-world data from YouTube and podcasts. To address misalignments common in mel-spectrogram-based autoregressive models, it uses a multi-codebook vector quantization approach to improve both speech intelligibility and diversity. MQTTS aligns closely with the goals of this work, as we aim to develop a dataset that spans real-world data for both bona fide and spoofed speech. We used the official implementation.[13]

VITS. [53] is an E2E TTS model that combines a conditional VAE with stochastic duration prediction to generate waveforms from textual input. The model uses normalizing flow to learn latent representations from speech, while the stochastic duration predictor captures diverse speech prosody from text. For waveform generation, adversarial loss is used to produce high-quality waveforms from the latent representations. We trained VITS using the ESPnet toolkit.[4]

D. Attack generation, partitioning, and protocols

Diverse combinations of acoustic and waveform models, alongside E2E and SpeechLM models, result in a total of 23 spoofing attacks. This approach is inspired by previous research, which demonstrated that both acoustic and waveform models impact the perceptual quality of synthesized speech [54]. Table 3 provides a detailed overview of the 23 spoofing attacks included in SpoofCeleb.

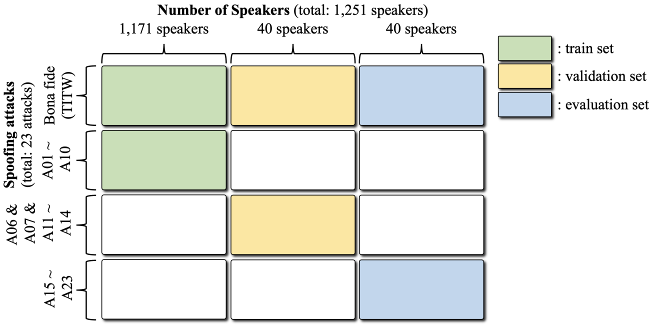


FIGURE 2. **Illustration of how SpoofCeleb is partitioned.**

Data partitioning for SpoofCeleb requires a more sophisticated approach compared to existing ASV or SDD datasets. An SDD dataset only requires the binary bona fide or spoof label, while an ASV dataset focuses on speaker identities. SpoofCeleb, as a dataset for both SDD and SASV, must account for both bona fide/spoof labels and speaker identities simultaneously.

Speakers. For the speaker partitioning, we divide the 1,251 speakers in the bona fide data into three sets: $1,171$ for training, 40 for validation, and 40 for evaluation. This ensures that there are no overlapping speakers between any of the sets.

Spoofing attacks. For spoofing attacks, we divide the bona fide data (A00) and the 23 spoofing attacks (A01–A23) as follows. In the training set, 10 attacks (A01 to A10) are combined with the bona fide data. Among these attacks, six are derived from a combination of acoustic and waveform models, while the remaining four originate from E2E and SpeechLM TTS systems.

In the validation set, there are 6 attacks: A06, A07, and A11 to A14, combined with the bona fide data (A00). Attacks A06 and A07 represent known attacks from unknown speakers. Attacks A11 and A12 involve the same architecture as other attacks but differ in model training details. Specifically, A11 is fully trained from scratch using the TITW dataset, while in A02, the decoder was pre-trained. Similarly, A12 is fully trained from scratch on TITW, whereas A04 was pre-trained on LibriSpeechGigaSpeech [55] and the English subset of Multilingual LibriSpeech [33], then fine-tuned on TITW. Attacks A13 and A14 serve as partially known attacks. In A13, the acoustic model (GradTTS) is known, but the waveform model (NSF HiFiGAN) is unknown. Similarly, in A14, the acoustic model (Matcha-TTS) is known, but the waveform model (HiFiGAN) is unknown.

In the evaluation set, there are 9 attacks, A15 to A23. Attacks A15 and A16 involve known architectures but differ in configurations. For A15, the decoder is initialized with a pre-trained model, and the speaker embeddings are taken from target utterances, simulating a scenario in which an attacker has access to the target speaker's utterance pool.

[13] https://github.com/b04901014/MQTTS.

