Computer Speech and Language 95 (2026) 101825

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

ASVspoof 5: Design, collection and validation of resources for spoofing, deepfake, and adversarial attack detection using crowdsourced speech

Xin Wang [a,*], Héctor Delgado [b], Hemlata Tak [c], Jee-weon Jung [d,1], Hye-jin Shim [d], Massimiliano Todisco [e], Ivan Kukanov [f], Xuechen Liu [a], Md Sahidullah [g], Tomi Kinnunen [h], Nicholas Evans [e], Kong Aik Lee [i], Junichi Yamagishi [a], Myeonghun Jeong [j], Ge Zhu [k,2], Yongyi Zang [k], You Zhang [k], Soumi Maiti [d,3], Florian Lux [l], Nicolas Müller [m], Wangyou Zhang [n], Chengzhe Sun [o], Shuwei Hou [o], Siwei Lyu [o], Sébastien Le Maguer [p], Cheng Gong [q], Hanjie Guo [r], Liping Chen [r], Vishwanath Singh [h]

[a] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
[b] Microsoft, P.º Club Deportivo, 1, Edificio 1, 28223 Pozuelo de Alarcón, Madrid, Spain
[c] Pindrop, 1115 Howell Mill Rd NW #700, 30318, Atlanta GA, USA
[d] Carnegie Mellon University, 5000 Forbes Avenue, 15213, Pittsburgh, USA
[e] EURECOM, Campus SophiaTech, 450 Route des Chappes, 06410 Biot, France
[f] KLASS Engineering and Solutions, 30A Kallang Pl, #11-03, 339213, Singapore
[g] Institute for Advancing Intelligence, TCG CREST, 700091, Kolkata, India
[h] University of Eastern Finland, Joensuu campus, Länsikatu 15, FI 80110, Joensuu, Finland
[i] The Hong Kong Polytechnic University, Kowloon, Hong Kong, China
[j] Seoul National University, 1 Gwanak-ro, Gwanak-gu, 08826, Seoul, Republic of Korea
[k] University of Rochester, 720 Computer Studies Building, 14627, Rochester, USA
[l] University of Stuttgart, Pfaffenwaldring 5 b, 70569, Stuttgart, Germany
[m] Fraunhofer AISEC, Lichtenbergstrasse 11, 85748 Garching, Germany
[n] Shanghai Jiao Tong University, 200240, Shanghai, China
[o] University at Buffalo, 14260 NY, Buffalo, USA
[p] University of Helsinki, FI 00100, Helsinki, Finland
[q] Tianjin University, No. 135 Yaguan Road, 300350, Tianjin, China
[r] University of Science and Technology of China, No. 96, JinZhai Road, 230026, Hefei, China

ARTICLE INFO

ABSTRACT

* Corresponding author.
E-mail addresses: wangxin@nii.ac.jp (X. Wang), hector.delgado@microsoft.com (H. Delgado), Hemlata.Tak@pindrop.com (H. Tak), jeeweonj@ieee.org (J.-w. Jung), hyejinsh@andrew.cmu.edu (H.-j. Shim), Massimiliano.Todisco@eurecom.fr (M. Todisco), Ivan@kukanov.com (I. Kukanov), xuecliu@nii.ac.jp (X. Liu), sahidullahmd@gmail.com (M. Sahidullah), tkinnu@cs.uef.fi (T. Kinnunen), evans@eurecom.fr (N. Evans), kong-aik.lee@polyu.edu.hk (K.A. Lee), jyamagis@nii.ac.jp (J. Yamagishi), mhjeong@hi.snu.ac.kr (M. Jeong), gzhu@adobe.com (G. Zhu), yzang4@u.rochester.edu (Y. Zang), yzh298@ur.rochester.edu (Y. Zhang), soumimaiti@meta.com (S. Maiti), florian.lux@ims.uni-stuttgart.de (F. Lux), nicolas.mueller@aisec.fraunhofer.de (N. Müller), zwyemrys@gmail.com (W. Zhang), csun22@buffalo.edu (C. Sun), shuweiho@buffalo.edu (S. Hou), siweilyu@buffalo.edu (S. Lyu), sebastien.lemaguer@helsinki.fi (S. Le Maguer), gongchengcheng@tju.edu.cn (C. Gong), ghj2001@mail.ustc.edu.cn (H. Guo), lipchen@ustc.edu.cn (L. Chen), vishwanath.singh@uef.fi (V. Singh).
[1] Currently at Apple.
[2] Currently at Adobe Research.
[3] Currently at Meta.

https://doi.org/10.1016/j.csl.2025.101825
Received 3 March 2025; Received in revised form 24 April 2025; Accepted 8 May 2025
Available online 28 May 2025

Keywords:
ASVspoof
Spoofing
Countermeasures
Deepfakes
Presentation attack detection
Corpus design

ASVspoof 5 is the fifth edition in a series of challenges which promote the study of speech spoofing and deepfake attacks as well as the design of detection solutions. We introduce the ASVspoof 5 database which is generated in a crowdsourced fashion from data collected in diverse acoustic conditions (cf. studio-quality data for earlier ASVspoof databases) and from ∼2000 speakers (cf. ∼100 earlier). The database contains attacks generated with 32 different algorithms, also crowdsourced, and optimised to varying degrees using new surrogate detection models. Among them are attacks generated with a mix of legacy and contemporary text-to-speech synthesis and voice conversion models, in addition to adversarial attacks which are incorporated for the first time. ASVspoof 5 protocols comprise seven speaker-disjoint partitions. They include two distinct partitions for the training of different sets of attack models, two more for the development and evaluation of surrogate detection models, and then three additional partitions which comprise the ASVspoof 5 training, development and evaluation sets. An auxiliary set of data collected from an additional 30k speakers can also be used to train speaker encoders for the implementation of attack algorithms. Also described herein is an experimental validation of the new ASVspoof 5 database using a set of automatic speaker verification and spoof/deepfake baseline detectors. With the exception of protocols and tools for the generation of spoofed/deepfake speech, the resources described in this paper, already used by participants of the ASVspoof 5 challenge in 2024, are now all freely available to the community.

1. Introduction

The potential threats and risks related to the misuse of deepfakes—synthetic or manipulated media generated with the aid of deep learning—are well acknowledged both by various professional communities (e.g. security, forensics, biometrics) and government bodies, as well as by the general public. Having emerged as a novel type of information security threat which covers multiple domains—news, social media, and communication to name a few—it is more timely than ever to develop proactive defences against deepfakes. The focus of this article is specifically *speech* deepfakes. Speech, as the primary means of human communication, carries not only the linguistic message but also *paralinguistic* (beyond language) information. This includes voice timbre in addition to other personally identifiable attributes, representations of which can be estimated from just a few seconds of speech and then used to infer the speaker identity using automatic speaker verification (ASV) technology. The voice timbre can then be transplanted into synthetic or manipulated speech to impersonate a specific, target individual.

The most widely studied solution to defend against the potentially harmful impact of speech deepfakes takes the form of *detection* solutions which provide a means to distinguish between bona fide (genuine) and synthetic or manipulated (spoofed) speech. Their reliability is strongly dependent on the availability of realistic and representative *data* with which a detection model is implemented and trained. Among other factors which complicate the design of reliable detection solutions is the diversity in text-to-speech (TTS) synthesis and voice conversion (VC) technology now available (even to the layman) to generate or manipulate speech signals. Furthermore, the multitude of different model architectures and training algorithms continues to evolve at an astonishing pace, implying that detectors trained even only a few months ago have the potential to fail in the face of deepfakes generated using technologies which have emerged since. Media distribution platforms alike continue to evolve, as do the encoding and lossy compression algorithms they use, all of which can introduce distortions which interfere with or perturb a detection system trained without identically treated or sufficiently similar data.

Born from a special session held at Interspeech in 2013, the efforts to develop detection solutions for speech data were initially spearheaded through the **ASVspoof initiative and challenge series**, founded over a decade ago (Evans et al., 2013; Wu et al., 2015a, 2017; Kinnunen et al., 2017; Todisco et al., 2019; Yamagishi et al., 2021; Wang et al., 2024a). ASVspoof has evolved from a challenge in the detection of spoofing attacks implemented with now-legacy TTS and VC algorithms to encompass also the detection of deepfake attacks implemented with the most recent deepfake technology as well as their impact upon the reliability of ASV systems. The tracking of developments in the speech synthesis and voice conversion communities is key to the pursuit of reliable and robust detection solutions. Each challenge edition was hence accompanied by the collection and public release of an updated ASVspoof database. The most recent challenge edition, ASVspoof 5, is no exception. The design, collection and validation of such data is a non-trivial undertaking and requires coordinated and collaborative international effort; in this article we summarise the collective effort of 2.5 years of work conducted by members of 18 teams spanning three continents.

Earlier ASVspoof databases were generated predominantly from the Voice Cloning Tool Kit (VCTK) database (Yamagishi et al., 2019). It contains studio-quality recordings of speech collected in a hemi-anechoic chamber and hence lacks distortions which might typify recordings collected in the wild, e.g. additive and convolutional noise, non-linear noise, coding and compression artefacts, narrow bandwidths, etc. Evaluation results derived using such databases might not provide the most reliable estimates of performance which could be expected in practical scenarios. Since spoof/deepfake attacks are generated using *clean* training and speaker-specific adaptation data, the resulting high quality of synthesised or converted speech might inflate increases in the false alarm/acceptance rate for both ASV and spoof/deepfake detection systems, referred to as countermeasures (CMs); an adversary is unlikely to have the luxury of using studio-quality adaptation data collected from the victim. For the same reason, the reliability of CMs, when trained with data of matched, high quality might also be exaggerated.

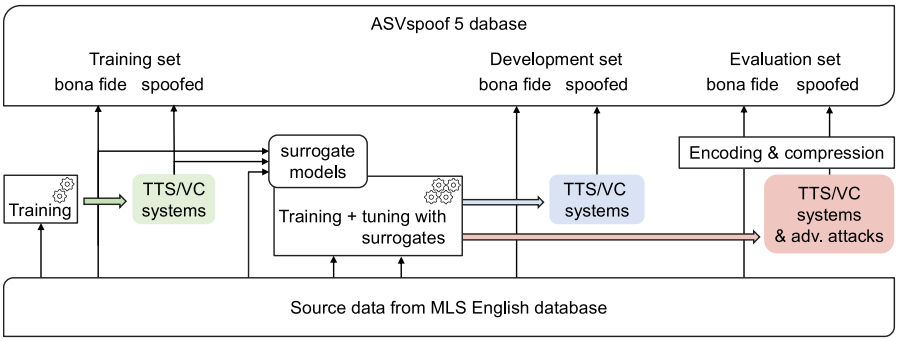

Fig. 1. Overview of the ASVspoof 5 database. Bona fide utterances contained in the speaker-disjoint training, development, and evaluation sets are sourced directly from the MLS English database. Spoofed utterances in each of the three sets are generated using distinct TTS and VC attack algorithms and optionally combined with supplementary adversarial attacks. All attacks are trained using held-out training data again sourced from the MLS English database. Attacks in the development and evaluation sets are optionally tuned using surrogate ASV and CM models, both trained using bona fide and spoofed data contained within the ASVspoof 5 training set in addition to additional, reserved data.

The adoption of a new source database hence offers an opportunity to provide the community with data which is more representative of that likely to be encountered in the wild. This, in turn, should help provide more reliable estimates of the impact of attacks as well as CM performance. It also offers an opportunity to provide data collected from a greater number of speakers as well as to assess detection performance when the quantity of TTS/VC training/adaptation data varies. All earlier ASVspoof databases contain data collected from ∼100 speakers.[4] The low number of speakers is a constraint on not just the protocol design, but also on the potential to train speaker-independent CMs. The adoption of a new source database also necessitates the design of new attack algorithms capable of generating spoofs/deepfakes using lower-quality training and adaptation data, as well as contemporary coding and compression techniques, specifically those developed since the last ASVspoof challenge held in 2021. Last, we have been alerted to the presence of *shortcut* artefacts (Geirhos et al., 2020) which plagues some earlier ASVspoof databases (Chettri et al., 2020; Müller et al., 2021; Shim et al., 2023; Zhang et al., 2023; Liu et al., 2023), namely artefacts which are semantically unrelated to the spoof/deepfake problem but which can nonetheless be utilised for detection. The collection of a new database hence offers an opportunity to reduce such biases.

As illustrated in Fig. 1, the new **ASVspoof 5 database**[5] is generated from the English partition of the Multilingual Librispeech (MLS) database (Pratap et al., 2020). It is itself sourced from LibriVox (Kearns, 2014), a collection of free public domain audiobooks contributed by volunteers from around the world.[6] Having been collected in each contributor's own recording setting, the MLS source database contains far greater acoustic variation than the VCTK database. The MLS database furthermore contains speech recordings collected from thousands of speakers. This order-of-magnitude increase in the number of speakers (cf. previous ASVspoof databases) permits greater flexibility in database and protocol design, in particular so that attack algorithms can be tuned using a set of *surrogate* CM and ASV systems (middle of Fig. 1) trained, developed and evaluated using held-out data. Spoofing and deepfake attacks contained within the ASVspoof 5 training, development and evaluation sets (green, blue and red boxes in Fig. 1) are generated by a group of data contributors, all experts in TTS and VC, again using disjoint data. Adversarial attacks (Szegedy et al., 2013; Goodfellow et al., 2015) are introduced (evaluation set only, red box in Fig. 1) for the first time, as is the use of neural codecs and a new pipeline to tackle the biases of shortcut artefacts. The latter is used to reduce the mismatch between distributions of peak waveform amplitude, the duration of leading and trailing non-speech segments and whole utterance duration, for bona fide and spoofed/deepfake utterances.

We describe the design of the ASVspoof 5 database (Section 2), the crowd-sourcing approach to the collection of spoofing/deepfake attacks (Section 2–3), the pipeline to reduce shortcut artefacts (Section 4), data visualisations to illustrate the similarities and differences between different attacks (Section 5), and an experimental validation using CM and ASV baselines (Section 7). A presentation of challenge results, beyond that already available in Wang et al. (2024a), is in preparation for later publication.

2. Database generation

We describe the design of the ASVspoof 5 protocols and database and its generation from new source data. The treatment is dense and, for this reason, readers who are interested only in using the ASVspoof 5 database for their own research in detection

[4] The ASVspoof 2015 (Wu et al., 2015a), 2019 Logical Access (Wang et al., 2020), 2021 Logical Access and Deepfake databases (Yamagishi et al., 2021) contain data collected from ∼100 speakers sourced from the VCTK database. The ASVspoof 2021 Deepfake database contains data collected from an additional 26 speakers sourced from the Voice Conversion Challenge 2018 (Lorenzo-Trueba et al., 2018) and 2020 (Zhao et al., 2020) databases. The ASVspoof 2017 (Kinnunen et al., 2017) and 2019 Physical Access databases, which contain data collected from 42 and 106 speakers respectively, focus on replay spoofing attacks.

[5] ASVspoof 5 database link: https://doi.org/10.5281/zenodo.14498691.

[6] https://librivox.org/

tasks (speakers or attacks) should consider reading at least Sections 2.1 and 2.2. These readers might then prefer to move directly either to Section 3 or to Section 5. Readers with a keen eye for protocol design, or those with an interest in using ASVspoof 5 protocols to generate their own attacks[7] should consider reading this section in its entirety. The focus throughout is upon source data and protocols. A description of specific TTS/VC systems and adversarial attacks is provided in Section 3.

2.1. Source database

Earlier ASVspoof databases were generated using primarily the VCTK (Yamagishi et al., 2019) source database which contains utterances collected from approximately 100 speakers in a single, hemi-anechoic chamber. This choice was made to ease the generation of spoofed speech yet, as a consequence, both bona fide and spoofed utterances are of higher quality than might be expected in practice. Adversaries may also not be able to acquire training data of similar studio quality. The evaluation of detection performance in the face of non-studio-quality data calls for the adoption of an alternative source database.

The ASVspoof 5 database is generated using the MLS English database (Pratap et al., 2020). It contains data collected from approximately 2400 female and 2300 male speakers in diverse recording conditions (acoustic environments and devices). The MLS database is itself sourced from LibriVox (Kearns, 2014), a collection of free, public domain audiobooks. Recordings are made by individual contributors in their own home or office and with their own recording devices. The MLS database hence contains speech data captured in variable, non-studio-quality conditions. Utterances contained within the MLS English database are of 10-to −20 s and originate from the segmentation of LibriVox audiobook recordings.

As illustrated in Fig. 2, the MLS database is partitioned into 7 speaker-disjoint subsets. They are constructed carefully according to a number of speaker/utterance criteria explained later in Section 2.5. First, we describe the purpose of each partition, starting with those which comprise the ASVspoof 5 database, then a pair used for the training of TTS/VC systems and adversarial attacks, followed by a pair of auxiliary partitions related to the use of surrogate CM and ASV systems.

2.2. ASVspoof 5 database partitions

At a high level, the ASVspoof 5 database is similar in structure to that of earlier editions: it contains the usual three training, development and evaluation partitions illustrated by shaded, grey boxes in Fig. 2. The training partition (partition #2 in Fig. 2) is intended for the training of CM and ASV systems as well as spoofing-robust ASV (SASV) systems.[8] It comprises a set of utterances collected from 400 speakers (second row), 19k bona fide utterances (#2.1, sourced directly from the MLS database) and 164k spoofed utterances (#2.2), generated using a set of 8 distinct TTS/VC algorithms (described in Section 3.2).

The ASVspoof 5 development set (#6), is intended for the usual purpose and contains two subsets. The first comprises utterances corresponding to 398 speakers designated as target speakers. There are between 1 and 3 utterances per target speaker for ASV enrolment (#6.0), 17k bona fide utterances (#6.1) and 110k spoofed utterances (#6.2). The latter are again generated using a set of distinct TTS/VC algorithms (described in Section 3.3) which are different to those used for generation of the training set. The second subset contains 14k bona fide utterances collected from 387 speakers designated as non-target speakers (#6.3).

Last, the ASVspoof 5 evaluation set, again intended for the usual purpose, has an identical structure to the development set. The number of speakers in each subset is 367 target speakers and 370 non-target speakers. There are again between 1 and 3 utterances for ASV enrolment (#7.0) but a far greater number of bona fide and spoofed utterances, counting 71k and 68k bona fide utterances for target speakers (#7.1) and non-target speakers (#7.3) respectively. The 542k spoofed utterances (#7.2) are generated using 16 distinct TTS/VC and adversarial attacks (described in Section 3.4). Once again, they are different to those used for generation of the training and development sets. To simulate a broad variety of applications where speech data is transmitted or compressed, various lossy encoding schemes are also applied to all evaluation data (both bona fide and spoofed). The encoding schemes are described in Section 4.1.

2.3. Attack generation

In the following we delve deeper into use of the database and protocols for attack generation. We describe the disjoint MLS database partitions used for generation of spoofed data in the ASVspoof 5 training, development and evaluation partitions. Attack algorithms were designed through two phases and in collaboration with external contributors, all experts in TTS/VC technology.

As has been the case for earlier databases, the ASVspoof 5 database and protocols were designed to support use of the very latest TTS/VC technologies. Whereas earlier databases were generated using TTS/VC systems which required extensive training data for each target speaker, today's state-of-the-art systems require much less. Typically, they are trained using huge collections of speech collected from a large number of speakers. Pre-trained systems are then fine-tuned using relatively few utterances collected from the specific target speaker (Wu et al., 2015b; Arik et al., 2018; Yan et al., 2021; Chen et al., 2021a). Some systems even operate without fine-tuning and, instead, generate speech in the voice of a specific target speaker by using a single speaker embedding (Jia

[7] For ethical reasons, attack generation protocols are not shared publicly. Those interested in the use of attack generation protocols to support further research in detection can request access to the generation protocols by sending an email to the ASVspooforganisers.

[8] An SASV system functions in the same way as combined CM/ASV subsystems—they should operate to accept only input utterances which are bona fide and which contain the target speaker voice, and to reject anything else (spoofs/deepfakes) and bona fide utterances which do not match the target speaker voice.

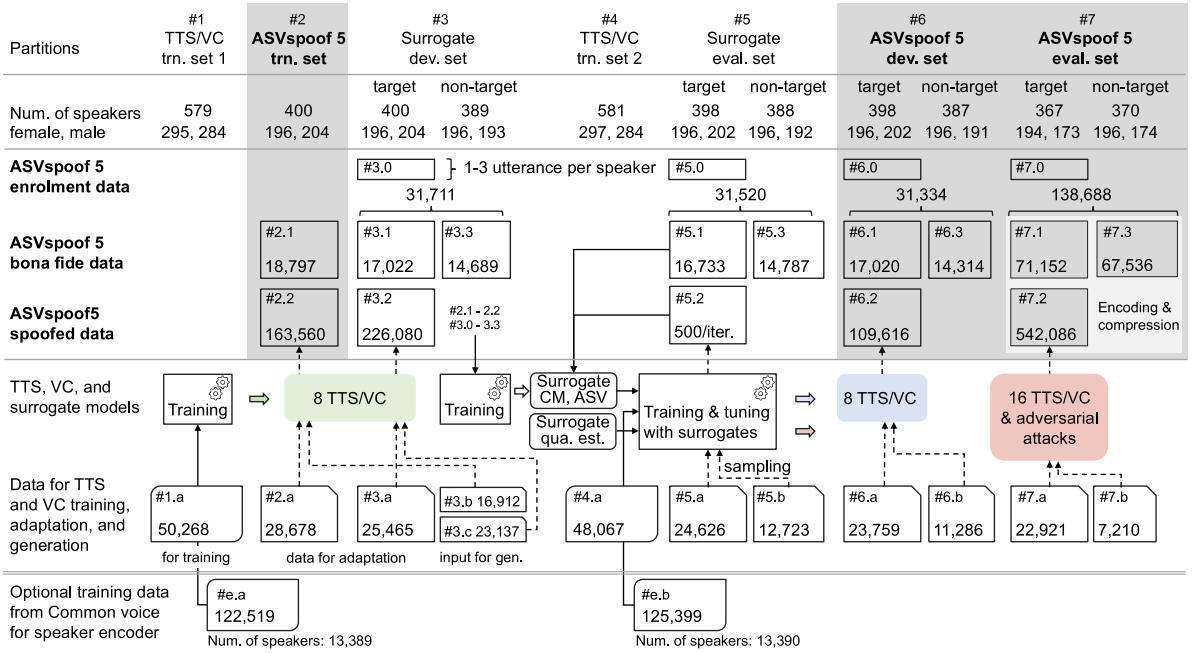

Fig. 2. An illustration of the MLS English source database partitioning scheme and its use in generation of the ASVspoof 5 database, including training, development and evaluation sets illustrated in grey boxes. Numbers in the second row indicate the number of speakers in each partition. Those in rectangular blocks denote the number of utterances. The green box denotes the 8 TTS/VC systems used for generation of the ASVspoof 5 training set. The blue box denotes 8 TTS/VC systems used for generation of the development set whereas the red box denotes the 16 TTS/VC systems and adversarial attacks used for generation of the evaluation set. Solid black and coloured arrows denote use of data for the training of TTS/VC attacks or surrogate models. Dashed arrows denote the adaptation towards target speaker and attack generation using input text or source speech utterance.

et al., 2018; Arik et al., 2018; Cooper et al., 2020; Wu et al., 2022), i.e. zero-shot voice cloning. The ASVspoof 5 database and protocols are hence devised with such TTS/VC systems in mind.

The TTS/VC training set 1 (`#1.a`) is reserved for the training of 8 TTS/VC systems used for the generation of spoofed utterances in the ASVspoof 5 training set (`#2.2`). It contains 50k utterances collected from 579 speakers. Spoofed utterances are generated using held-out subsets of adaptation utterances collected from target speakers and input utterances collected from non-target speakers. Adaptation is typically used to clone the voices of the 400 target speakers in the ASVspoof 5 training set, for example by fine-tuning towards the voice of a specific target, or by extracting and then using target speaker embeddings for zero-shot voice cloning. Input utterances[9] are used for VC or for TTS using corresponding transcriptions. The choice depends on the specific attack algorithm (described in Section 3).

To observe the influence of different durations of adaptation data, spoofed utterances are generated using different numbers of adaptation utterances. The configurations are listed in Table 1.[10] The first possible, but technically-demanding scenario involves the use of recordings collected surreptitiously from the target speaker during their interaction with the ASV system (`AC1`).[11] These recordings are assumed to be in the order of 10 s duration. Their collection in acoustic conditions identical to those for collection of enrolment data is expected to result in a stronger attack and hence to represent a worst case scenario. The two additional scenarios reflect the case where the adversary uses short recordings found online (≈30 s, `AC2`) or a large set of utterances from multiple sources collected from a more intensive search (≈20 minutes, `AC3`). With these two cases we aim to study the influence of session mismatch,[12] which might result in a weaker attack with respect to `AC1` as well as the quantity of adaptation data.

The TTS/VC training set 2 (`#4.a`) contains 48k utterances and is used for training of the 16 TTS/VC systems and adversarial attacks used for the generation of spoofed utterances contained within the ASVspoof 5 evaluation set. The same set is used for training of the eight TTS/VC systems for the ASVspoof 5 development set. Held-out adaptation and input utterances are again used in similar fashion as for the TTS/VC training set 1.[13]

[9] For reasons described shortly, input utterances are selected from the disjoint surrogate development partition described in Section 2.4.

[10] Following (Tan et al., 2021), we experimented with many different numbers of utterances ranging from 10 s to 20 min of adaptation data. To maintain a reasonable database size, we retained the reported three configurations which result in substantially different surrogate ASV performance derived using the model described in Section 2.4.

[11] Utterances segmented from the same audiobook are assumed to be collected in the same recording session. Audiobook contributors are requested to make recordings without interruptions (see Librivoxrecordingguidelines). The session ID is retrieved from the MLS utterance file name.

[12] Utterances segmented from different audiobooks.

Table 1
TTS/VC adaptation configurations.

Configuration ID	AC1	AC2	AC3
Number of adaptation utterances	≤3	3	≈100
Total duration of adaptation utterances	≤30 s	≈30 s	≈20 m
Adaptation and enrolment data from the same session?	✓	×	×

2.4. Surrogate ASV, CM, and perceptual quality estimation systems

Most TTS/VC systems are developed for applications which demand high perceptual quality, whereas adversaries in the context of spoofs/deepfakes might tune TTS/VC systems to increase the likelihood of fooling CM and ASV systems. The threat of TTS/VC systems used for generation of spoofed utterances contained within the ASVspoof 5 development and evaluation sets is gauged using surrogate CM and ASV systems as well as a perceptual speech quality estimation system. We assume that attackers cannot access the CM and ASV systems used by the defender but can instead choose other open-source implementations as substitutes (Papernot et al., 2016), hence the term 'surrogate'.

The surrogate systems are illustrated towards the lower middle of Fig. 2 and are trained, developed and evaluated in the usual way. While the provision of surrogate systems[14] offers an opportunity for attack algorithms to be fine-tuned or optimised, given that spoofed utterances are generated by independent contributors, it is not possible to estimate reliably the relative degree to which each attack is tuned, though some analysis is provided in Section 6. Invariably, tuning involves merely experimentation to verify the degree to which attacks are effective and to adapt the algorithms accordingly. Surrogate systems were used for fine-tuning attack algorithms that are later used in generation of ASVspoof 5 development and evaluation sets only (not for the train set).

Surrogate CM/ASV systems are trained using ASVspoof 5 bona fide and spoofed utterances (`#2`). The enrolment, bona fide, and spoofed data from partition `#3` are used as surrogate development data. The structure of the surrogate development (`#3`) and evaluation sets (`#5`), additional disjoint partitions illustrated in Fig. 2, is similar to that of the ASVspoof 5 development and evaluation sets. They are also partitioned into target and non-target subsets. Each subset contains data collected from approximately 400 speakers. The surrogate development set contains 17k bona fide utterances (`#3.1`) and 226k spoofed utterances (`#3.2`) for target speakers and 15k bona fide utterances for non-target speakers (`#3.3`). Spoofed utterances are generated using the same 8 TTS/VC algorithms used for generation of the ASVspoof 5 training set, again using a set of adaptation utterances (`#3.a`) and input utterances for voice conversion, or transcriptions for text-to-speech synthesis (`#3.c`).

The surrogate evaluation set contains 17k bona fide utterances (`#5.1`) for target speakers and 15k bona fide utterances for non-target speakers (`#5.3`). TTS and VC systems trained using the TTS/VC training set 2 (`#4.a`) are optionally fine-tuned to increase surrogate CM/ASV system error rates via multiple iterations. In each iteration, TTS/VC systems are used to generate up to 500 spoofed utterances (`#5.2`) after the sampling of adaptation utterances (from `#5.a`) or input utterances for generation (from `#5.b`). The generated spoofed utterances, together with the target and non-target utterances in the surrogate evaluation set (`#5.1` and `#5.3`), are fed to the surrogate models to compute detection error rates which can then be used for TTS/VC system fine-tuning.

2.5. Speaker and utterance selection

The number of utterances per speaker in the source MLS English database is unbalanced (between 1 and 300k utterances per speaker). Such imbalance can lead to bias in the training and evaluation of any derived system. The partitions illustrated in Fig. 2 are hence created in deterministic fashion by ensuring that all speakers designated as targets (ASVspoof 5 development and evaluation sets as well as both surrogate sets) have a minimum of two recording sessions. Furthermore, ASVspoof 5 enrolment and bona fide utterances for each target speaker are selected from different recording sessions. Last, so as to maintain a suitable balance between the influence of speakers for which data is abundant and those for which data is sparse, the number of bona fide utterances for each speaker (target and non-target) is capped at 50 and floored at 30.

Data for all speakers in the evaluation set, including both targets and non-targets, are removed if data corresponding to the same speakers also appears in the Librispeech database (Panayotov et al., 2015). Both Librispeech and MLS databases are sourced from LibriVox. Many popular self-supervised learning (SSL) models (Mohamed et al., 2022, Table IV) are trained using the Librispeech database and use of the same data within the ASVspoof 5 evaluation set would lead to biases in the evaluation of derived systems.

[13] Since the ASVspoof 5 development and evaluation sets necessarily include both target and non-target data (to support ASV evaluation), input data for TTS/VC generation can be selected from non-target utterances within the same partition. This is why source utterances are selected from a different partition in case of the ASVspoof 5 train set for which there are no non-target speakers; it is not designed to support ASV evaluation.

[14] To reduce the burden on data contributors, access to surrogate systems was provided through an online platform. Contributors could submit generated TTS/VC utterances and use estimated ASV/CM/perceptual quality scores for fine-tuning.

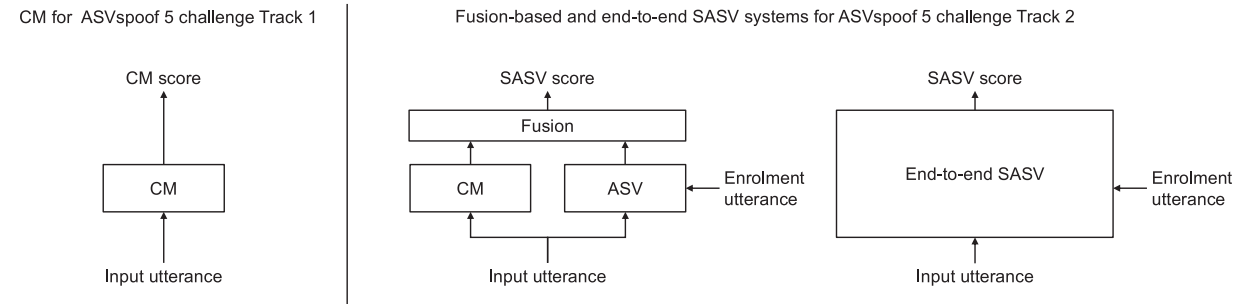

Fig. 3. An illustration of the Track 1 spoof/deepfake detection task which involves the design of a CM and the Track 2 spoofing robust automatic speaker verification (SASV) task for which there are two general approaches.

2.6. Speaker encoder training

To support the use of TTS/VC systems that use speaker representations in the form of ASV embeddings to generate outputs in the voice of specific target speakers (Jia et al., 2018; Cooper et al., 2020), contributors could select to use a subset of the Common Voice (ver. 11.0) (Ardila et al., 2020) database for ASV model training. Similar to the MLS English database, the distribution in terms of data quantity per speaker is unbalanced. To maintain a reasonable balance, but still a suitable number of utterances and speakers, the subset corresponds to the set of speakers for which there is between 10 and 240 s of speech. This results in a total of 200 h of data. Two different subsets were created, each containing data from approximately 13k speakers. They contain approximately 123k utterances (`#e.a`) and 125k utterances (`#e.b`), as illustrated in the last row of Fig. 2. Their use is optional and is reserved exclusively for the training of TTS/VC system speaker encoders.

2.7. Challenge tracks

The ASVspoof 5 database was designed to support two different tasks which form a pair of challenge Tracks. As illustrated to the left of Fig. 3, Track 1 is a spoof/deepfake detection task and concerns the design of CM systems which produce a score for each input utterance, wherein higher scores indicate a higher likelihood that the input utterance is bona fide (or, equivalently, a lower likelihood that it is spoofed). Track 1 does not involve the design of an ASV system. CMs are implemented using the ASVspoof 5 training (`#2.1-#2.2`) and development (`#6.1-#6.3`) sets with evaluation then being performed using the evaluation set (`#7.1-#7.3`).

As illustrated to the right of Fig. 3, Track 2 involves the design of SASV systems which produce a score for each input utterance, but now where a higher score indicates a higher chance that the input utterance is bona fide and that it matches the voice in the enrolment utterance, i.e. the target speaker (or, equivalently, a lower chance of being anything else). SASV systems typically take one of two different forms involving either the fusion of independent CM and ASV systems or a single, end-to-end system. SASV systems are implemented and evaluated using the same training, development, and evaluation sets as for Track 1, except that utterances in `#6.0` and `#7.0` are used in addition. They provide data for the enrolment of target speakers for corresponding trials in the development and evaluation sets. Readers are encouraged to consult the ASVspoof 5 challenge evaluation plan (Delgado et al., 2024) for full details.

3. Spoofing attacks

The TTS/VC research landscape has evolved dramatically since the release of the most recent, previous ASVspoof database. New techniques borrowed from the deep learning community, e.g., diffusion, have been used to create even higher quality synthetic speech. However, successful spoofing attacks are not necessarily implemented using the latest techniques, and might not even require the highest quality synthetic speech to fool a CM (Warren et al., 2024) or an ASV system (Jung et al., 2024). It is hence imperative that attacks generated with both recent and legacy TTS/VC algorithms are included in the ASVspoof 5 database. Also included are the latest adversarial attacks. While it is impractical to cover all related TTS/VC architectures and adversarial attacks, we considered those to the extent possible and collected a database of attacks generated with a set of 32 attack algorithms listed in Table 2.

Before describing each attack, we first provide a general definition of TTS, VC and adversarial attacks. Readers who are already familiar with ASVspoof databases and associated generative technologies can choose to skip this material and move directly to Section 3.2 which provides a description of the attacks used in creating the training, development, and evaluation sets.

3.1. Attack types

TTS-based attacks are performed using a synthetic utterance which contains the voice of a target speaker learned from one or more adaptation utterances and the linguistic content of a provided text input. As illustrated in Fig. 4, TTS systems follow one of

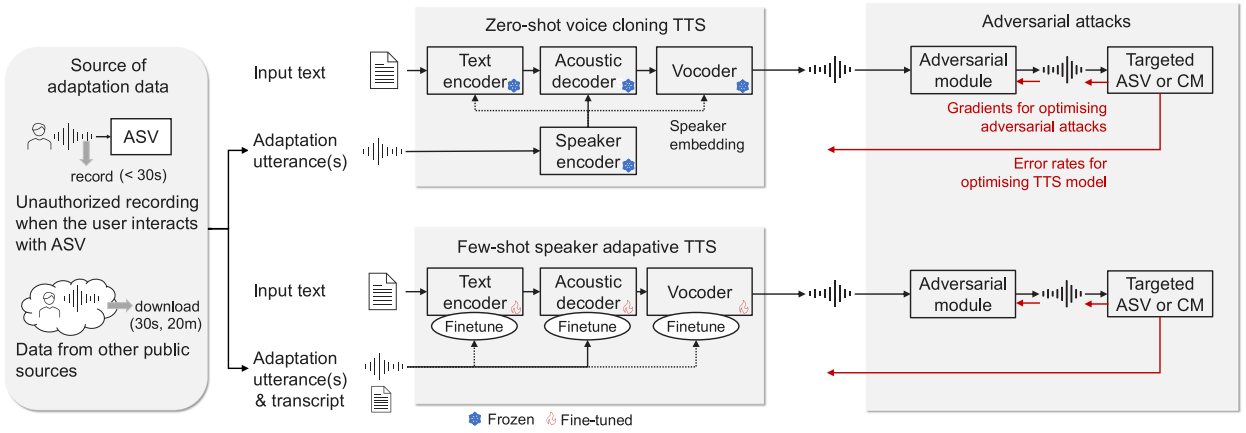

Fig. 4. Illustration of speech generation using zero-shot (top-middle) and few-shot (bottom-middle) TTS systems. Few-shot TTS modules are fine-tuned using the adaptation data before generation, while zero-shot TTS modules are frozen (not fine-tuned). Dotted lines indicate the optional use of fine-tuning or speaker embeddings by the text encoder and vocoder. Illustrated to the left is the source of adaptation utterances for a target speaker. Illustrated to the right is the creation of adversarial attacks, whereby a TTS system is optimised using either the gradients or the error rates returned by the targeted ASV or CM system. The red lines indicate operations applied only for training. A similar figure can be plotted for VC attacks by replacing the input text, the text encoder, and the acoustic decoder with an input utterance from a non-target speaker, a waveform encoder, and a VC-oriented acoustic decoder, respectively.

two general approaches, namely zero-shot voice cloning or few-shot speaker adaptive TTS.[15] A text encoder is used to transform the input text into features which (explicitly or implicitly) encode word pronunciation information and other suprasegmental attributes (e.g., pitch and rhythm). The acoustic decoder then produces acoustic features (e.g., mel-spectrograms), which are further transformed by the vocoder into a speech waveform.[16] The two approaches illustrated in Fig. 4 differ in how the target speaker voice is learned. In the case of zero-shot voice cloning (top-middle), a target speaker embedding is extracted from the adaptation utterances by a speaker encoder. The acoustic decoder, and optionally also the text encoder and vocoder, are then conditioned upon the target speaker embedding. Since they use target speaker embeddings, zero-shot voice cloning TTS systems are pre-trained using data collected from a large number of speakers, but are not fine-tuned at the level of the target speaker. This sets zero-shot voice cloning systems apart from few-shot speaker adaptive TTS systems (bottom-middle) which are fine-tuned to the voice of the target speaker using the set of adaptation utterances and their transcriptions.