TABLE 2. Number of speech files and protocols. Number of trials equals that of speech files for SDD protocols.

	# speech files	# trials in SASV protocols
Train	$2,540,421$	N/A
Validation	$55,741$	$39,353$
Evaluation	$91,130$	$133,448$

TABLE 3. Spoofing attacks of SpoofCeleb. There are 23 different attacks stemming from 23 different TTS systems. †: pre-trained, ‡: decoder is pre-trained, $^{\diamond}$: speaker embeddings from target utterances.

AttackID	Partition	Acoustic model	Waveform model
A01	trn	VITS	N/A
A02	trn	MQTTS‡	N/A
A03	trn	VALL-E	N/A
A04	trn	Delay†	N/A
A05	trn	GradTTS	DiffWave
A06	trn&dev	GradTTS	BigVGAN
A07	trn&dev	GradTTS	WaveGlow
A08	trn	MatchaTTS	DiffWave
A09	trn	MatchaTTS	BigVGAN
A10	trn	MatchaTTS	WaveGlow
A11	dev	MQTTS	N/A
A12	dev	Delay	N/A
A13	dev	GradTTS	NSF HiFiGAN
A14	dev	MatchaTTS	HiFiGAN
A15	eval	MQTTS$^{\ddagger,\diamond}$	N/A
A16	eval	VALL-E†	N/A
A17	eval	GradTTS	HiFiGAN
A18	eval	MatchaTTS	NSF HiFiGAN
A19	eval	Multi-scale Transformer	N/A
A20	eval	Multi-scale Transformer†	N/A
A21	eval	TransformerTTS	ParallelWaveGAN
A22	eval	BVAE-TTS	HiFiGAN
A23	eval	BVAE-TTS	NSF HiFiGAN

A16 was pre-trained using the same data composition as A04. Attacks A17 and A18 represent partially known attacks where the acoustic models are known, but the waveform models are not. Finally, A19 to A23 are fully unknown attacks, meaning no part of their models was encountered during training.

Figure 2 illustrates the three partitions of SpoofCeleb and Table 2 provides the statistics of each partition. In total, SpoofCeleb contains over 2.5 M speech samples.

Protocols. SpoofCeleb includes protocols for validating and evaluating developed SDD and SASV models. The SDD protocols for validation and evaluation specify the speech samples to be assessed, while the SASV protocols list pairs of trials with an enrollment utterance and a test utterance. Table 2 provides details on the number of utterances for the SDD protocols and the number of trials for the SASV protocols.

V. Baselines

A. SDD

Two E2E SDD models, RawNet2 [56] and AASIST [57], are used as the baselines. The RawNet2 model for SDD is an adapted version of RawNet2 originally designed for ASV. It features an input layer that processes raw waveforms directly and uses convolution-based residual blocks. Frame-level representations are aggregated, projected, then passed through a binary classification head.

AASIST is one of the most widely used SDD models in recent literature. Like RawNet2, it includes an input layer that processes raw waveforms and uses convolution-based residual blocks. However, unlike RawNet2, AASIST incorporates graph attention network-based modules designed to capture spectral and temporal spoofing artifacts separately. It then uses heterogeneous stacking of graph attention layers to jointly model spectral and temporal information concurrently.

B. SASV

We employ three models as SASV baselines, all of which use the SKA-TDNN architecture [58]. These models are used to assess the impact of different training data and scenarios. SKA-TDNN is a convolution-based model with residual connections, incorporating dedicated modules and architectural design choices for multi-scale processing. It is an advanced version of the ECAPA-TDNN architecture [59].

Among the three SASV baselines, the first model ("Conventional ASV") is trained as a conventional ASV system using the VoxCeleb1&2 datasets, without considering spoof robustness. We use a pre-trained model from ESPnet-SPK [60]. The second model ("SASV trained on out-of-domain data") is trained as an SASV model but uses out-of-domain data from the ASVspoof2019 logical access dataset [9]. We use a pre-trained model from [61]. The third model ("SASV trained on SpoofCeleb") is trained as an SASV model using the training set from SpoofCeleb.