VC-based attacks are implemented using an input utterance containing the voice of a non-target speaker in place of input text, but again using one or more adaptation utterances which contain the voice of a target speaker. The voice in the input utterance is converted to that of the target speaker, whereas the linguistic content of the former is preserved. Approaches to VC can be categorised similarly to the TTS systems illustrated in Fig. 4, though with input utterances in place of input text and with a waveform encoder in place of the text encoder. VC acoustic decoders typically use similar statistical models or deep neural networks (DNNs) as TTS, but with necessary modifications to accommodate input features of different dimension.

Adversarial attacks involve the introduction to an utterance of discreet perturbations designed specifically to increase CM/ASV error rates. Adversarial attacks usually take the form of subtle, additive noise, e.g., estimated via a fast gradient sign method (Goodfellow et al., 2015), designed for a specific input utterance and targeted CM/ASV system (Liu et al., 2019; Wang et al., 2024c; Li et al., 2020). These techniques can be costly in terms of computation, especially when adversarial noise is estimated for a large number of utterances. Though still designed for a specific CM/ASV system, less costly, more universal adversarial attacks, have also been reported (Panariello et al., 2023; Todisco et al., 2024). While these can be applied to any input, they are applied to already-spoofed utterances with the aim of making them more difficult to detect and are hence designed for the specific, underlying spoofing attack and/or speaker. Our use of such adversarial attacks is illustrated to the right of Fig. 4, wherein post-processing adversarial modules are applied to spoofed utterances produced by a given TTS/VC system. Adversarial modules are trainable, convolutive filters which target a specific CM system (Panariello et al., 2023) or a specific ASV system (Todisco et al., 2024). The trainable parameters (i.e., filter coefficients) are learned from the gradients back-propagated from the targeted CM/ASV system. The fine-tuning of a TTS/VC system according to the surrogate CM/ASV or perceptual quality estimator outputs (Section 2.4) is also a form of black-box adversarial training. In this case, the TTS/VC system observes scores or decisions rather than gradients.

[15] Some legacy approaches such as unit-selection TTS are neither zero-shot nor few-shot in strict terms. To help clarify the concept, we consider the operation of the unit-selection TTS, which is used in the ASVspoof 5 database, as a special form of few-shot speaker-adaptive TTS. These systems synthesise speech by selecting and concatenating small speech segments called units. The acoustic decoders (i.e., the selection algorithms) extract units from the adaptation utterance(s). The resulting sequence of units is then simply concatenated by the 'vocoder'. Readers are encouraged to refer to TTS literature for an in-depth explanation of legacy approaches (Taylor, 2009).

[16] The modules are not necessarily implemented in the form of DNNs. They can instead be a set of linguistic rules or decision trees. So-called end-to-end TTS systems in the form of a single DNN can be divided conceptually into distinct blocks, each of which corresponds to one of the modules illustrated in Fig. 4.

Table 2
Summary of spoofing attacks in the ASVspoof 5 database. The column Reference lists the name of the TTS system used in other literature, if available. Note that A30 applies both Malafide and Malacopula to the base attack A17. TF: Transformer; CF: Conformer; GST: Globle style token.

			Category	Text/wav. encoder	Acoustic decoder	Speaker encoder	Vocoder	Reference
Training set	TTS	A01	Zero-shot TTS	TF encoder	Glow	ECAPA	HiFi-GAN	Glow-TTS (Kim et al., 2020)
		A02	Zero-shot TTS	TF encoder	Glow	x-vec.	HiFi-GAN	Glow-TTS
		A03	Zero-shot TTS	TF encoder	Glow	y-vec.	HiFi-GAN	Glow-TTS
		A04	Zero-shot TTS	TF encoder	Diffusion	ECAPA	HiFi-GAN	Grad-TTS (Popov et al., 2021)
		A05	Zero-shot TTS	TF encoder	Diffusion	x-vec.	HiFi-GAN	Grad-TTS
		A06	Zero-shot TTS	TF encoder	Diffusion	y-vec.	HiFi-GAN	Grad-TTS
		A07	Zero-shot TTS	TF encoder	TF encoder	ECAPA	HiFi-GAN	FastPitch (Łańcucki, 2021)
		A08	Zero-shot TTS	TF encoder	Normalising flow	x-vector	HiFi-GAN	VITS (Kim et al., 2021)
Development set	TTS and VC	A09	Zero-shot TTS	CF	CF + Glow, F0	GST	HiFi-GAN	ToucanTTS (Lux et al., 2023)
		A10	Zero-shot TTS	CF	CF + Glow	GST	HiFi-GAN v2	ToucanTTS
		A11	Zero-shot TTS	CNN + RNN	Attention + AR	RNN-G2E	WaveGrad	Tacotron2 (Shen et al., 2018)
		A12	Few-shot TTS	Linguistic-based	Unit (phone) select	–	Wav. concat.	–
		A13	Zero-shot VC	CNN	CNN	GRU-RNN	WaveGlow	StarGAN-ZSVC (Baas and Kamper, 2020)
		A14	Zero-shot TTS	TF encoder	Normalising flow	RawNet3	HiFi-GAN	YourTTS (Casanova et al., 2022)
		A15	Few-shot VC	CNN-VAE	CycleGAN	–	WaveNet	AlBadawy and Lyu (2020)
		A16	Zero-shot VC	wav2vec + F0	–	CAM++	HiFi-GAN	–
Evaluation set	TTS and VC	A17	Zero-shot TTS	BERT + TF encoder	VQ-VAE	ECAPA	HiFi-GAN	ZMM-TTS (Gong et al., 2024)
		A19	Few-shot TTS	Linguistic-based	Non-uniform unit-select	–	Wav. concat.	MaryTTS (Schröder et al., 2011)
		A21	Zero-shot TTS	CF	CF + Glow, F0	GST	BigVGAN	ToucanTTS
		A22	Zero-shot TTS	CF	CF + Glow	GST	BigVGAN	ToucanTTS
		A24	Zero-shot VC	RNN-CNN PPG	CNN	ECAPA	HiFi-GAN	–
		A25	Zero-shot VC	TF encoder	Duffsion	RNN-G2E	HiFi-GAN	DiffVC (Popov et al., 2022)
		A26	Zero-shot VC	wav2vec + F0	–	CAM++	HiFi-GAN	–
		A28	Zero-shot TTS	TF encoder	Normalising flow	H/ASP	HiFi-GAN	YourTTS, pre-trained
		A29	Zero-shot TTS	GPT-2	–	H/ASP	HiFi-GAN	XTTS (Casanova et al., 2024), pre-trained
			Category	Base attack	Targeted model	Filter length	#. branches	
	Adversarial	A18	Malafide	A17 (TTS)	CM (AASIST)	$L = 1025$	–	
		A20	Malafide	A12 (TTS)	CM (AASIST)	$L = 1025$	–	
		A23	Malafide	A09 (TTS)	CM (AASIST)	$L = 1025$	–	
		A27	Malacopula	A26 (VC)	ASV (CAM++)	$L = 1025$	$K = 5$	
		A30	Malafide +Malacopula	A17 (TTS)	ASV (CAM++)	$L = 1025$ $L = 257$	– $K = 3$	
		A31	Malacopula	A22 (TTS)	ASV (CAM++)	$L = 513$	$K = 3$	
		A32	Malacopula	A25 (VC)	ASV (CAM++)	$L = 1025$	$K = 5$	

3.2. Training set

The set of attack algorithms used to generate spoofed data contained within the ASVspoof 5 training set are all forms of zero-shot voice cloning TTS (top-middle of Fig. 4). A summary of each is presented in the top-most block of Table 2. More detailed descriptions are presented in the following.

A01: a neural TTS system based on Glow-TTS (Kim et al., 2020). At the core is an acoustic decoder which generates mel-spectrograms via a normalising-flow-based deep generative model named Glow (Kingma and Dhariwal, 2018). The acoustic decoder is conditioned on latent linguistic features which are extracted from the input text by a text encoder using the Transformer encoder architecture (Vaswani et al., 2017). The vocoder is a generative adversarial network (GAN)-based DNN named HiFi-GAN (Kong et al., 2020). The speaker encoder is an ECAPA-TDNN (Desplanques et al., 2020). Glow-TTS is open-sourced and is widely used within the TTS community.

A02: identical to A01, except for use of a ResNet-34-based speaker encoder (He et al., 2016).

A03: identical to A01, except for use of a TDNN-Y-vector-based speaker encoder (Zhu et al., 2021).

A04: a neural TTS system based on Grad-TTS (Popov et al., 2021), featuring a diffusion-based acoustic decoder. The acoustic decoder generates a mel-spectrogram via an iterative reverse diffusion process (Song et al., 2021). From latent linguistic features generated by the text encoder, the acoustic decoder with a U-Net architecture (Ronneberger et al., 2015) is used to predict a 'residual' (or, more specifically, the gradient field which maximises the probability of data to be generated). The residual is then summed with the latent features, the output of which is again fed to the acoustic decoder to produce another residual. The process is repeated ten times, and the output of the last iteration is used as a generated mel-spectrogram. The text encoder is a Transformer encoder with text and speaker embedding inputs. The vocoder is a HiFi-GAN. The implementation is based on the code released by the original paper, but speaker embeddings are extracted using an ECAPA-TDNN speaker encoder.

A05: identical to A04, except for use of a ResNet-34-based speaker encoder (He et al., 2016).

A06: identical to A04, except for use of a TDNN-Y-vector-based speaker encoder (Zhu et al., 2021).

A07: a neural TTS system based on FastPitch (Łańcucki, 2021) which uses a feedforward DNN without a recurrent layer to reduce generation time. The text encoder uses a stack of Transformer feedforward blocks to convert the input text into a sequence of latent linguistic feature vectors. The acoustic decoder then uses a convolutional neural network (CNN) to predict pitch for each latent vector. The paired pitch predictions are transformed to the same dimension as the latent vector before they are summed together. The summed vectors are then fed to another stack of Transformer feedforward blocks to generate a mel-spectrogram. A HiFi-GAN-based vocoder is then used to convert the mel-spectrogram into a speech waveform. The speaker encoder is an ECAPA-TDNN. FastPitch is known for its fast generation speed.

A08: a variational inference with adversarial learning for end-to-end text-to-Speech (VITS) (Kim et al., 2021) system which uses a Transformer-based text encoder and a HiFi-GAN vocoder. The acoustic decoder is based on normalising flow, similar to A01–A03. However, the text encoder, the acoustic decoder, and the vocoder are jointly optimised by maximising a variational evidence lower bound (Kim et al., 2021). Speaker embeddings are x-vectors (Snyder et al., 2018). VITS produces especially high quality synthetic speech and is readily accessible with the open-sourced toolkit ESPNet (Hayashi et al., 2020).

3.3. Development set

The ASVspoof 5 development set contains attacks generated with a mix of zero-shot and few-shot TTS and VC systems. A summary of each is presented in the second block of Table 2.

A09: a zero-shot TTS system (Lux et al., 2023) implemented using the IMS Toucan speech synthesis toolkit (Lux et al., 2021). The system is similar to A07 in terms of using feedforward DNNs but features a number of more advanced techniques. Following (Wu et al., 2022), the speaker encoder is equipped with Global Style Tokens (GSTs) (Wang et al., 2018), a set of 2000 latent vectors (or tokens) trained to encode speaker-related information. For voice cloning, a CNN is first applied to extract an initial speaker embedding from the adaptation utterance(s). A refined speaker embedding is then produced using a query-key–value attention block (Vaswani et al., 2017). The raw embedding serves the query and the GSTs are the keys and values. GSTs are optimised jointly with other system components. Both the text encoder and acoustic decoder use stacks of Conformer blocks (Gulati et al., 2020), and the acoustic decoder is further supplemented with a Glow-based (Kim et al., 2020) post-processing module (Ren et al., 2021). The acoustic decoder is also conditioned on pitch and energy estimates extracted from an input utterance to clone the prosodic patterns of a non-target speaker (Lux et al., 2022). The vocoder is based on HiFi-GAN. The Toucan toolkit is available as open-source and generation speed is faster than real time.

A10: identical to **A09**, except for the prediction of pitch and energy from the input text using CNNs (Łańcucki, 2021) and the use of a smaller HiFi-GAN vocoder model (Kong et al., 2020) than the A07 attack upon which A09 is based.

A11: a zero-shot TTS system based on Tacotron 2 (Shen et al., 2018). The speaker encoder is a three-layer recurrent neural network (RNN) with long-short-term-memory (LSTM) units (Graves, 2008), trained for a speaker diarization task using generalised end-to-end (GE2E) loss (Wan et al., 2018). The text encoder comprises convolution and recurrent layers. The acoustic decoder uses a location-sensitive attention block (Chorowski et al., 2015) and an autoregressive (AR) DNN (Shen et al., 2018) to generate a mel-spectrogram which is then converted into a speech waveform using the diffusion-based WaveGrad (Chen et al., 2021b) vocoder. Tacotron 2 has been reported to produce speech of near-to-natural quality for single-speaker, neutral style TTS (Shen et al., 2018).

A12: a simplified unit-selection TTS system. A set of speech units (phones) is constructed for each target speaker. The set is derived from the segmentation of the adaptation utterance(s) given the phone alignments produced by a pre-trained automatic speech recognition (ASR) system provided with the Toucan toolkit (Lux et al., 2021). This is the only data used; there is no use of any ASVspoof 5 training data, nor of any other externally-sourced training data. During generation, the system selects and concatenates together sequences of units which match the phonemic symbols of the input text. Phonemic symbols not covered by the pool of units are ignored. If multiple candidate units are available, one is selected at random. Beyond use of the pre-trained ASR model, the attack makes no use of any additional DNN models and is hence both technically and computationally less demanding than some other attacks.

A13: a zero-shot VC (ZSVC) system based on the StarGAN-ZSVC model (Baas and Kamper, 2020). The waveform encoder extracts a mel-spectrogram from the input utterance. The StarGAN (Miyato and Koyama, 2018) acoustic decoder then converts the mel-spectrogram so that it encodes the same linguistic content but the voice of the target speaker. The speaker embedding is extracted using a stack of recurrent layers using gated recurrent units (GRUs) (Cho et al., 2014). The vocoder is WaveGlow (Prenger et al., 2019), a normalising-flow-based waveform generation model. StarGANs are known to perform well in voice conversion tasks (Kaneko et al., 2019) and the StarGAN-ZSVC model is available as open-source (Baas and Kamper, 2020).

A14: a zero-shot neural TTS system built upon YourTTS (Casanova et al., 2022), which is itself based on VITS (Kim et al., 2021). A14 is a variant of A08. Advances include the higher number of Transformer blocks used in the text encoder and use of a RawNet3 speaker encoder (Jung et al., 2022b).

A15: a few-shot VC system (AlBadawy and Lyu, 2020). The waveform encoder, a CNN-based variational auto-encoder (VAE) (Kingma and Welling, 2014), is used to extract latent linguistic features from the mel-spectrogram of the input speech. They are used by a CycleGAN (Zhu et al., 2017) acoustic model to produce a new mel-spectrogram conditioned on the target speaker embedding. In contrast to other zero-shot VC systems, the acoustic model is fine-tuned to the target speaker using adaptation data and a cycle-consistency training loss (Zhu et al., 2017). The VAE encoder is nonetheless still pre-trained without fine-tuning. The vocoder is the CNN-based WaveNet model (Oord et al., 2016). CycleGAN-based systems were popular among submissions to the 2020 Voice Conversion Challenge (Zhao et al., 2020).

A16: a zero-shot VC system which uses a voice disentanglement technique (Sun et al., 2016). The input is first decomposed into speaker embeddings, linguistic latent vectors, and F0 features. The speaker encoder uses a context-aware masking (CAM++) model (Wang et al., 2023), a more efficient time-delay DNN variant of the ECAPA-TDNN model. Linguistic latent vectors are derived using a wav2vec 2.0 base model (Baevski et al., 2020) which is pre-trained using unlabelled data within the VoxPopuli dataset (Wang et al., 2021) and fine-tuned for ASR using the English subset. F0 dynamics are extracted using the Praat toolkit (Jadoul et al., 2018). During conversion, the speaker embedding is replaced with that of the target speaker. F0 features are linearly scaled to match the first and second statistical moments of those extracted from the adaptation utterance(s). The set of three representations is fed to a HiFi-GAN vocoder to generate a waveform, without use of an acoustic decoder. Similar disentanglement-based VC systems were among the top performing submissions to the 2020 Voice Conversion Challenge (Zhao et al., 2020).

3.4. Evaluation set

Spoofed data in the ASVspoof 5 evaluation set are generated using nine TTS/VC systems and seven adversarial attacks. Three of the TTS attacks, namely A17, A28, and A29, are generated using off-the-shelf systems pre-trained using external datasets. They are included to support the evaluation of detection performance when attack models are well trained using huge quantities of non-matching data. All other systems are trained in the same way as those described above and with the protocols described in Section 2. The TTS/VC systems are described first, followed by the adversarial attacks. A summary of each is presented in the two lower-most blocks of Table 2.