VI. Metrics

A diverse set of metrics is employed to evaluate the SpoofCeleb dataset, as well as the SDD and SASV models. To assess the quality of the speech samples and the strength of the attacks, we use SPF-EER, Mean Cepstral Distortion (MCD), UTMOS [62], DNSMOS [63], and Word Error Rate (WER), with the WER evaluated using the OpenAI Whisper-Large model [64]. SPF-EER measures speaker characteristics, UTMOS and DNSMOS are objective approximations of perceived quality and noisiness of synthesized speech, and WER measures intelligibility. For evaluating the performances of the SDD baselines, we use Equal Error Rate (EER) and the min Detection Cost Function (minDCF) [65]. To assess the SASV baselines, we adopt the recently proposed architecture-agnostic Detection Cost Function (min a-DCF) [66], along with Speaker Verification EER (SV-EER) and SPooF EER (SPF-EER). Table 5 outlines the trial types

TABLE 4. Quality and strength of 23 spoofing attacks included in SpoofCeleb. SPF-EER (%) measures how hard it is to reject an attack by a pre-trained ASV system. MCD, UTMOS, and DNSMOS demonstrate how noisy the attacks are and WER (%) measures the intelligibility.

Attack ID	Partition	SPF-EER (%)↓	MCD↓	UTMOS↑	DNSMOS↑	WER (%)↓
A00 (bona fide)	trn&val&eval	N/A	N/A	3.32	2.78	9.10
A01	trn	29.22	8.61	2.77	2.74	53.00
A02	trn	49.47	7.09	3.08	2.83	23.60
A03	trn	12.51	10.85	3.28	2.93	28.50
A04	trn	20.86	10.42	3.59	2.83	4.80
A05	trn	23.63	6.76	2.18	2.39	11.90
A06	trn&val	29.42	9.23	2.08	2.16	11.30
A07	trn&val	24.61	5.61	1.30	1.51	11.90
A08	trn	32.00	5.36	2.47	2.59	15.80
A09	trn	31.07	9.10	2.38	2.48	15.90
A10	trn	26.20	5.66	1.32	1.79	15.70
A11	val	47.78	6.99	3.08	2.83	23.30
A12	val	14.52	10.91	3.26	2.84	32.50
A13	val	27.13	5.52	1.97	2.13	12.90
A14	val	29.36	5.11	2.52	2.48	14.50
A15	eval	65.21	6.79	3.14	2.83	21.20
A16	eval	21.63	10.43	3.87	2.93	3.40
A17	eval	30.20	5.44	2.62	2.43	11.20
A18	eval	25.84	5.19	2.04	2.24	16.00
A19	eval	17.20	10.69	3.29	2.88	11.80
A20	eval	22.36	10.73	3.53	2.92	5.50
A21	eval	22.32	11.68	2.06	2.50	24.90
A22	eval	5.65	5.74	1.37	1.62	21.50
A23	eval	6.75	5.65	1.30	1.50	25.90

TABLE 5. Three metrics used for gauging performances of SASV baselines. a-DCF measures the overall performance. SV-EER measures the ability to reject non-target speakers and SPF-EER measures spoof-robustness. "+": a system should accept, "-": a system should reject.

Trial type \ metric	a-DCF [66]	SV-EER	SPF-EER [13]
Target	+	+	+
Bona fide non-target	-	-	
Spoof non-target	-		-

TABLE 6. SDD baseline performances.

System	Train set	Validation		Evaluation	
		EER	minDCF	EER	minDCF
RawNet2	ASVspoof2019	56.33	0.9996	58.79	0.9990
AASIST	ASVspoof2019	26.64	0.6048	23.51	0.4710
RawNet2	SpoofCeleb	8.63	0.1910	1.12	0.0290
AASIST	SpoofCeleb	0.61	0.0160	2.37	0.0328

involved in the SASV metrics; a-DCF includes all three trial types, while SV-EER and SPF-EER cover only a subset.