A17: a zero-shot TTS system, dubbed ZMM-TTS (Gong et al., 2024), originally designed for multi-speaker and multi-lingual applications. The text encoder comprises a BERT-like module (Nguyen et al., 2023) and a stack of feedforward blocks. The acoustic decoder, which uses a hierarchical vector-quantisation VAE (VQ-VAE) (Van Den Oord et al., 2017) architecture (Guo et al., 2023), and the HiFi-GAN-based vocoder are jointly optimised. Training data is sourced from multiple datasets in six languages.[17] The speaker encoder is an ECAPA-TDNN pre-trained using the VoxCeleb 2 dataset (Chung et al., 2018).

A19: a classical unit-selection TTS system based on the MaryTTS platform (Schröder et al., 2011) and the Voice Building Plugin (v5.4) (Steiner and Le Maguer, 2018). The adaptation utterance(s) is used to construct a pool of speech units for each target speaker. A19 is a more sophisticated variant of the A12 unit-selection attack. Speech units are of a non-uniform granularity (Sagisaka, 1988), ranging from individual diphones to segments which span consecutive diphones. The unit selection algorithm considers not only whether the selected units match the phoneme sequence of the input text but also the distortion introduced through concatenation (Hunt and Black, 1996). MaryTTS was used in the generation of previous ASVspoof databases (i.e., S10, ASVspoof 2015 and A04/A16, ASVspoof 2019). MarryTTS attacks are known to pose a threat to ASV reliability. There is also some evidence that they can be difficult to detect (Wu et al., 2015a; Wang et al., 2020; Jung et al., 2024), even if speech quality might not be as high as that for some of the more recent neural-based TTS approaches.

A21: a variant of the zero-shot TTS system A09. Enhancements include use of a vocoder based on the Big Vocoding GAN (BigVGAN) (Lee et al., 2022), a DNN based upon HiFi-GAN. Trainable periodic activation functions (Ziyin et al., 2020) are used to improve the quality of voiced sounds and anti-aliasing operations inside the GAN generator are used to reduce high-frequency artefacts in the generated waveforms.

A22: a zero-shot TTS system identical to A10, except for use of the same BigVGAN vocoder used by A21.

A24: a zero-shot VC system. An RNN-CNN waveform encoder is used to extract a phonetic posteriorgram (Kintzley et al., 2011; Sun et al., 2016), a latent representation of mostly speaker-independent phonetic information. The acoustic decoder, also a CNN, combines the posteriorgram with ECAPA-TDNN speaker embeddings to predict target-speaker-dependent latent features. A HiFi-GAN vocoder is then used to produce speech in the voice of the target speaker. All components are trained in an end-to-end manner. A24 is similar to A16 in that input speech is disentangled into speaker-dependent and speaker-independent features. A key difference to A16 is use of GAN-based, end-to-end training.

A25: a zero-shot, diffusion-based VC system (Popov et al., 2022). The Transformer-based waveform encoder extracts latent features resembling the mel-spectra of an 'average voice', i.e., mel-spectra averaged over the set of speakers in the training data. Latent features and speaker embeddings extracted from an RNN-GE2E speaker encoder (Wan et al., 2018) are then fed to the acoustic decoder and used as priors for a reverse diffusion process (Song et al., 2021). The resulting mel-spectrogram is then fed to a HiFi-GAN vocoder for waveform generation.

A26: a variant of the A16 zero-shot VC system. A denoiser (Défossez et al., 2020) is applied to the training data. During generation, noise contained in the input is extracted and added to converted speech at the output. The addition of background noise may lead to more convincing spoofs/deepfakes and/or act to mask conversion artefacts and hence increase the difficulty of detection.

A28: a pre-trained, zero-shot YourTTS (Casanova et al., 2022) system released with the Coqui toolkit (Eren and The Coqui TTS Team, 2021). The training data is in English, Brazilian Portuguese and French languages with no overlap with speakers in the ASVspoof 5 evaluation set. Different to A14, also based upon YourTTS, is use of an H/ASP-based speaker encoder (Heo et al., 2020)

[17] The training data includes 2.4k utterances collected from 10 target and 2 non-target speakers, for whom there is also data in the ASVspoof 5 evaluation set but not overlapped with the 2.4k TTS training utterances. The use of public datasets across multiple application domains is now widespread meaning that data overlap is increasingly more difficult to avoid. Bearing in mind that this represents less than 2% of the speakers in the evaluation set, the overlap simulates a worse-case scenario whereby training is performed using data collected from the target speaker. Nevertheless, experimental validation (Section 7.2) shows that A17 is not more effective in attacking the 'seen' target speakers than the 'unseen'.

pre-trained using the VoxCeleb 2 database (Chung et al., 2018). The publicly available Coqui implementation of YourTTS is a strong performer for zero-shot TTS tasks (Casanova et al., 2022).

A29: another pre-trained, zero-shot TTS system, named XTTS (Casanova et al., 2024). Designed for multi-lingual TTS and trained using a mix of four datasets containing speech in 16 languages, it uses the GPT-2 text encoder architecture (Radford et al., 2019), a HiFi-GAN vocoder, and an H/ASP speaker encoder (same as for A28). The XTTS system is also publicly available in the Coqui toolkit.

Adversarial attacks: The remaining attacks are all adversarial filtering attacks designed to increase the threat of a selection of TTS/VC systems. As illustrated to the right of Fig. 4, learnable parameters are optimised using gradients which are back-propagated from the compromised CM or ASV model. Once trained, adversarial filtering is applied to spoofed utterances in the evaluation set without further modification. There are three classes of attacks which are designed to compromise either CM or ASV systems or their combination.

- **A18, A20, A23**: a set of adversarial attacks which target an AASIST CM (described in Section 7.1). The coefficients of an adversarial, non-causal and trainable linear-time-invariant (LTI) filter named Malafide (Panariello et al., 2023) are optimised to increase the scores produced by the targeted CM[18] via a gradient ascent algorithm. As illustrated in Table 2, A18, A20 and A23 attacks result from the application of Malafide adversarial filtering with $L = 1025$ filter coefficients to utterances produced by A17, A12 and A09 TTS systems.
- **A27, A31, A32**: a set of adversarial attacks which target a CAM++ ASV model.[19] Speaker-specific filters, named Malacopula (Todisco et al., 2024), are composed of K parallel filtering branches, one purely linear and the others non-linear via static power polynomial functions $x^k, k \in [1, K]$, all with a trainable, non-causal LTI filter. Filter coefficients are optimised to minimise the difference between a pair of embeddings, the first extracted from a spoofed utterance, the second from an enrolment utterance collected from the target speaker. The optimisation acts to maximise ASV scores for the former and hence to provoke a greater rate of false accept decisions. As illustrated in Table 2, A27, A31 and A32 attacks result from the application of Malacopula filtering with either $L = 1025$ or 513 filter coefficients and either $K = 3$ or 5 branches to utterances produced by A26, A22 and A25 VC or TTS systems.
- **A30**: an adversarial attack which results from the application of Malacopula filtering with $L = 257$ components and $K = 3$ branches to utterances produced by the A18 attack which itself results from the application of Malafide filtering to utterances produced by the A17 TTS system. A30 is the only attack which involves the sequential application of both Malafide and Malacopula filtering.

Malafide and Malacopula implementations are identical to those described in Panariello et al. (2023) and Todisco et al. (2024) respectively which provide comprehensive descriptions of each approach.

4. Post processing

A subset of both bona fide and spoofed utterances are post-processed before being added to the ASVspoof 5 database. This includes the application of encoding/compression to utterances in the evaluation set and quality control to reduce shortcut artefacts.

4.1. Encoding and compression

To evaluate detection performance under bandwidth, encoding and compression variation, subsets of data within the ASVspoof 5 evaluation set, including both bona fide and spoofed utterances, were encoded or compressed according to one of the evaluation conditions listed in Table 3. Treatment of the evaluation set only helps to keep the database size manageable while also allowing users to choose their own encoding, compression, or any other data augmentation strategies for training and development pipelines. With the aim of maintaining a reasonable database size and facilitating the comparison of performance with and without encoding/compression for the same data, we treated a subset of 20% of the utterances in the evaluation set. For each utterance u in this subset, we created an exhaustive set of utterance-codec pairs, i.e., u-C00, u-C01, ⋯, u-C11. For each utterance v in the remaining 80% of data, we create only a single pair v-Cxx wherein Cxx is randomly selected from the 11 conditions in Table 3.

Condition C00 serves as a reference for which there is no encoding or compression. For conditions C00-C07, all data is sampled at a rate of 16 kHz. In contrast, conditions C08-C11 correspond to a narrow band setting under which all data is down-sampled to 8 kHz. Conditions C01-C11 all involve the application of lossy encoding/compression algorithms at the bitrates indicated in Table 3. C04 and C07 use the recent, deep-learning-based Encodec (Défossez et al., 2023) scheme. For C07, mp3 encoding is applied prior to Encodec to simulate a channel with double compression. Finally, C11 is an experimental setting in which utterances are linearly convolved with the pre-computed, non-linear responses of a set of eight end-to-end calling pipelines, namely calls made from a device to a call center platform over a public switched telephone network (PSTN). Responses were estimated using the synchronised swept sine approach described in Novak et al. (2015) using one of six different calling devices and one of four different audio injection methods. Specific configurations for C11 are detailed in Appendix A.

[18] By established ASVspoof convention, higher scores indicate support for the bona fide hypothesis.

[19] The ASV model is the CAM++ model from Wang et al. (2023). It is chosen only for the reasons that it is different to the baseline ASV system described in Section 7.1 and that its use hence supports evaluation in a black-box setting.

Table 3
Summary of codec and compression conditions in evaluation sets.

	Coding format	Bandwidth	Bitrate range (kbit/s)
C00	none	16 kHz	–
C01	opus	16 kHz	6.0–30.0
C02	amr	16 kHz	6.6–23.05
C03	speex	16 kHz	5.75–34.20
C04	Encodec (Défossez et al., 2023)	16 kHz	1.5–24.0
C05	mp3	16 kHz	45–256
C06	m4a	16 kHz	16–128
C07	mp3+Encodec	16 kHz	varied
C08	opus	8 kHz	4.0–20.0
C09	amr	8 kHz	4.75–12.20
C10	speex	8 kHz	3.95–24.60
C11	varied	8 kHz	varied

All data is distributed with a common sampling rate of 16 kHz; data initially downsampled to 8 kHz in conditions C08-C11 are upsampled to 16 kHz after encoding/compression. Although upsampling does not increase the effective bandwidth, distribution of data at a common sampling rate reduces the risk that users treat data sampled at 8 kHz with a detector trained using data sampled at 16 kHz.

4.2. Shortcut artefacts

Shortcut artefacts are semantically unrelated to the spoofing/deepfake detection problem but can nonetheless be utilised for detection. Usually the result of dataset collection or generation procedures, the problem of shortcut artefacts can be particularly insidious. When assessed in the laboratory using evaluation datasets contaminated by such artefacts, detection performance can be strong (Lapuschkin et al., 2019). But, when these same systems are deployed in the wild, conditions in which database artefacts cannot be relied upon, performance may degrade catastrophically.

To illustrate the problem, consider a set of TTS systems which set the peak waveform amplitude to specific, pre-set levels. The peak amplitude is then an informative indicator of the content having been generated by one of these same TTS systems. A detector which exploits the peak amplitude shortcut might perform well when it is assessed using similarly generated data, but will likely fail if, for example, a TTS system is configured to generate outputs with random peak amplitude. In reality, the peak amplitude is semantically unrelated to the detection problem; an utterance generated using TTS synthesis is a spoof/deepfake *regardless* of the peak amplitude.

We created a post-processing pipeline to detect and reduce the impact of potential shortcut artefacts. It was applied to the ASVspoof 5 evaluation and development sets, but *not* to the training set. This choice encourages database users to explore techniques which are able to extract the most semantically relevant cues while also avoiding the pitfalls of shortcut learning. Our analysis revealed five potential shortcut artefacts: the peak waveform amplitude; the durations of the leading and trailing non-speech segments (Chettri et al., 2020; Müller et al., 2021); the duration and energy of the whole utterance. The non-speech and speech segments of the utterance in the ASVspoof 5 evaluation set are annotated using the Whisper system (Radford et al., 2023). Fig. 5 shows the distribution of each shortcut artefact for bona fide utterances (black profiles) and spoofed/deepfake utterances (grey profiles) before post-processing. Distributions are shown in the first three rows for utterances generated using A17 (TTS), A21 (TTS), and A25 (VC) attacks. They show that, for A17 and A25 attacks, the peak amplitude is rescaled to 1.0, a distinct difference to the distribution for bona fide utterances. Whereas there is little difference between the distributions of leading non-speech segment durations for bona fide and spoofed/deepfake utterances for A25, they tend to be of shorter duration for A21. For A17, the duration of leading non-speech segments is uniformly distributed, even if the range is similar to that of bona fide utterances. Even so, the range in the duration of trailing non-speech segments for A17 differs to that for bona fide utterances. The duration of trailing non-speech segments for A21 is generally again shorter than those for bona fide utterances. The distribution of total duration for A25 overlaps with that of bona fide data. This is because A25, a VC system, does not alter the total duration of the input utterance. In the case of TTS-based attacks A17 and A21, the total duration is controlled by the acoustic model, hence the difference between distributions for bona fide and spoofed/deepfake utterances in these cases. There are also distinct differences in distributions of average energy for all three attacks.

The post-processing pipeline is designed to reduce the differences between the artefact distributions for spoofed/deepfake and bona fide utterances. Let us assume an input waveform $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ with N sampling points. First, the waveform is linearly scaled by a factor $r \in \mathbb{R}^+$ so that the peak amplitude $\max_n r\|x_n\|$ is equal to 1.0. Next, the duration of leading and trailing non-speech segments are trimmed, after annotation of segment boundaries using the Whisper system if the input is from the evaluation set, or an energy-based speech activity detector (Kinnunen and Li, 2010, §5.1) for the development set utterances. Given the annotated leading non-speech segment $(x_1, x_2, \ldots, x_{N_s})$ with N_s samples, an index $\tilde{n}_s$ is drawn at random from the probability mass function $\Pr(n) = \frac{\exp(-|x_n|)}{\sum_{i=1}^{N_s}\exp(-|x_i|)}, n \in [1, N_s]$. The sub-segment $(x_1, x_2, \ldots, x_{\tilde{n}_s})$ is then trimmed.[20] For the trailing non-speech segment, an index $\tilde{n}_e$

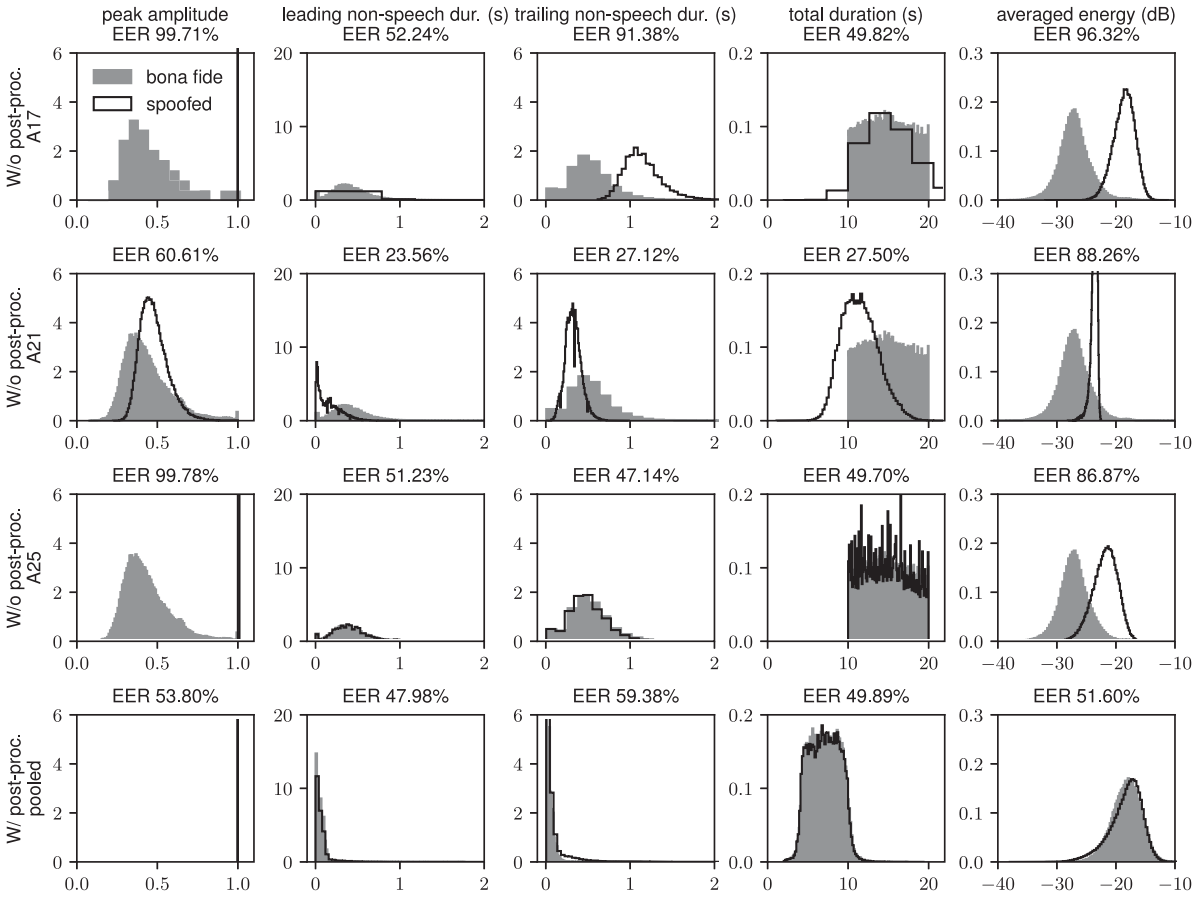

Fig. 5. Distributions of potential shortcut artefacts (different columns) **without (top three rows) and with post-processing (bottom row)**. Spoofed data without post-processing are from A17 (TTS), A21 (TTS), and A25 (VC). The bottom row shows the distribution from all the evaluation data after post-processing. The equal error rate (EER) shown at the top of each sub-figure is computed using the shortcut artefacts of the bona fide and spoofed utterances. An EER closer to 50% suggests a greater overlap between the distributions of the two classes.

is drawn at random in the same way so that the sub-segment $(x_{\tilde{n}_e}, x_2, \ldots, x_N)$ is trimmed. Finally, a float-valued duration is drawn at random from a uniform distribution between 4.0 and 10.0 s. A contiguous chunk of this duration is then selected from within the remaining speech segment as the pipeline output. There is no further scaling of the utterance energy since we observed similar energy distributions at the pipeline output for bona fide and spoofed/deepfake utterances without additional normalisation. All data is distributed in FLAC format with a sampling rate of 16 kHz. Immaterial metadata, such as the audiobook title, is removed lest it also serve as a shortcut to help distinguish between spoof/deepfake and bona fide utterances.