VII. Results

A. Spoofing attacks

Table 4 presents various metrics to assess the speech quality of the 23 synthesized spoofing attacks and how effectively they threaten ASV systems. SPF-EER is the most critical metric, as it measures the extent to which the generated attacks can deceive existing ASV systems. We evaluated SPF-EER using a pre-trained RawNet3 model [68], which is publicly available through ESPnet-SPK [60].

In the top row, the speech quality evaluations for A00 (bona fide speech) are provided as reference values. The results confirm that the spoofing attacks in SpoofCeleb are highly threatening, with most attacks achieving an SPF-EER over 20%. The majority of attacks exhibit relatively minor degradation in UTMOS and DNSMOS, indicating the high quality of the synthesized speech samples. Intelligibility, measured using the WER, shows that for most attacks, there is no more than a 10% deterioration in performance.

B. SDD

Table 6 presents the results of four baseline SDD systems. We evaluate two SDD models, RawNet2 and AASIST, trained on two different datasets. The models trained on the ASVspoof2019 logical access dataset are used to assess the zero-shot performance on validation and evaluation SDD protocols of SpoofCeleb. The other two models demonstrate the performance of systems trained on in-domain SpoofCeleb training data.

The zero-shot results in the top two rows indicate that existing SDD models not trained on in-the-wild data struggle to distinguish between spoofed samples and bona fide speech. As shown in rows 3 and 4, there is a significant

TABLE 7. Attack-wise performance of RawNet2 SDD baseline on validation and evaluation sets. Performances reported using EER (%).

System	A06	A07	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23
RawNet2 [56]	0.69	0.40	19.55	0.97	6.54	1.35	0.16	0.27	0.28	2.72	0.53	0.63	1.40	0.01	0.36

TABLE 8. SASV baseline performances. SKA-TDNN [58] model architecture is employed. Three models are trained in different scenarios. "Conventional ASV" is trained with VoxCelebs1&2 [14], [67] and "SASV trained on out-of-domain data" is trained on ASVspoof2019 logical access [9].

System	Validation			Evaluation		
	a-DCF	SV-EER	SPF-EER	a-DCF	SV-EER	SPF-EER
Conventional ASV [60]	0.4973	2.55	23.62	0.4923	3.84	23.44
SASV trained on out-of-domain data [61]	0.2901	33.67	55.01	0.9998	38.94	52.24
SASV trained on SpoofCeleb	0.3101	41.96	64.50	0.2902	12.78	5.00

performance improvement when these models are trained using the SpoofCeleb training set, highlighting the importance of training SDD models on in-the-wild data. However, the RawNet2's result in row 3 is unexpected, as it shows better performance on the evaluation set than on the validation set, while the evaluation set includes totally unknown attacks. To further investigate this, we conduct an analysis of the attack-wise results.

Table 7 presents the attack-wise performance of the RawNet2 baseline SDD model trained on the SpoofCeleb training set. Attacks A06 and A07 are classified as known attacks. Attacks A11 to A18 are partially unknown; in these cases, either the acoustic or waveform model is known, or the architecture is familiar but trained with a different configuration. Attacks A19 to A23 represent entirely unknown attacks.

We found that the inferior performance on attack A11 contributed to the validation set results being worse than those on the evaluation set. Interestingly, when comparing A11 and A15, attack A15 is more difficult to distinguish for a conventional ASV system that does not account for spoofing, with SPF-EER values of 47.78% for A11 and 65.21% for A15. Both attacks originate from MQTTS; however, A11 was trained entirely from scratch, while A15 utilized a pre-trained decoder. Once an SASV system is trained on the SpoofCeleb training data, A11 becomes more challenging to detect. A deeper investigation into the reasons behind this phenomenon is left for future work.

The comparative analysis in Tables 4 and 7 reveals a discrepancy between the rankings of attacks' SPF-EER on the pre-trained ASV system trained with VoxCeleb and the rankings of attacks' EER on the SDD system trained with SpoofCeleb. This divergence may be attributed to the differences in training data, whether the models were trained on SpoofCeleb. The discrepancy could be a result of the fundamental differences in the tasks themselves, as SDD and SASV systems are optimized for distinct objectives.