Plotted in the bottom row of Fig. 5 are distributions for each of the shortcut artefacts for the full evaluation set after post-processing, again for spoofed/deepfake and bona fide utterances. Post-processing reduces the discrepancies substantially. Even though the average energy is not manipulated directly, the distributions post-processing are near-to-identical, as they are too for each of the shortcut artefacts. Fully overlapping distributions indicate that post-processing successfully reduces shortcut artefacts linked to peak amplitude, non-speech and full utterance durations as well as average energy and provide assurances that they are unlikely to provide cues relevant to detection.

5. Visualisation

We present a visualisation of the database characteristics using t-distributed stochastic neighbour embedding (t-SNE) plots (Van der Maaten and Hinton, 2008) and a hierarchical clustering dendrogram (Hastie et al., 2009, Section 14.3). The objective is to highlight similarities and differences between bona fide and spoof/deepfake utterances.

[20] The time resolution of the segment boundary produced by the Whisper system (and the energy-based speech activity detector) is 10 ms on account of the stride of the input mel-spectrogram. The value of N_s is hence a multiple of 160 (= 10 ms × 16 kHz). We draw $\tilde{n}_s$ from Pr(n) so that there is randomness in the location of the trimming. Pr(n) gives a higher chance to select the index at samples where the waveform amplitude is low.

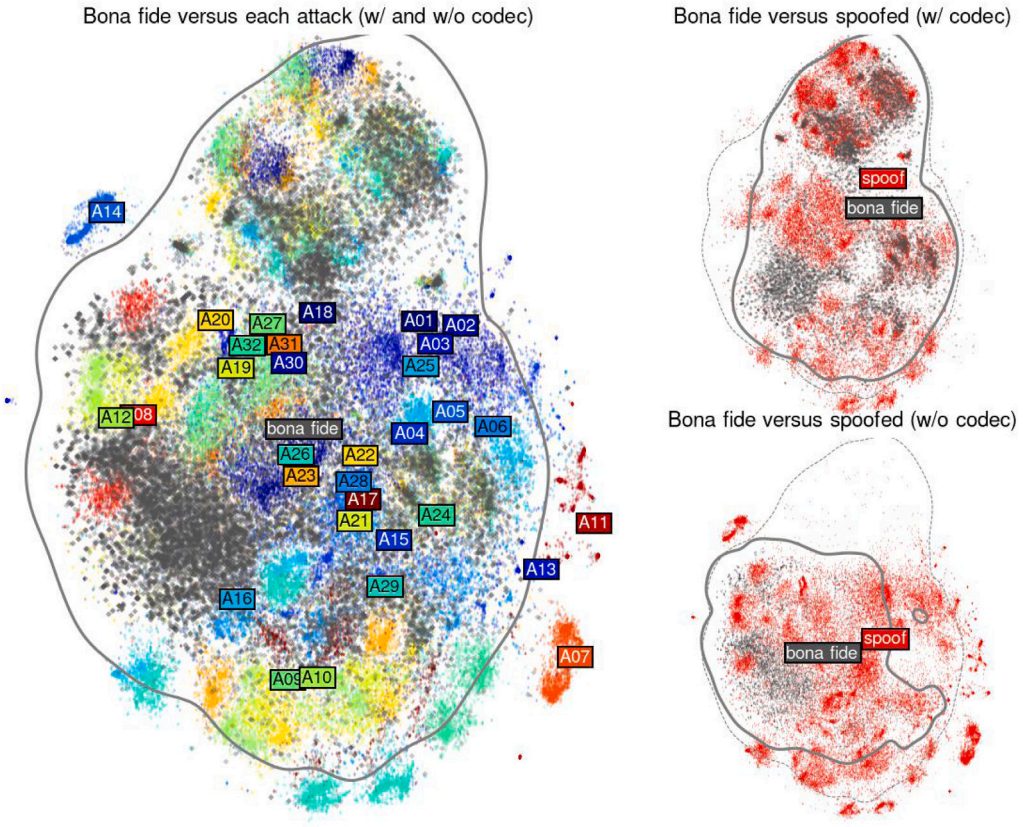

Fig. 6. Illustration of bona fide and spoofed samples from each attack in a t-SNE embedding space (sub-figure on the left). Dots in a grey colour correspond to the bona fide samples; others are spoofed samples. The contour encloses 98.9% of the probability mass of the bona fide samples estimated using a kernel density estimate method. The label of each data class is positioned at the mean of its corresponding data distribution. As alternative views, the top-right and bottom-right sub-figures, which plot all the spoofed samples in the same colour, show the data with and without codec/compression, respectively. The contours in solid lines encloses 98.9% bona fide samples in the sub-figures. For comparison, the contour from the sub-figure on the left side is plotted as dotted lines in the sub-figures on the right side.

Illustrated in Fig. 6 is a representation of the ASVspoof 5 database in the form of a 2-dimensional t-SNE plot (Van der Maaten and Hinton, 2008). Each point corresponds to an embedding extracted using an ASV system (the baseline system described in Section 7). Grey points correspond to bona fide utterances while coloured points correspond to one of the 32 attacks. To reduce computation time and visual clutter, points are shown for a random selection of 10% of the utterances in the training, development and evaluation sets, with and without encoding/compression for the evaluation set only. The recipe used to produce Fig. 6 is the same as that used to produce similar plots in Wu et al. (2017) and Wang et al. (2020) and involves attack-level, within-class covariance normalisation, length normalisation and speaker-level data whitening. Also plotted as a grey solid line in Fig. 6 is a confidence contour which encloses 98.9% of the probability mass of the bona fide data distribution. It is estimated using a kernel density estimate (KDE) method.[21]

Compared to a similar plot for the ASVspoof 2019 database presented in Wang et al. (2020, Fig. 3), a greater proportion of the attacks lie within the confidence contour. However, attacks A07, A11, and A14 lie outside of the contour, suggesting distinct differences to bona fide utterances. A13, another outlier, collapses into a cluster with low variance, indicating a lack of inter-speaker variation. Illustrated to the right in Fig. 6 is a pair of additional t-SNE plots, the difference between which shows the impact of encoding/compression. The contours are plotted in the same way as before. The overlap between bona fide and spoofed/deepfake utterances is greater with encoding/compression than without, indicating a greater challenge to detect spoofed/deepfake utterances.

Another expected finding is the apparent similarity between spoofing attacks based on the same generative technology, for example, {A01, A02, A03}, all neural TTS systems based on Glow-TTS, {A04, A05, A06}, all based on Grad-TTS, and {A09, A10}, both zero-shot TTS systems implemented using the IMS Toucan speech synthesis toolkit. Attacks {A01, A02, A03} (and {A04, A05, A06}) share the same acoustic model and vocoder, but use different types of speaker embedding. The use of different approaches

[21] We use the SeabornKDEAPI (Waskom, 2021). The confidence level of 98.9% was used in deriving a similar visualisation for the ASVspoof 2019 database (Wang et al., 2020, Figure 3), though in the form of a confidence ellipse.

Table 4
Evaluation metrics. Note that, in ASV EER, there is only one negative class of data, which can be either bona fide non-target or spoofed. In a-DCF, the bona fide non-target and spoofed data are treated as two independent negative classes. MOS EER is computed in a similar manner to CM EER but based on the output of the MOS estimator.

Metric	Input scores	Positive class	Negative class(es)
ASV EER	ASV scores	Bona fide AND target	(Bona fide AND non-target) OR Spoofed
CM EER	CM scores	Bona fide	Spoofed
a-DCF	SASV scores	Bona fide AND target	Bona fide AND non-target, Spoofed
MOS EER	MOS	Bona fide	Spoofed

to predict prosodic parameters (i.e., F0, energy, and duration) does not appear to cause substantial differences between A09 and A10. Use of a different vocoder architecture in an otherwise similar attack, as is the case for A09 versus A21 and A10 versus A22, leads to more substantial differences. This observation suggests that, at least in this case, the vocoder has a greater influence on the differences between bona fide and spoofed/deepfake utterances than the approach to predict prosodic parameters. More detailed analysis is complicated by the manifold differences between each attack algorithm.

Fig. 7 provides an alternative visualisation of the database in the form of a *dendrogram* obtained using an agglomerative hierarchical clustering approach, similar to that presented in Wang et al. (2020, Figure 3) for the ASVspoof 2019 database. The dendrogram is derived from the same set of speaker embeddings used in generating the t-SNE visualisation. Implementation details are described in Appendix B. The clustering is based upon pairwise cosine similarities between embeddings extracted from bona fide utterances and spoofed/deepfake utterances generated using each of the 32 attacks, a visualisation of which is depicted in the colour-scaled heatmap also shown in Fig. 7. The most similar utterances are indicated in dark blue, whereas the most different are indicated in dark red.

The dendrogram (and the similarity heatmap) show similar clusters of attacks as the t-SNE plot, e.g., {A01, A02, A03} and {A04, A05, A06}. The high similarities within these two groups of TTS systems may be because the three systems within each group use the same acoustic decoder, text encoder and vocoder, even if they use different speaker embedding extractors. However, the two sub-groups have low inter-group similarity — they are located in different branches of the dendrogram tree. We observe a similar clustering of {A09, A10, A21, A22}, all implemented using the ToucanTTS toolkit.

It is of interest to see whether or not the similarities between bona fide and spoofed/deepfake utterances and between spoofed/deepfake utterances generated with different algorithms correlate with those observed from automatic detection results. A discussion of this correlation is presented later in Section 7. We nonetheless stress that the visualisations presented above are derived using *oracle* speaker and attack labels which are unavailable during detection.

6. TTS and VC system optimisation using surrogate models

We describe the optimisation of a subset of TTS and VC systems using the surrogate models introduced in Section 2.4. Surrogates are based on open-sourced implementations of popular ASV and CM architectures. The surrogate ASV system is an ECAPA-TDNN (Desplanques et al., 2020) model with cosine scoring implemented using the SpeechBrain toolkit (Ravanelli et al., 2024). The system is trained using the VoxCeleb1 (Nagrani et al., 2017) training set and VoxCeleb2 (Chung et al., 2018) development set, with data augmentation based upon additive noises from the MUSAN dataset (Snyder et al., 2015) and reverberation from the RIR dataset (Ko et al., 2017). Surrogate CM systems include AASIST (Jung et al., 2022a), RawNet2 (Tak et al., 2021), and LCNNs with LFCC features (Todisco et al., 2019), all implemented using source codes provided by the respective authors. Training is performed from scratch using the ASVspoof 5 training set. The best checkpoint is selected based on the lowest EER for the surrogate development set.

Fig. 8 illustrates the increase in ASV and CM equal error rates (EERs) which three data contributors (of five in total) were able to achieve using ASV and CM surrogates. The four plots show EERs for each of the four surrogate models (1 ASV and 3 CM systems) and for a selection of three attacks {A10, A11, A24} the performance of which was evaluated using the surrogate models in multiple rounds (horizontal axis). The contributors decided the number of rounds and received results for their own attacks after each round. A10 contributors monitored the training process and selected the checkpoint that achieved the highest ASV EER. A11 contributors selected the checkpoint that achieved the highest overall surrogate CM EER. A24 contributors manually tweaked the training configuration of the speaker encoder (e.g. from independent training to joint training with the VC system) and selected the checkpoint that gave the highest surrogate ASV EER.

Results show that the surrogate models can be used successfully to implement stronger attacks. Use of surrogate systems among the data providers was nonetheless modest. This is likely due to the associated optimisation cost. It is not necessarily evident how TTS and VC systems should be modified, especially given that most systems are designed to maximise some form of quality-based measure instead of EERs. Still, some contributors were successful and it would be unwise to assume that an adversary will never optimise an attack in similar fashion.

7. Experimental validation

We describe ASV, CM, and SASV systems, experiments and results. These serve two purposes, namely to validate the protocol design and database collection and to provide baselines for the ASVspoof 5 challenge; a comparison of different baselines is not the

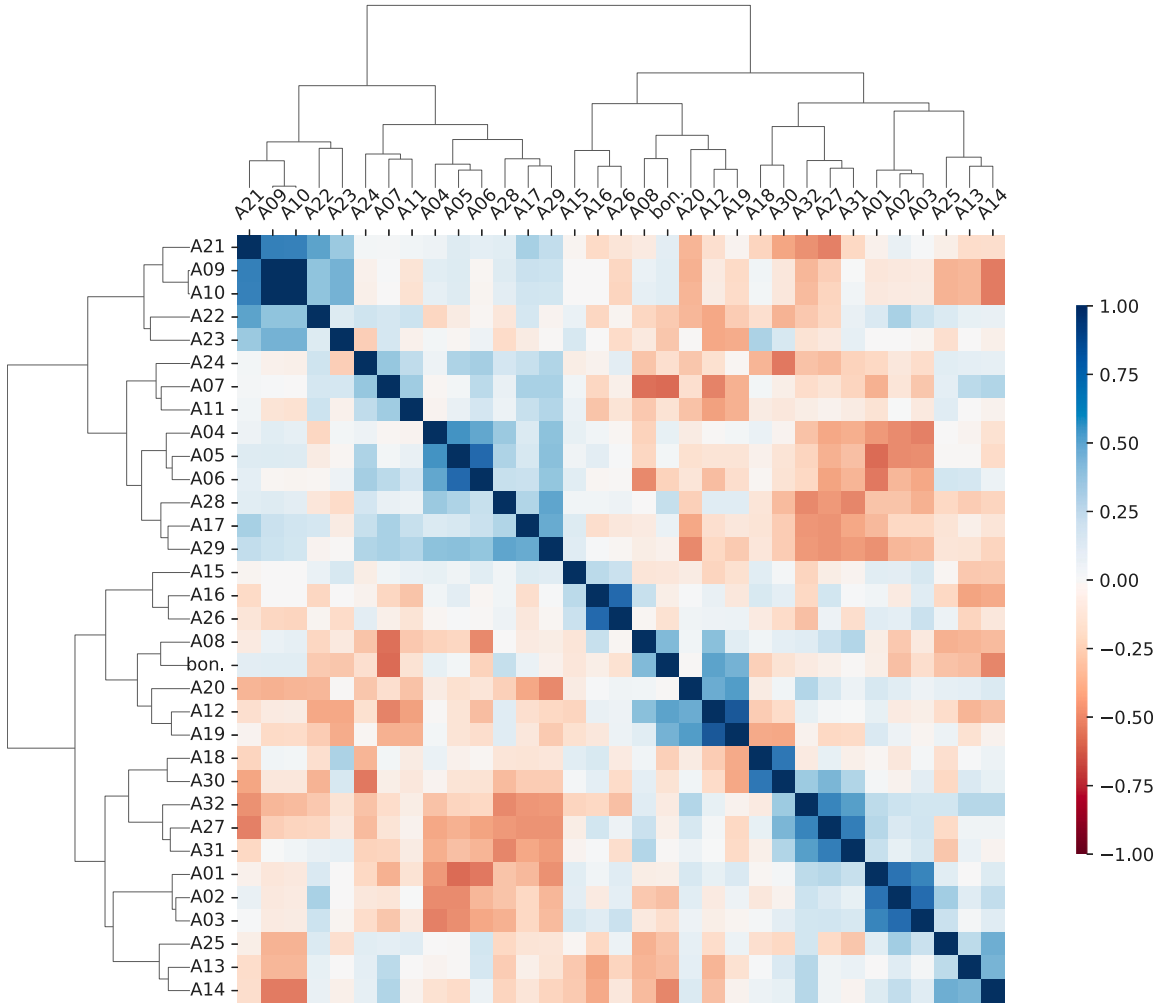

Fig. 7. Dendrogram (hierarchical cluster map) of attacks in the ASVspoof 5 database. For visualisation, heatmap plots the cosine similarity between the each pair of the speaker embeddings subsets. Implementations of the dendrogram and cosine similarity are described in Appendix B.

objective. The systems, all open-sourced, include one ASV system, a pair of CM systems, and a pair of SASV systems. As illustrated in Fig. 3 and described in Section 2.7, CM systems can be used on their own as solutions to Challenge Track 1. Solutions to Track 2 involve SASV systems, implemented either as a fusion of standalone CM and ASV subsystems or more 'end-to-end' approaches (as discussed in Section 2.7). Also used for validation is an open-sourced mean-opinion-score (MOS) estimator which was used to predict the perceptual quality of bona fide and differently-generated spoofed/deepfake utterances.

7.1. Baselines, MOS estimator, and metrics

The baseline ASV system is used to gauge the threat of each attack in terms of spoofing target speakers. It is based upon another implementation of ECAPA-TDNN (Desplanques et al., 2020)[22] speaker encoder, with a cosine scoring backend. The speaker encoder is trained on VoxCeleb 2 (Chung et al., 2018) development set, with data augmentation based upon additive noises (Snyder et al., 2015), room reverberation (Ko et al., 2017), and masking of input spectral features (Park et al., 2019). There are two CM baselines, RawNet2 (Jung et al., 2020; Tak et al., 2021) (B01) and AASIST (Jung et al., 2022a) (B02). Both systems are trained in end-to-end fashion, as illustrated at the left to Fig. 3. These models operate directly on raw waveforms and require a fixed length 4-s audio input. RawNet2 uses a fixed bank of 20 sinc filters (Ravanelli and Bengio, 2018) and six residual blocks with gated recurrent units (GRUs) to convert frame-level representations into utterance-level representations. Output scores are generated using fully connected layers. AASIST uses a RawNet2-based encoder (Jung et al., 2020) to extract spectro-temporal features from the raw input waveform.