C. SASV

Table 8 presents the performances of three SASV baselines on the SpoofCeleb validation and evaluation protocols. Min a-DCF assesses the overall performance, while SV-EER and SPF-EER evaluate the systems' ability to reject bona fide and spoof non-target trials, respectively.

As expected, a conventional ASV system that does not account for spoof attacks, shown in the first row, fails to reject synthesized speech samples, with an a-DCF exceeding 0.49 on both the validation and evaluation sets. However, it performs well at rejecting bona fide non-target trials. The results in the second row indicate an improvement in a-DCF for the validation set, but even worse performance on the evaluation set. Both SV-EER and SPF-EER remain very high, indicating that the system trained for SASV with out-of-domain data struggles to reject both types of non-target trials. The a-DCF of 0.9998 also signifies that the model fails to find an operating point where it can reject both types of non-target trials. Finally, when trained on the SpoofCeleb training data, the a-DCF on the evaluation set drops to its lowest value (0.2902), and both SV-EER and SPF-EER are more balanced compared with row 1, where the system was only capable of rejecting bona fide non-target trials.

VIII. Conclusion and remarks

This paper introduces SpoofCeleb, a dataset for SDD and SASV based on in-the-wild data. To create a dataset that incorporates real-world conditions, we used a fully automated pipeline to process the VoxCeleb1 dataset, making it possible to use it for training TTS systems. We further trained 23 TTS systems, partitioning TITW and the TTS systems into SpoofCeleb, which includes training, validation, and evaluation sets. Protocols were defined to train and test both SDD and SASV models, and baseline systems for SDD and SASV were established, trained, and evaluated.

While there are numerous SDD datasets, many are limited in scale or speaker diversity, which has hindered research on single SASV models. We hope SpoofCeleb will serve as the first dataset with enough data to effectively train single SASV systems. Yet, SpoofCeleb has its limitations. In the experiments, some spoofing attacks are shown to be less challenging, as the wild nature of the TITW data complicates the training of robust TTS systems. Future work will focus on advancing TTS training techniques that can better leverage this challenging in-the-wild data.

REFERENCES

[1] Y. Lipman, R. T. Chen *et al.*, "Flow matching for generative modeling," in *Proc. ICLR*, 2023.

[2] C. Wang, S. Chen *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint:2301.02111*, 2023.

[3] L.-W. Chen, S. Watanabe, and A. Rudnicky, "A vector quantized approach for text to speech synthesis on real-world spontaneous speech," in *Proc. AAAI*, vol. 37, no. 11, 2023, pp. 12 644–12 652.

[4] https://www.resemble.ai/api/, accessed: 2024-09-16.

[5] https://voice.ai/voice-cloning, accessed: 2024-09-16.

[6] "A voice deepfake was used to scam a CEO out of 243, 000," www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/, accessed: 2024-09-16.

[7] "Fake Joe Biden robocall tells New Hampshire Democrats not to vote tuesday," https://nbcnews.com/politics/2024-election/fake-joe-biden-robocall-tells-new-hampshire-democrats-not-vote-tuesday-rcna134984, accessed: 2024-09-16.

[8] J.-w. Jung, X. Wang *et al.*, "To what extent can ASV systems naturally defend against spoofing attacks?" in *Proc. Interspeech*, 2024, pp. 3240–3244.

[9] X. Wang, J. Yamagishi *et al.*, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, p. 101114, 2020.

[10] Z. Wu, A. Khodabakhsh *et al.*, "SAS: A speaker verification spoofing database containing diverse attacks," in *Proc. IEEE ICASSP*, 2015.

[11] Z. Wu, T. Kinnunen *et al.*, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Proc. Interspeech*, 2015.

[12] J. Yi, R. Fu *et al.*, "ADD 2022: the first audio deep synthesis detection challenge," in *Proc. IEEE ICASSP*, 2022.

[13] J. W. Jung, H. Tak *et al.*, "Sasv 2022: The first spoofing-aware speaker verification challenge," in *Proc. Interspeech*, 2022.

[14] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017.

[15] J.-w. Jung, W. Zhang, S. Maiti, Y. Wu, X. Wang *et al.*, "Text-to-speech synthesis in the wild," in *arXiv preprint:2409.08711*, 2024.

[16] H. Khalid, S. Tariq *et al.*, "FakeAVCeleb: A novel audio-video multimodal deepfake dataset," in *Proc. NeurIPS Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[17] N. Müller, P. Czempin *et al.*, "Does audio deepfake detection generalize?" in *Proc. Interspeech*, 2022.

[18] X. Tian, Z. Wu *et al.*, "Spoofing detection under noisy conditions: a preliminary investigation and an initial database," *arXiv preprint:1602.02950*, 2016.

[19] J. Yamagishi, X. Wang *et al.*, "ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *Proc. Interspeech*, 2021.

[20] X. Wang and J. Yamagishi, "Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders," in *Proc. IEEE ICASSP*, 2023.

[21] L. Zhang, X. Wang *et al.*, "The partialspoof database and countermeasures for the detection of short fake speech segments embedded in an utterance," *IEEE/ACM Trans. ASLP.*, vol. 31, pp. 813–825, 2022.

[22] J. Frank and L. Schönherr, "WaveFake: A data set to facilitate audio deepfake detection," in *Proc. NeuralIPS*, 2021.

[23] J. Yi, J. Tao *et al.*, "ADD 2023: the second audio deepfake detection challenge," *arXiv preprint:2305.13774*, 2023.

[24] J. Yi, Y. Bai *et al.*, "Half-truth: A partially fake audio detection dataset," in *Proc. Interspeech*, 2021.

[25] H. Ma, J. Yi *et al.*, "CFAD: A Chinese dataset for fake audio detection," *Speech Communication*, vol. 164, p. 103122, 2024.

[26] X. Wang, H. Delgado *et al.*, "ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale," *arXiv preprint:2408.08739*, 2024.

[27] N. M. Müller, P. Kawa *et al.*, "MLAAD: The multi-language audio anti-spoofing dataset," in *Proc. IJCNN*, 2024.

[28] Z. Zhang, Y. Gu *et al.*, "FMFCC-A: A challenging mandarin dataset for synthetic speech detection," in *Proc. IWDW*, 2021.

[29] R. Reimao and V. Tzerpos, "FoR: A dataset for synthetic speech detection," in *Proc. SpeD*, 2019.

[30] V. Hoang, V. T. Pham, H. N. Xuan, N. Pham, P. Dat, and T. T. T. Nguyen, "VSASV: A Vietnamese dataset for spoofing-aware speaker verification," in *Proc. Interspeech*, 2024, pp. 4288–4292.

[31] H. Tak, M. Kamble *et al.*, "RawBoost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing," in *Proc. IEEE ICASSP*, 2022.

[32] J. Kominek and A. W. Black, "The CMU arctic speech databases," in *Proc. SSW*, 2004.

[33] V. Pratap *et al.*, "MLS: A large-scale multilingual dataset for speech research," in *Proc. Interspeech*, 2020, pp. 2757–2761.

[34] M. Todisco, H. Delgado *et al.*, "Integrated presentation attack detection and automatic speaker verification: Common features and Gaussian back-end fusion," in *Proc. Interspeech*, 2018.

[35] H.-j. Shim, J.-w. Jung *et al.*, "Integrated replay spoofing-aware text-independent speaker verification," *Applied Sciences*, vol. 10, no. 18, p. 6292, 2020.

[36] M. Bain, J. Huh *et al.*, "WhisperX: Time-accurate speech transcription of long-form audio," in *Proc. Interspeech*, 2023, pp. 4489–4493.

[37] A. Radford, J. W. Kim *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023, pp. 28 492–28 518.

[38] Y. Peng, J. Tian *et al.*, "OWSM v3.1: Better and faster open Whisper-style speech models based on e-branchformer," in *Proc. Interspeech*, 2024, pp. 352–356.