[22] https://github.com/TaoRuijie/ECAPA-TDNN

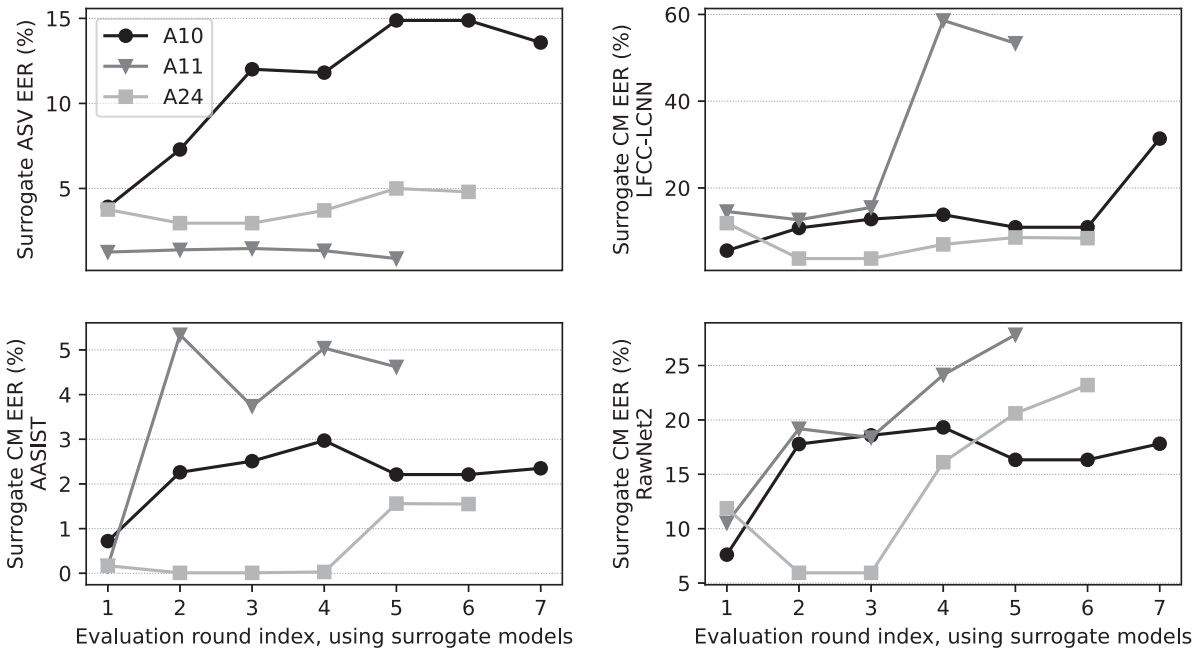

Fig. 8. Progress in TTS and VC system optimisation, measured using the EERs of the surrogate models.

Spectro-temporal heterogeneous graph attention layers and max graph operations are then used to integrate temporal and spectral representations. Output scores are generated using a readout operation and a fully connected output layer.

The SASV systems are an ASV-CM fusion-based system (Jung et al., 2022c) (B03, middle panel of Fig. 3), and an end-to-end system (B4 (Mun et al., 2023), right panel of Fig. 3). B03 is the SASV 2022 challenge baseline (Jung et al., 2022c; Shim et al., 2022) and is an LLR-based fusion (Wang et al., 2024b) of the ASV and AASIST CM baselines described above. B04, based on MFA-Conformer (Zhang et al., 2022), extracts a single embedding from the input waveform and produces a single output score. It is trained in three stages: pre-training for speaker classification; copy synthesis training (Wang and Yamagishi, 2023) with adapted SASV loss functions; in-domain fine-tuning using ASVspoof 5 training data.

A MOS predictor (Cooper et al., 2022) is used to estimate the quality of bona fide and spoofed utterances as might be measured from human listening tests. It produces a MOS on a five-point scale for each input utterance. The system architecture is the combination of a self-supervised-learning-based wav2vec 2.0 (Baevski et al., 2020) front-end, a global-average pooling layer, and a linear output layer. We used the publicly available, pre-trained system[23] without modification. This time we did not recruit human listeners to assess the quality due to the prohibitively high manpower cost.

As depicted in Table 4, we used the same EER metric for the evaluation of ASV and CM performance. The ASV EER is usually estimated from a set of bona fide target trial scores (positive class) and bona fide non-target trial scores (negative class). When subjected to attacks, the latter are replaced by spoofed target scores.

The CM EER is estimated using a set of bona fide and spoofed trial scores. SASV performance estimates are expressed in terms of the architecture-agnostic detection cost function (a-DCF) (Shim et al., 2024; Delgado et al., 2024), a cost-based performance metric designed specifically for the SASV task. While the SASV makes a binary decision, there are now three classes, namely trials involving bona fide targets (positive class) and then bona fide non-targets and spoofed target trials (negative class). Lower a-DCF values indicate better performance.[24] Further details for the evaluation metrics are available in the ASVspoof 5 challenge evaluation plan (Delgado et al., 2024).

The MOS EER is computed in similar fashion to the CM EER but using predicted MOS values for bona fide and spoofed utterances. Higher MOS EERs indicate spoofed utterances of (predicted) perceptual quality more comparable to that of bona fide utterances, while lower MOS EERs suggest that spoofed utterances are perceptually inferior to, and more easily distinguishable from bona fide utterances. We report the MOS EERs rather than the attack-level MOS mean values because the former gauges the overlap of the spoofed and bona fide MOS score distributions. One practical benefit is that, unlike the predicted MOS values (that may exhibit systematic domain shift), EER is invariant to any order-preserving transforms of the predicted MOS scores (including global scaling and shifting) (Van Leeuwen and Brümmer, 2007, § 3.1).

[23] https://github.com/nii-yamagishilab/mos-finetune-ssl

[24] Implementations of all baseline systems and evaluation metrics are accessible from the ASVspoof~5Githubrepository.

Table 5
Performance of ASV and CM baselines in terms of EER (%) and SASV baselines in terms of a-DCF on development set (A09–A16) and evaluation set (A17–A32). Results are shown for ASV (ECAPA-TDNN), CM baselines: B01 (RawNet2) and B02 (AASIST), SASV baselines: B03 (Fusion-based) and B04 (single SASV system). Pooled EER and min a-DCF are computed from pooled scores across all attacks. EERs based on the MOS scores from the MOS estimator are listed in the rightmost column. It is computed from the MOS for bona fide and spoofed utterances and hence reflects classification performance when performed using estimated quality scores alone. A darker cell colour indicates a higher EER or min a-DCF value. ASV EERs discriminating bona fide data of target and non-target speakers are listed in the rows marked by *Non-target*.

			ASV (EER %)	Track 1 (EER %)		Track 2 (min a-DCF)		MOS (EER %)
		Attacks		B01	B02	B03	B04	
Development set		Non-target	1.88					
		A09	16.97	16.79	7.14	0.4016	0.2009	23.78
		A10	16.96	16.95	7.15	0.4027	0.1997	22.53
		A11	2.13	15.17	4.81	0.0494	0.0272	13.55
		A12	37.10	78.86	78.90	0.8465	0.8549	1.82
		A13	1.49	40.18	14.43	0.0353	0.0216	7.81
		A14	1.47	21.43	1.18	0.0333	0.0221	0.07
		A15	6.20	24.03	12.93	0.1449	0.0746	18.70
		A16	5.90	33.34	19.31	0.1422	0.0662	6.69
		Pooled	14.66	29.49	17.83	0.3156	0.2254	14.31
Evaluation set		Non-target	5.22					
	TTS/VC	A17	44.64	22.58	16.44	0.8818	0.5598	43.07
		A19	49.80	63.75	59.99	0.8881	0.9450	13.05
		A21	30.74	25.67	17.05	0.7187	0.3647	37.27
		A22	27.52	24.50	17.63	0.6402	0.4731	30.05
		A24	21.08	23.61	13.35	0.4846	0.3363	33.10
		A25	9.96	29.78	21.01	0.2373	0.3325	9.15
		A26	15.77	41.95	31.35	0.3765	0.4781	20.86
		A28	38.44	39.43	32.10	0.8519	0.8133	55.96
		A29	31.88	17.54	8.93	0.7328	0.3163	49.60
	Adversarial	A18	34.28	57.64	50.02	0.7434	0.7223	9.16
		A20	35.82	46.95	33.70	0.7244	0.8794	1.34
		A23	27.13	30.35	32.78	0.6420	0.5015	12.97
		A27	34.68	38.85	27.60	0.6994	0.5662	1.10
		A30	41.12	42.26	42.85	0.7798	0.6737	3.37
		A31	37.14	33.22	27.48	0.7220	0.5641	5.31
		A32	29.30	29.78	19.50	0.6299	0.5481	0.80
		Pooled	32.11	36.04	29.12	0.6806	0.5741	26.55

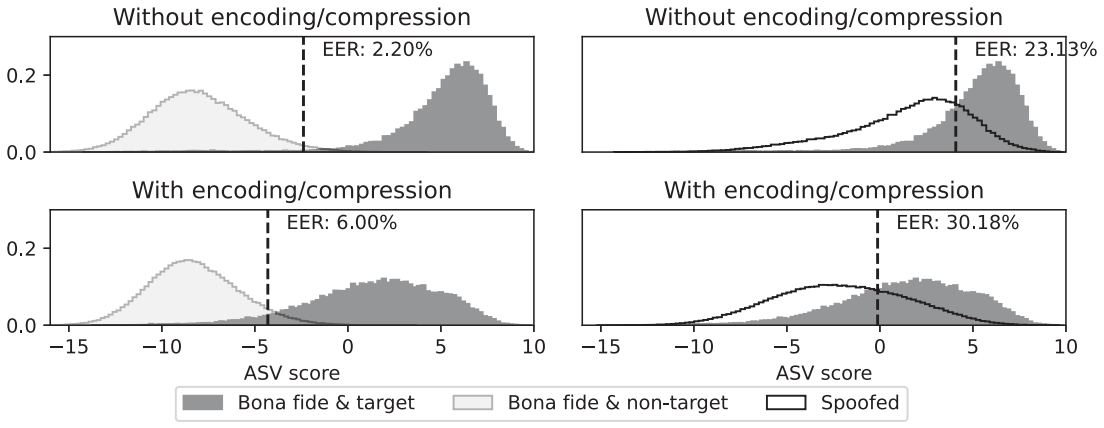

Fig. 9. Distribution of ASV scores on the evaluation set when the trials are processed with encoding/compression (top-row) or without (bottom-row). Sub-figures in the left column show the distributions for bona fide utterances from target and non-target speakers. Sub-figures in the right column show the distributions for bona fide utterances from target speakers and spoofed utterances from all the attacks. The thresholds producing the EERs are plotted as vertical dash lines.

7.2. Impacts upon ASV

In the following we report an evaluation of the baseline ASV system when it encounters bona fide target and bona fide non-target trials only, and then when it is subject to attack. We report EERs only in the following, readers with an interest in performance at other operating points are referred to detection error tradeoff (DET) profiles (Martin et al., 1997) presented in Appendix C.

ASV results for the development and evaluation sets are listed in the second column of Table 5. When assessed using a mix of only target and non-target trials the EER is 1.88% for the development set and 5.22% for the evaluation set. The higher EER for the evaluation set is due mostly to the variation in encoding/compression and bandwidth. The effect of encoding/compression is illustrated in the score histograms shown in the left column on Fig. 9. They show score distributions without (top) and with (middle) encoding/compression, along with their EERs of 2.20% and 6.00%, respectively. Encoding/compression reduces the differences between the distributions resulting in a larger overlap and therefore a higher EER.

Also shown in the second column of Table 5 are EERs for the same ASV system when evaluated using a mix of target and spoofed/deepfake trials. When the ASV system is under attack, EERs are almost universally higher, indicating that most attacks are successful in provoking higher scores. A11, A13, and A14 attacks, all in the development set, are relatively ineffective and lead to a trivial increase or even a reduction in the EER. This finding is consistent with observations made from the t-SNE visualisations shown in Fig. 6 wherein the same three attacks are among the outliers and far from the centroid of the cluster for bona fide utterances. A12, a simplified unit-selection based TTS system, generates the highest EER of 37.10% for the development set.

Attacks in the evaluation set result in some of the highest EERs. Of the 16 attacks, 14 lead to EERs higher than 20%. A19, also a legacy, unit-selection based TTS system, is once again the attack which provokes the highest EER. These results are in line with previously reported findings (Wang et al., 2020; Jung et al., 2024), a challenge recognised since the very first ASVspoof challenge (Wu et al., 2015a, 2017). A decade after, this persistent challenge is worthwhile emphasising: *the most effective attack algorithms are not necessarily those implemented with the latest technology*. Legacy TTS systems, based on the concatenation of bona fide speech segments extracted from the utterances of a target speaker are particularly effective.

Pre-trained TTS attacks A17, A28, and A29 give the next highest EERs, all above 31%.[25] A21 and A22, both trained using only data specified in the ASVspoof 5 contributors' protocol, also provoke high EERs. VC-based attacks A24, A25 and A26 are among the least effective. The effect of encoding/compression for the evaluation set is illustrated to the right in Fig. 9, which shows score distributions for bona fide target and spoofed/deepfake trials. As one might expect, once again, encoding/compression reduces the differences between the distributions, provoking higher EERs.

Results for the set of adversarial attacks A18, A20, A23, A27, A30, A31, and A32 shown to the bottom of Table 5 also cause a significant degradation in ASV performance. Malacopula adversarial attacks, designed to compromise ASV systems, result in higher EERs than their underlying TTS/VC attacks (A26 15.77% → A27 34.68%, A22 27.52% → A31 37.14%, A25 9.96% → A32 29.30%). Malafide attacks, which are not designed to compromise ASV systems, are ineffective with one exception. For A23, the EER increases to 27.13% from the 16.97% EER for the underlying A09 attack.

In the following we seek to determine if the attacks might also be more difficult to detect using a CM designed, unlike ASV systems, specifically for this task.

7.3. CM results

Table 5 shows performance of both CM baselines, B01 and B02, in terms of the EER (%). For the development set, the unit-selection based TTS attack A12 is the most challenging to detect.[26] Even though the t-SNE plot in Fig. 6 shows that A13 is an outlier, it is the second most difficult to detect for B01, while the EER for B02 is much lower. For B02 the second most difficult to detect is A16, a zero-shot VC system, for which the EER is also high for B01. All other attacks in the development set produce lower EERs for each CM, but many are still high. Pooled EERs of 29.49% and 17.83% show that neither CM is especially reliable in detecting the full set of attacks.

EERs for attacks in the evaluation set are generally higher, increasing to 36.04% and 29.12%. Higher EERs are expected given that the evaluation set includes more advanced attack algorithms as well as a large portion of encoded/compressed data. Interested readers can refer to Table D.7 for a breakdown of results with and without encoding/compression. Those shown in Table 5 are for pooled results derived using the ASVspoof 5 evaluation set.

Last, except for attack A20, Malafide adversarial filtering, which is applied to already-spoofed/deepfake utterances to further compromise the CM, leads to additional increases in the EER. As shown in Table 5, Malafide filtering provokes increases in the EER in the cases of A17 {22.58%, 16.44%} → A18 {57.64%, 50.02%}, A09 {16.79%, 7.14%} → A23 {30.35%, 32.78%}. The comparison of results for A17 and A30 shows that the combination of Malafide and Malacopula filtering also provokes increases in the EER. However, the application of Malacopula alone (designed to further compromise an ASV system), generally fails to increase the CM EER. Only for A31 does Malacopula filtering increase the EER beyond that for the underlying TTS attack A22.

7.4. SASV results

Thus far we have seen that while some attacks, e.g. A13 and A14, fail to provoke high ASV EERs, they can be difficult to detect, especially in the case of baseline B01. For other attacks, e.g. A12, ASV and CM EERs are both high, while for others they are both

[25] A small portion of A17's training data consists of utterances from 12 target speakers but does not overlapped with the ASVspoof 5 evaluation set (Section 3.4). The ASV EER computed on the bona fide and spoofed data of the 'seen' 12 target speakers is 42.51%, which is lower than that of the 'unseen' (44.65%). The high ASV EER provoked by A17 is not due to the 'seen' target speakers.

[26] A12 provokes EERs higher than 50%. An EER higher than 50% means that the majority of CM scores for spoofed/deepfake utterances are higher than those for bona fide utterances. By tradition, higher CM scores indicate greater support for the bona fide hypothesis. Therefore, from the perspective the baselines CMs, the spoofed/deepfake utterance of A12 are more likely to be 'bona fide' than the actual bona fide utterances.

relatively low, e.g. for A14 and baseline B02. For all of these cases there are separate ASV and CM sub-systems. In the following we explore the reliability of SASV systems which produce a single score. Results are presented for B03 and B04 SASV systems in Table 5 in terms of the min a-DCF (Shim et al., 2024). Again, lower a-DCF values indicate better performance. Results indicate that both SASV baselines struggle with unit-selection based TTS attacks A12 and A19. For both attacks and both baselines, the min a-DCF values are the highest for the development and evaluation sets. The A28 attack, based upon the recent YourTTS system, also produces high min a-DCF values, again for both baselines, indicating the higher threat of more sophisticated attacks.

For both the development and evaluation sets, attacks that provoke the lowest increase in ASV EERs, e.g. A25 and A26, also provoke the lowest min a-DCF values, even when CM EERs are relatively high. This implies that the B03 baseline, which is based upon the score-level fusion of independent CM and ASV systems, is reasonably successful in combining the merits of each in protecting against spoofs/deepfakes. Malacopula attacks A27, A31, and A32 result in higher min a-DCF values. In contrast, for B03, Malafide attacks A18, A20, and A23 fail to increase min a-DCF values beyond those of the underlying attacks.

Even so, pooled min a-DCF values remain high for both baselines and are substantially higher for the evaluation set than for the development set: pooled min a-DCF values for the development set are 0.3156 and 0.2254, while those for the evaluation subset are 0.6806 and 0.5741. A similar trend is observed under encoding/compression conditions (See Table D.7), where the a-DCF values on the evaluation set are even higher. These results indicate the challenge to maintain reliability when faced with the most advanced attacks and in the condition of encoding/compression. We would also like to remind the reader that the *minimum* a-DCF provides *the most optimistic* view of evaluation set performance. In particular, similar to EER, the minimum a-DCF metric involves an 'oracle' detection threshold obtained using ground truth labels. While useful for analysis of results, such as those presented in Table 5, the ground truth labels are *never available* in real-world operational settings (otherwise, why bother about developing classifiers?). The threshold(s) must be selected in advance at the system development time. Substantial future challenge remains on this front, particularly when we are faced with novel attacks and coded/compressed data. Investigation of the decision threshold is left to an upcoming analysis paper on the ASVspoof 5 challenge results.

7.5. MOS results

Estimated MOS EERs for each attack are shown in the rightmost column of Table 5. Lower MOS EERs suggest that spoofed data is of perceptually lower quality than bona fide data. As one might have expected, the results indicate that spoofed/deepfake utterances of lower (estimated) perceptual quality are generally easier to detect, whereas those of higher quality present a greater challenge. Pre-trained TTS system A28 produces the highest of all MOS EERs. The same attack also provokes a high ASV EER, high CM EER, and high SASV min a-DCF value. Other TTS/VC attacks, namely, A17, A21, A22, A24, and A29, also lead to poor performance of baselines in the case of all metrics, even if A29 is detected by B02 with an CM EER below 10%.