[39] N. Li, S. Liu *et al.*, "Neural speech synthesis with transformer network," in *Proc. AAAI*, vol. 33, no. 01, 2019, pp. 6706–6713.

[40] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[41] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. A. Kudinov, "Grad-TTS: A diffusion probabilistic model for text-to-speech," in *Proc. ICML*, 2021, pp. 8599–8608.

[42] S. Mehta, R. Tu *et al.*, "Matcha-TTS: A fast TTS architecture with conditional flow matching," in *Proc. IEEE ICASSP*, 2024, pp. 11 341–11 345.

[43] Y. Lee, J. Shin, and K. Jung, "Bidirectional variational inference for non-autoregressive text-to-speech," in *Proc. ICLR*, 2021.

[44] Z. Kong, W. Ping *et al.*, "Diffwave: A versatile diffusion model for audio synthesis," in *Proc. ICLR*, 2020.

[45] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeuralIPS*, 2020, pp. 17 022–17 033.

[46] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. IEEE ICASSP*, 2020, pp. 6199–6203.

[47] A. Van Den Oord, S. Dieleman *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint:1609.03499*, 2016.

[48] X. Wang, S. Takaki, and J. Yamagishi, "Neural Source-Filter Waveform Models for Statistical Parametric Speech Synthesis," *IEEE/ACM Trans. ASLP.*, vol. 28, pp. 402–415, 2020.

[49] S.-g. Lee, W. Ping *et al.*, "BigVGAN: A universal neural vocoder with large-scale training," in *Proc. ICLR*, 2023.

[50] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. IEEE ICASSP*, 2019, pp. 3617–3621.

[51] D. Yang, J. Tian *et al.*, "UniAudio: Towards universal audio generation with large language models," in *Proc. ICML*, 2024.

[52] J. Copet, F. Kreuk *et al.*, "Simple and controllable music generation," in *Proc. NeuralIPS*, vol. 36, 2023, pp. 47 704–47 720.

[53] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. ICML*, 2021, pp. 5530–5540.

[54] O. Watts, G. Eje Henter, J. Fong, and C. Valentini-Botinhao, "Where do the improvements come from in sequence-to-sequence neural tts?" in *Proc. SSW*, 2019, pp. 217–222.

[55] G. Chen *et al.*, "GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio," in *Proc. Interspeech*, 2021, pp. 3670–3674.

[56] H. Tak, J. Patino *et al.*, "End-to-end anti-spoofing with rawnet2," in *Proc. IEEE ICASSP*, 2021.

[57] J.-w. Jung, H.-S. Heo *et al.*, "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *Proc. IEEE ICASSP*, 2022.

[58] S. H. Mun, J.-w. Jung *et al.*, "Frequency and multi-scale selective kernel attention for speaker verification," in *Proc. IEEE SLT*, 2023.

[59] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Proc. Interspeech*, 2020.

[60] J.-w. Jung, W. Zhang *et al.*, "Espnet-spk: full pipeline speaker embedding toolkit with reproducible recipes, self-supervised front-ends, and off-the-shelf models," in *Proc. Interspeech*, 2024.

[61] S. H. Mun, H. J. Shim *et al.*, "Towards single integrated spoofing-aware speaker verification embeddings," in *Proc. Interspeech*, 2023.

[62] T. Saeki, D. Xin *et al.*, "UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022," in *Proc. Interspeech*, 2022, pp. 4521–4525.

[63] C. K. A. Reddy, V. Gopal, and R. Cutler, "DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors," in *Proc. IEEE ICASSP*, 2021, pp. 6493–6497.

[64] A. Radford, J. W. Kim *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023.

[65] NIST, "NIST 2016 speaker recognition evaluation plan," 2016.

[66] H.-j. Shim, J.-w. Jung *et al.*, "A-DCF: An architecture agnostic metric with application to spoofing-robust speaker verification," in *Proc. Speaker Odyssey*, 2024.

[67] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018.

[68] J.-w. Jung, Y. J. Kim *et al.*, "Pushing the limits of raw waveform speaker recognition," in *Proc. Interspeech*, 2022.