There are some exceptions, however. Some attacks with a low MOS EER nonetheless pose a substantial threat. For the legacy unit-selection TTS system A12 the MOS EER is less than 2%, though it provokes CM EERs higher than 78%. This finding suggests that, while human listeners may be able to distinguish easily between bona fide utterances and spoofs/deepfakes generated using the A12 attack, the challenge for automatic systems is substantial. We observe similar results for A19 and the full set of adversarial attacks. Casual listening test reveal notable concatenation artefacts in the case of A12 and A19, and the comparatively poor quality of adversarial attacks. The quality of speech produced by concatenative TTS approaches is limited by the number of speech units available among adaptation utterances (Section 3.4) while the filter coefficients of the two adversarial attacks are optimised to misguide only automatic systems—but not to preserve perceived quality (Todisco et al., 2024). The combination of quality-based estimates with more traditional CM techniques may lead to more generalisable and reliable detection techniques.

8. Conclusions

We described the design, collection and validation of the ASVspoof 5 database. Adoption of a new source database, itself collected by volunteers in the wild, gives rise to new challenges, involving both the generation of spoofed and deepfake speech as well as in detection. The ASVspoof 5 database contains greater acoustic diversity and speech data collected from a vastly greater number of speakers than any previous ASVspoof database. Spoofed/deepfake speech, generated with a mix of legacy and contemporary text-to-speech synthesis, voice conversion and adversarial attack technology, is also crowdsourced, further adding to variability. The ASVspoof 5 database supports the evaluation of automatic speaker verification systems when subjected to attacks, of independent spoofing/deepfake detection solutions, and of spoofing-robust automatic speaker verification systems of almost any architecture.

Comprehensive database validation experiments and results show the challenges to develop robust, generalisable detection, especially when faced with data encoded and compressed using the latest neural codecs. Despite the use of more challenging source data, both legacy and contemporary generative technology still pose a grave threat to the reliability of automatic systems, and show the need for continued progress in spoofed/deepfake detection.

The protocols and other resources described in this paper will be helpful to the participants of the ASVspoof 5 challenge who wish to analyse their results, as we hope it will be to others seeking to use the database to support other work in spoofing/deepfake detection. Most of the resources described in the paper are all already freely available to the community and can be used to reproduce our results. Being mindful of the obvious ethical considerations and potential for their misuse, the generation protocols and access to surrogate systems are available only upon request. They should be used to help the spoofing/deepfake detection community track future developments in generative technology, e.g. by contributing data produced using algorithms which emerge in the future, or for other beneficial and harmless applications.

CRediT authorship contribution statement

Xin Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Héctor Delgado:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hemlata Tak:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jee-weon Jung:** Writing – review & editing, Writing – original draft, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hye-jin Shim:** Writing – review & editing, Writing – original draft, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Massimiliano Todisco:** Writing – review & editing, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ivan Kukanov:** Writing – review & editing, Validation, Software, Resources, Investigation, Formal analysis, Data curation, Conceptualization. **Xuechen Liu:** Writing – review & editing, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization. **Md Sahidullah:** Writing – review & editing, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Tomi Kinnunen:** Writing – review & editing, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Nicholas Evans:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Kong Aik Lee:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Junichi Yamagishi:** Writing – review & editing, Validation, Project administration, Investigation, Conceptualization. **Myeonghun Jeong:** Writing – original draft, Resources, Data curation. **Ge Zhu:** Writing – original draft, Resources, Data curation. **Yongyi Zang:** Writing – original draft, Resources, Data curation. **You Zhang:** Writing – original draft, Resources, Data curation. **Soumi Maiti:** Writing – original draft, Resources, Data curation. **Florian Lux:** Writing – original draft, Software, Resources, Data curation. **Nicolas Müller:** Writing – original draft, Resources, Data curation. **Wangyou Zhang:** Writing – review & editing, Writing – original draft, Resources, Data curation. **Chengzhe Sun:** Writing – original draft, Resources, Data curation. **Shuwei Hou:** Writing – original draft, Resources, Data curation. **Siwei Lyu:** Visualization, Data curation. **Sébastien Le Maguer:** Writing – original draft, Software, Resources, Data curation. **Cheng Gong:** Writing – original draft, Software, Resources, Data curation. **Hanjie Guo:** Writing – original draft, Software, Data curation. **Liping Chen:** Supervision, Software, Data curation. **Vishwanath Singh:** Writing – original draft, Software, Data curation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Xin Wang reports financial support was provided by Japan Science and Technology Agency. Nicholas Evans reports financial support was provided by French National Research Agency. Tomi Kinunnen reports financial support was provided by Academy of Finland. Xin Wang reports equipment was provided by TSUBAME supercomputer grid, Institute of Science Tokyo.

K.A.Lee and M. Sahidullah are Editorial Board members of computer speech & language If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The ASVspoof 5 organising committee expresses its gratitude and appreciation to other contributors: Hengcheng Kuo and Hung-yi Lee, National Taiwan University; Yihan Wu, Renmin University of China; Yu Tsao, Academia Sinica; Minki Kang, KRAFTON, Korea.

This work is partially supported by JST, Japan, PRESTO, Japan Grant Number JPMJPR23P9, JST AIP Acceleration Project Grant Number JPMJCR24U3, Japan, and with funding received from the French Agence Nationale de la Recherche (ANR) via the BRUEL (ANR-22-CE39-0009) and COMPROMIS (ANR22-PECY-0011) projects. This work was also partially supported by the Academy of Finland, Finland (Decision No. 349605, project "SPEECHFAKES"). Part of the computation and data generation was performed using the TSUBAME4.0 supercomputer at Tokyo Institute of Science, and by supercomputing infrastructure provided by CSC — IT Center for Science (Finland).

Appendix A. Encoding and compression condition C11

See Table A.6.

Appendix B. Dendrogram of ASVspoof 5 data

Fig. 7 is plotted with the following configurations. We first compute a 33×33 pairwise similarity matrix $\boldsymbol{S} = [s(X_i, X_j)]$ that represents pairwise similarity between collections of speaker embeddings corresponding to each attack (or bona fide utterances).

Table A.6
C11 configurations.

ID	Device type	Calling device/software	Audio injection method
C11–1	PC	Microsoft Teams	Virtual audio cable driver
C11–2	PC	Microsoft Teams w/ noise cancellation	Virtual audio cable driver
C11–3	Smartphone	Poco Phone 4 5G	Bluetooth
C11–4	Smartphone	Redmi Note 8 Pro	Bluetooth
C11–5	Smartphone	Redmi Note 8 Pro	Analog cable
C11–6	Smartphone	Samsung Galaxy A12	Bluetooth
C11–7	Smartphone	Samsung Galaxy A12	Analog wired connection
C11–8	Smartphone	Samsung Galaxy S23 Ultra	Bluetooth

This similarity is computed as

$$s(X_i, X_j) = 1 - \frac{1}{2}\left(\frac{1}{|X_i|}\sum_{\boldsymbol{a}\in X_i}\min_{\boldsymbol{b}\in X_j} c(\boldsymbol{a},\boldsymbol{b}) + \frac{1}{|X_j|}\sum_{\boldsymbol{a}\in X_j}\min_{\boldsymbol{b}\in X_i} c(\boldsymbol{a},\boldsymbol{b})\right), \tag{B.1}$$

where $|\cdot|$ denotes the number of observations and $c(\boldsymbol{a},\boldsymbol{b})$ is cosine distance of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The similarity matrix $\boldsymbol{S}$ is used for hierarchical agglomerative clustering, using the Ward's minimum variance criterion as the clustering objective. The figure is rendered using the Seaborn `clustermap` API (Waskom, 2021).

Appendix C. DET curves

See Fig. C.10.

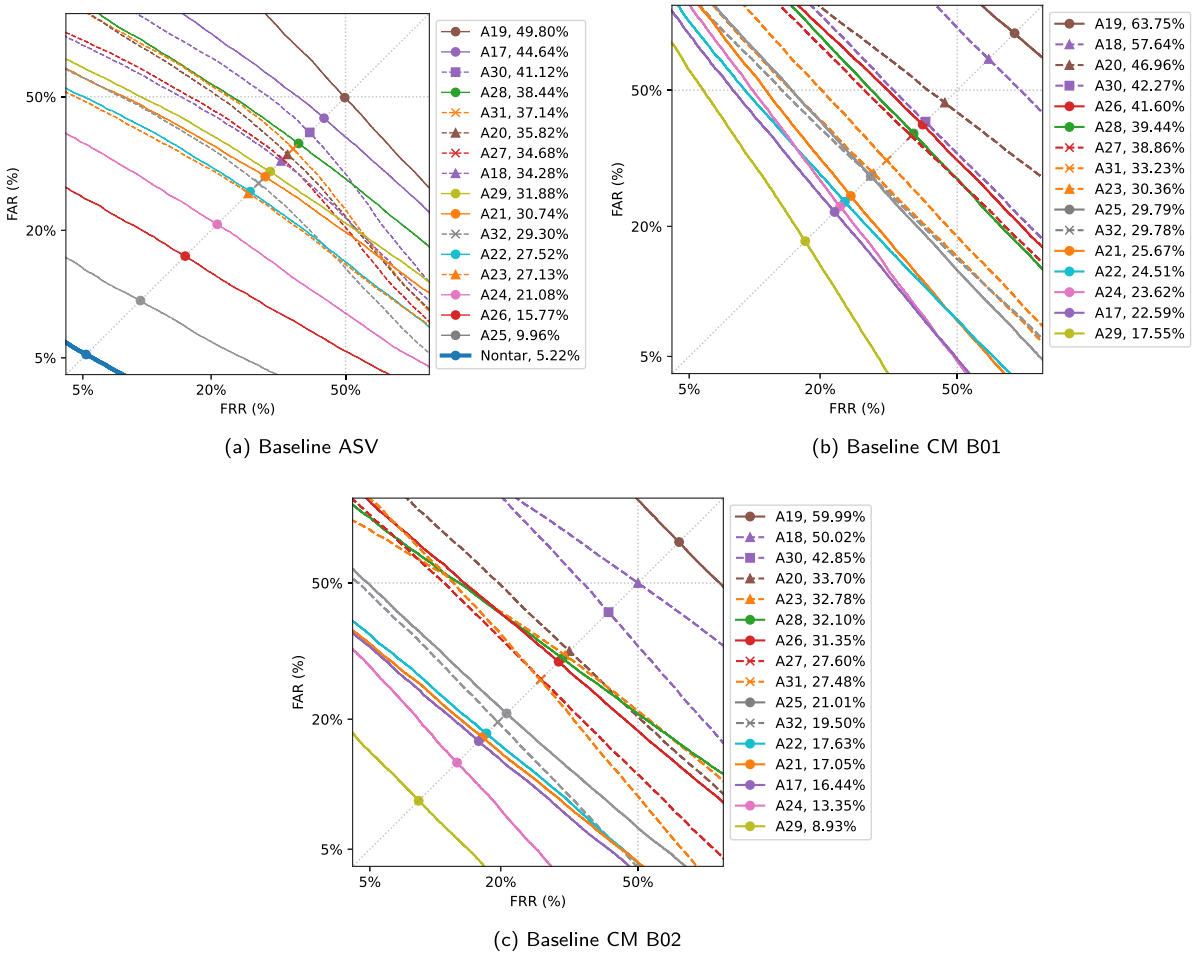

Fig. C.10. DET curves of ASV and CM baselines illustrating performance across different attacks in the evaluation set. Solid lines are zero-effort (from non-target speakers) and non-adversarial attacks. Dashed lines are adversarial attacks, including Malafide, Malacopula, and their combination. The colour of the dashed line is consistent with the non-adversarial attack counterpart. EER operating points of non-adversarial, Malafide, Malacopula, and combination of Malafide and Malacopula attacks are marked with •, ▲, ×, and ■, respectively. Numbers listed in the legend are EER values. The order of attacks in the legend is sorted based on the EER.

Table D.7
Performance of ASV and CM baselines in terms of EER (%) and SASV baselines in terms of a-DCF on evaluation set (A17–A32), with and without encoding/compression. The results are presented in a similar fashion to Table 5 except that the upper and bottom sub-tables show the breakdown results in conditions with and without encoding/compression, respectively.

			ASV (EER %)	Track 1 (EER %)		Track 2 (min a-DCF)		MOS (EER %)
		Attacks		B01	B02	B03	B04	
Evaluation set **without** encoding/compression		Non-target	2.20					
	TTS/VC	A17	37.44	14.48	8.30	0.8486	0.2032	23.02
		A19	52.53	66.46	62.60	0.8708	0.9053	3.34
		A21	16.19	17.27	6.63	0.3873	0.1060	17.94
		A22	13.72	18.72	8.05	0.3263	0.2018	13.30
		A24	9.31	15.36	3.63	0.2141	0.1065	15.11
		A25	3.51	20.48	9.76	0.0814	0.0813	1.70
		A26	6.01	36.31	20.03	0.1423	0.1571	6.79
		A28	26.67	33.42	23.17	0.6305	0.4794	39.76
		A29	17.12	5.88	1.19	0.3988	0.0821	30.72
	Adversarial	A18	21.96	53.24	52.24	0.5279	0.4029	1.53
		A20	27.21	48.54	30.72	0.6486	0.8789	0.09
		A23	13.33	25.08	30.02	0.3236	0.2321	2.47
		A27	23.42	37.94	22.91	0.5668	0.2553	0.06
		A30	33.35	41.82	43.84	0.8034	0.3772	0.33
		A31	28.39	32.39	24.12	0.6840	0.2901	0.62
		A32	17.02	29.58	15.98	0.4103	0.2457	0.03
		All attacks	23.13	32.33	25.28	0.5240	0.3296	14.89
Evaluation set **with** encoding/compression		Non-target	6.00					
	TTS/VC	A17	44.07	24.66	19.18	0.8863	0.6813	48.69
		A19	48.17	62.85	59.32	0.8924	0.9491	15.58
		A21	30.87	28.11	20.46	0.7070	0.4793	42.98
		A22	26.82	26.12	20.81	0.6063	0.5466	35.27
		A24	21.15	25.96	16.41	0.4790	0.4352	38.37
		A25	10.79	32.23	24.77	0.2561	0.4592	10.97
		A26	15.97	43.22	35.17	0.3842	0.6104	25.14
		A28	39.20	41.40	35.05	0.8553	0.8738	61.64
		A29	33.07	20.13	11.20	0.7507	0.4277	55.39
	Adversarial	A18	31.05	59.26	49.08	0.6870	0.7415	11.17
		A20	30.51	46.38	34.30	0.6583	0.8652	1.62
		A23	26.07	31.81	32.63	0.5990	0.5353	16.04
		A27	28.21	39.10	29.20	0.6136	0.5965	1.38
		A30	34.66	42.29	42.32	0.7307	0.6753	4.12
		A31	30.23	33.41	28.59	0.6469	0.5685	6.64
		A32	23.87	29.73	20.53	0.5300	0.5666	1.01
		All attacks	30.18	37.11	30.61	0.6500	0.6284	29.74

Appendix D. Results with and without encoding/compressiong

See Table D.7.

Data availability

Database is publicly available. Experimental results are produced using publicly available ASVspoof 5 baseline systems.

References

AlBadawy, E.A., Lyu, S., 2020. Voice conversion using speech-to-speech neuro-style transfer. In: Proc. Interspeech. pp. 4726–4730.
Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., Weber, G., 2020. Common Voice: A massively-multilingual speech corpus. In: Proc. LREC. pp. 4218–4222.
Arik, S., Chen, J., Peng, K., Ping, W., Zhou, Y., 2018. Neural voice cloning with a few samples. In: Proc.NeurIPS, vol. 31.
Baas, M., Kamper, H., 2020. StarGAN-ZSVC: Towards zero-shot voice conversion in low-resource contexts. In: Artificial Intelligence Research. SACAIR 2021. Communications in Computer and Information Science, vol. 1342, pp. 69–84.
Baevski, A., Zhou, Y., Mohamed, A., Auli, M., 2020. Wav2vec 2.0: A framework for self-supervised learning of speech representations. In: Proc. NuerIPS, vol. 33, pp. 12449–12460.
Casanova, E., Davis, K., Gölge, E., Göknar, G., Gulea, I., Hart, L., Aljafari, A., Meyer, J., Morais, R., Olayemi, S., et al., 2024. XTTS: A massively multilingual zero-shot text-to-speech model. Proc. Interspeech 4978–4982.
Casanova, E., Weber, J., Shulby, C.D., Junior, A.C., Gölge, E., Ponti, M.A., 2022. YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone. In: Proc. ICML. pp. 2709–2720.

Chen, M., Tan, X., Li, B., Liu, Y., Qin, T., Zhao, S., Liu, T.-Y., 2021a. Adaspeech: Adaptive text to speech for custom voice. In: Proc. ICLR.
Chen, N., Zhang, Y., Zen, H., Weiss, R.J., Norouzi, M., Chan, W., 2021b. WaveGrad: Estimating gradients for waveform generation. In: Proc. ICLR.
Chettri, B., Benetos, E., Sturm, B.L.T., 2020. Dataset artefacts in anti-spoofing systems: A case study on the ASVspoof 2017 benchmark. IEEE/ ACM Trans. Audio, Speech, Lang. Process. 28, 3018–3028.
Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proc. EMNLP. pp. 1724–1734.
Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y., 2015. Attention-based models for speech recognition. In: Proc. NeurIPS. pp. 577–585.
Chung, J.S., Nagrani, A., Zisserman, A., 2018. VoxCeleb2: Deep speaker recognition. In: Proc. Interspeech. pp. 1086–1090.
Cooper, E., Huang, W.-C., Toda, T., Yamagishi, J., 2022. Generalization ability of MOS prediction networks. In: Proc. ICASSP. pp. 8442–8446.
Cooper, E., Lai, C.-I., Yasuda, Y., Fang, F., Wang, X., Chen, N., Yamagishi, J., 2020. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In: Proc. ICASSP. pp. 6184–6188.
Défossez, A., Copet, J., Synnaeve, G., Adi, Y., 2023. High fidelity neural audio compression. Trans. Mach. Learn. Res.
Défossez, A., Synnaeve, G., Adi, Y., 2020. Real time speech enhancement in the waveform domain. In: Proc. Interspeech. pp. 3291–3295.
Delgado, H., Evans, N., Jung, J.-w., Kinnunen, T., Kukanov, I., Lee, K.-A., Liu, X., Shim, H.-j., Sahidullah, M., Tak, H., Todisco, M., Wang, X., Yamagishi, J., 2024. ASVspoof 5 Evaluation plan (phase 2). v0.6, https://www.asvspoof.org/file/ASVspoof5__Evaluation_Plan_Phase2.pdf. (Accessed 23 July 2024).
Desplanques, B., Thienpondt, J., Demuynck, K., 2020. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In: Proc. Interspeech. pp. 3830–3834.
Eren, G., The Coqui TTS Team, 2021. Coqui TTS. http://dx.doi.org/10.5281/zenodo.6334862.
Evans, N., Kinnunen, T., Yamagishi, J., 2013. Spoofing and countermeasures for automatic speaker verification. In: Proc. Interspeech. pp. 925–929.
Geirhos, R., et al., 2020. Shortcut learning in deep neural networks. Nat. Mach. Intell. 2 (11), 665–673.
Gong, C., Wang, X., Cooper, E., Wells, D., Wang, L., Dang, J., Richmond, K., Yamagishi, J., 2024. ZMM-TTS: Zero-shot multilingual and multispeaker speech synthesis conditioned on self-supervised discrete speech representations. IEEE/ ACM Trans. Audio, Speech, Lang. Process. 32, 4036–4051.
Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples. In: Proc. ICLR.
Graves, A., 2008. Supervised Sequence Labelling with Recurrent Neural Networks (Ph.D. thesis). TUM.
Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., et al., 2020. Conformer: Convolution-augmented transformer for speech recognition. In: Proc. Interspeech. pp. 5036–5040.
Guo, H., Xie, F., Wu, X., Soong, F.K., Meng, H., 2023. MSMC-TTS: Multi-stage multi-codebook VQ-VAE based neural TTS. IEEE/ ACM Trans. Audio, Speech, Lang. Process. 31, 1811–1824.
Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, vol. 2, Springer.
Hayashi, T., Yamamoto, R., Inoue, K., Yoshimura, T., Watanabe, S., Toda, T., Takeda, K., Zhang, Y., Tan, X., 2020. ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In: Proc. ICASSP. pp. 7654–7658.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proc. CVPR. pp. 770–778.
Heo, H.S., Lee, B.-J., Huh, J., Chung, J.S., 2020. Clova baseline system for the VoxCeleb speaker recognition challenge 2020. arXiv preprint arXiv:2009.14153.
Hunt, A.J., Black, A.W., 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In: Proc. ICASSP, vol. 1, IEEE, pp. 373–376.
Jadoul, Y., Thompson, B., de Boer, B., 2018. Introducing Parselmouth: A Python interface to Praat. J. Phon. 71, 1–15.
Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Moreno, I.L., Wu, Y., et al., 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In: Proc. NeurIPS. pp. 4480–4490.
Jung, J.-w., Heo, H.-S., Tak, H., Shim, H.-j., Chung, J.S., Lee, B.-J., Yu, H.-J., Evans, N., 2022a. AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In: Proc. ICASSP. pp. 6367–6371.
Jung, J.-w., Kim, Y., Heo, H.-S.H., Lee, B.-J., Kwon, Y., Chung, J.S., 2022b. Pushing the limits of raw waveform speaker recognition. In: Proc. Interspeech. pp. 2228–2232.
Jung, J.-w., Kim, S.-b., Shim, H.-j., Kim, J.-h., Yu, H.-J., 2020. Improved RawNet with feature map scaling for text-independent speaker verification using raw waveforms. In: Proc. Interspeech. pp. 1496–1500.
Jung, J.-w., Tak, H., Shim, H.-j., Heo, H.-S., Lee, B.-J., Chung, S.-W., Yu, H.-J., Evans, N., Kinnunen, T., 2022c. SASV 2022: The first spoofing-aware speaker verification challenge. In: Proc. Interspeech. pp. 2893–2897.
Jung, J.-w., Wang, X., Evans, N., Watanabe, S., Shim, H.-j., Tak, H., Arora, S., Yamagishi, J., Chung, J.S., 2024. To what extent can ASV systems naturally defend against spoofing attacks? In: Proc. Interspeech. pp. 3240–3244.
Kaneko, T., Kameoka, H., Tanaka, K., Hojo, N., 2019. StarGAN-VC2: Rethinking conditional methods for StarGAN-based voice conversion. In: Proc. Interspeech. pp. 679–683.
Kearns, J., 2014. Librivox: Free public domain audiobooks. Ref. Rev. 28 (1), 7–8.
Kim, J., Kim, S., Kong, J., Yoon, S., 2020. Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. In: Proc. NeurIPS. pp. 8067–8077.
Kim, J., Kong, J., Son, J., 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In: Proc. ICML. pp. 5530–5540.
Kingma, D.P., Dhariwal, P., 2018. Glow: Generative flow with invertible 1x1 convolutions. In: Proc. NeurIPS. pp. 10236–10245.
Kingma, D.P., Welling, M., 2014. Auto-encoding variational Bayes. In: Proc. ICLR.
Kinnunen, T., Li, H., 2010. An overview of text-independent speaker recognition: From features to supervectors. Speech Commun. 52 (1), 12–40.
Kinnunen, T., Sahidullah, M., Delgado, H., Todisco, M., Evans, N., Yamagishi, J., Lee, K., 2017. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. In: Proc. Interspeech. pp. 2–6.
Kintzley, K., Jansen, A., Hermansky, H., 2011. Event selection from phone posteriorgrams using matched filters. In: Proc. Interspeech. pp. 1905–1908.
Ko, T., Peddinti, V., Povey, D., Seltzer, M.L., Khudanpur, S., 2017. A study on data augmentation of reverberant speech for robust speech recognition. In: Proc. ICASSP. pp. 5220–5224.
Kong, J., Kim, J., Bae, J., 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In: Proc. NeurIPS. pp. 17022–17033.
Łańcucki, A., 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In: Proc. ICASSP. pp. 6588–6592.
Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.-R., 2019. Unmasking Clever Hans predictors and assessing what machines really learn. Nat. Commun. 10 (1), 1096.
Lee, S.-g., Ping, W., Ginsburg, B., Catanzaro, B., Yoon, S., 2022. BigVGAN: A universal neural vocoder with large-scale training. In: Proc. ICLR.
Li, X., Zhong, J., Wu, X., Yu, J., Liu, X., Meng, H., 2020. Adversarial attacks on GMM I-Vector based speaker verification systems. In: Proc. ICASSP. IEEE, pp. 6579–6583.
Liu, X., Wang, X., Sahidullah, M., Patino, J., Delgado, H., Kinnunen, T., Todisco, M., Yamagishi, J., Evans, N., Nautsch, A., Lee, K.A., 2023. ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild. IEEE/ ACM Trans. Audio, Speech, Lang. Process. 31, 2507–2522.
Liu, S., Wu, H., Lee, H.-Y., Meng, H., 2019. Adversarial attacks on spoofing countermeasures of automatic speaker verification. In: Proc. ASRU. pp. 312–319.
Lorenzo-Trueba, J., Yamagishi, J., Toda, T., Saito, D., Villavicencio, F., Kinnunen, T., Ling, Z., 2018. The Voice Conversion Challenge 2018: Promoting development of parallel and nonparallel methods. In: Proc. Odyssey. pp. 195–202.
Lux, F., Koch, J., Meyer, S., Bott, T., Schauffler, N., Denisov, P., Schweitzer, A., Vu, N.T., 2023. The IMS toucan system for the blizzard challenge 2023. In: Proc. Blizzard Challenge Workshop. ISCA, pp. 40–45.

Lux, F., Koch, J., Schweitzer, A., Vu, N.T., 2021. The IMS toucan system for the blizzard challenge 2021. In: Proc. Blizzard Challenge Workshop. ISCA, pp. 14–19.
Lux, F., Koch, J., Vu, N.T., 2022. Exact prosody cloning in zero-shot multispeaker text-to-speech. In: Proc. SLT. pp. 962–969.
Martin, A.F., Doddington, G.R., Kamm, T., Ordowski, M., Przybocki, M.A., 1997. The DET curve in assessment of detection task performance. In: Eurospeech, vol. 4, pp. 1895–1898.
Miyato, T., Koyama, M., 2018. cGANs with projection discriminator. In: Proc. ICLR.
Mohamed, A., Lee, H.-y., Borgholt, L., Havtorn, J.D., Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., Sainath, T.N., Watanabe, S., 2022. Self-supervised speech representation learning: A review. IEEE J. Sel. Top. Signal Process. 16 (6), 1179–1210.
Müller, N., Dieckmann, F., Czempin, P., Canals, R., Böttinger, K., Williams, J., 2021. Speech is silver, silence is golden: What do ASVspoof-trained models really learn? In: Proc. ASVspoof Challenge Workshop. pp. 55–60.
Mun, S.H., Shim, H.-j., Tak, H., Wang, X., Liu, X., Sahidullah, M., Jeong, M., Han, M.H., Todisco, M., Lee, K.A., et al., 2023. Towards single integrated spoofing-aware speaker verification embeddings. In: Proc. Interspeech. pp. 3989–3993.
Nagrani, A., Chung, J.S., Zisserman, A., 2017. VoxCeleb: A Large-Scale Speaker Identification Dataset. In: Proc. Interspeech. pp. 2616–2620.
Nguyen, L.T., Pham, T., Nguyen, D.Q., 2023. XPhoneBERT: A pre-trained multilingual model for phoneme representations for text-to-speech. In: Proc. Interspeech. pp. 5506–5510.
Novak, A., Lotton, P., Simon, L., 2015. Synchronized swept-Sine: Theory, application, and implementation. J. Audio Eng. Soc. 63 (10), 786–798.
Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.
Panariello, M., Ge, W., Tak, H., Todisco, M., Evans, N., 2023. Malafide: a novel adversarial convolutive noise attack against deepfake and spoofing detection systems. In: Proc. Interspeech. pp. 2868–2872.
Panayotov, V., Chen, G., Povey, D., Khudanpur, S., 2015. Librispeech: An ASR corpus based on public domain audio books. In: Proc. ICASSP. pp. 5206–5210.
Papernot, N., McDaniel, P., Goodfellow, I., 2016. Transferability in machine learning: From phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277.
Park, D.S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E.D., Le, Q.V., 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In: Proc. Interspeech. pp. 2613–2617.
Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., Kudinov, M., 2021. Grad-TTS: A diffusion probabilistic model for text-to-speech. In: Proc. ICML. pp. 8599–8608.
Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., Kudinov, M., Wei, J., 2022. Diffusion-based voice conversion with fast maximum likelihood sampling scheme. In: Proc. ICLR.
Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., Collobert, R., 2020. MLS: A large-scale multilingual dataset for speech research. In: Proc. Interspeech. pp. 2757–2761.
Prenger, R., Valle, R., Catanzaro, B., 2019. WaveGlow: A flow-based generative network for speech synthesis. In: Proc. ICASSP. pp. 3617–3621.
Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I., 2023. Robust speech recognition via large-scale weak supervision. In: Proc. ICML. pp. 28492–28518.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8), 9.
Ravanelli, M., Bengio, Y., 2018. Speaker recognition from raw waveform with sincnet. In: Proc. SLT. pp. 1021–1028.
Ravanelli, M., Parcollet, T., Moumen, A., de Langen, S., Subakan, C., Plantinga, P., Wang, Y., Mousavi, P., Della Libera, L., Ploujnikov, A., et al., 2024. Open-source conversational AI with SpeechBrain 1.0. JMLR 25 (333), 1–11.
Ren, Y., Liu, J., Zhao, Z., 2021. PortaSpeech: Portable and high-quality generative text-to-speech. In: Proc. NeurIPS. pp. 13963–13974.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Proc. MICCAI. Springer, pp. 234–241.
Sagisaka, Y., 1988. Speech synthesis by rule using an optimal selection of non-uniform synthesis units. In: Proc. ICASSP. pp. 679–682.
Schröder, M., Charfuelan, M., Pammi, S., Steiner, I., 2011. Open source voice creation toolkit for the MARY TTS platform. In: Proc. Interspeech. pp. 3253–3256.
Shen, J., Pang, R., Weiss, R.J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R., et al., 2018. Natural TTS synthesis by conditioning wavenet on Mel spectrogram predictions. In: Proc. ICASSP. pp. 4779–4783.
Shim, H.-j., Hautamäki, R.G., Sahidullah, M., Kinnunen, T., 2023. How to construct perfect and worse-than-coin-flip spoofing countermeasures: A word of warning on shortcut learning. In: Proc. Interspeech. pp. 785–789.
Shim, H.-j., Jung, J.-w., Kinnunen, T., et al., 2024. a-DCF: an architecture agnostic metric with application to spoofing-robust speaker verification. In: Proc. Speaker Odyssey. pp. 158–164.
Shim, H.-j., Tak, H., Liu, X., Heo, H.-S., Jung, J.-w., Chung, J.S., Chung, S.-W., Yu, H.-J., Lee, B.-J., Todisco, M., et al., 2022. Baseline systems for the first spoofing-aware speaker verification challenge: Score and embedding fusion. In: Proc. Odyssey. pp. 330–337.
Snyder, D., Chen, G., Povey, D., 2015. MUSAN: A music, speech, and noise corpus. arXiv:1510.08484v1.
Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-Vectors: Robust DNN embeddings for speaker recognition. In: Proc. ICASSP. pp. 5329–5333.
Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B., 2021. Score-based generative modeling through stochastic differential equations. In: Proc. ICLR.
Steiner, I., Le Maguer, S., 2018. Creating new language and voice components for the updated MaryTTS text-to-speech synthesis platform. In: Proc. LREC. pp. 3171–3175.
Sun, L., Wang, H., Kang, S., Li, K., Meng, H., 2016. Personalized, cross-lingual TTS using phonetic posteriorgrams. In: Proc. Interspeech. pp. 322–326.
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. In: Proc. ICLR.
Tak, H., Patino, J., Todisco, M., Nautsch, A., Evans, N., Larcher, A., 2021. End-to-end anti-spoofing with RawNet2. In: Proc. ICASSP. pp. 6369–6373.
Tan, X., Qin, T., Soong, F., Liu, T.-Y., 2021. A survey on neural speech synthesis. arXiv preprint arXiv:2106.15561.
Taylor, P., 2009. Text-to-Speech Synthesis. Cambridge University Press.
Todisco, M., Panariello, M., Wang, X., Delgado, H., Lee, K.-A., Evans, N., 2024. Malacopula: Adversarial automatic speaker verification attacks using a neural-based generalised Hammerstein model. In: Proc. ASVspoof Workshop 2024. pp. 94–100.
Todisco, M., Wang, X., Vestman, V., Sahidullah, M., Delgado, H., Nautsch, A., Yamagishi, J., Evans, N., Kinnunen, T., Lee, K.A., 2019. ASVspoof 2019: future horizons in spoofed and fake audio detection. In: Proc. Interspeech. pp. 1008–1012.
Van Den Oord, A., Vinyals, O., et al., 2017. Neural discrete representation learning. In: Proc. NeurIPS. pp. 6309–6318.
Van der Maaten, L., Hinton, G., 2008. Visualizing data using T-SNE. JMLR 9 (11).
Van Leeuwen, D.A., Brümmer, N., 2007. An introduction to application-independent evaluation of speaker recognition systems. In: Speaker Classification I. Springer, pp. 330–353.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Proc. NeurIPS. pp. 5998–6008.
Wan, L., Wang, Q., Papir, A., Moreno, I.L., 2018. Generalized end-to-end loss for speaker verification. In: Proc. ICASSP. pp. 4879–4883.
Wang, X., Delgado, H., Tak, H., Jung, J.-w., Shim, H.-j., Todisco, M., Kukanov, I., Liu, X., Sahidullah, M., Kinnunen, T., Evans, N., Lee, K.A., Yamagishi, J., 2024a. ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale. In: ASVspoof Workshop 2024. pp. 1–8.
Wang, X., Kinnunen, T., Kong Aik, L., Noe, P.-G., Yamagishi, J., 2024b. Revisiting and improving scoring fusion for spoofing-aware speaker verification using compositional data analysis. In: Proc. Interspeech. pp. 1110–1114.

Wang, L., Li, J., Luo, Y., Zheng, J., Wang, L., Li, H., Xu, K., Fang, C., Shi, J., Wu, Z., 2024c. Advsv: An over-the-air adversarial attack dataset for speaker verification. In: Proc. ICASSP. IEEE, pp. 4555–4559.
Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., Dupoux, E., 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In: Proc. ACL. pp. 993–1003.
Wang, Y., Stanton, D., Zhang, Y., Ryan, R.-S., Battenberg, E., Shor, J., Xiao, Y., Jia, Y., Ren, F., Saurous, R.A., 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: Proc. ICML. pp. 5180–5189.
Wang, X., Yamagishi, J., 2023. Spoofed training data for speech spoofing countermeasure can be efficiently created using neural vocoders. In: Proc. ICASSP. pp. 1–5.
Wang, X., Yamagishi, J., Todisco, M., Delgado, H., Nautsch, A., Evans, N., Sahidullah, M., Vestman, V., Kinnunen, T., Lee, K.A., Juvela, L., Alku, P., Peng, Y.-H., Hwang, H.-T., Tsao, Y., Wang, H.-M., Maguer, S.L., Becker, M., Henderson, F., Clark, R., Zhang, Y., Wang, Q., Jia, Y., Onuma, K., Mushika, K., Kaneda, T., Jiang, Y., Liu, L.-J., Wu, Y.-C., Huang, W.-C., Toda, T., Tanaka, K., Kameoka, H., Steiner, I., Matrouf, D., Bonastre, J.-F., Govender, A., Ronanki, S., Zhang, J.-X., Ling, Z.-H., 2020. ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. Comput. Speech & Lang. 64, 101114.
Wang, H., Zheng, S., Chen, Y., Cheng, L., Chen, Q., 2023. CAM++: A fast and efficient network for speaker verification using context-aware masking. In: Proc. Interspeech. pp. 5301–5305.
Warren, K., Tucker, T., Crowder, A., Olszewski, D., Lu, A., Fedele, C., Pasternak, M., Layton, S., Butler, K., Gates, C., et al., 2024. "Better be computer or I'm dumb": A large-scale evaluation of humans as audio deepfake detectors. In: Proc. ACM CCS. pp. 2696–2710.
Waskom, M.L., 2021. Seaborn: statistical data visualization. J. Open Source Softw. 6 (60), 3021.
Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilçi, C., Sahidullah, M., Sizov, A., 2015a. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In: Proc. Interspeech. pp. 2037–2041.
Wu, Z., Swietojanski, P., Veaux, C., Renals, S., King, S., 2015b. A study of speaker adaptation for DNN-based speech synthesis. In: Proc. Interspeech. pp. 879–883.
Wu, Y., Tan, X., Li, B., He, L., Zhao, S., Song, R., Qin, T., Liu, T.-Y., 2022. AdaSpeech 4: Adaptive text to speech in zero-shot scenarios. In: Proc. Interspeech. pp. 2568–2572.
Wu, Z., Yamagishi, J., Kinnunen, T., Hanilçi, C., Sahidullah, M., Sizov, A., Evans, N., Todisco, M., Delgado, H., 2017. ASVspoof: The automatic speaker verification spoofing and countermeasures challenge. IEEE J. Sel. Top. Signal Process. 11 (4), 588–604.
Yamagishi, J., Veaux, C., MacDonald, K., 2019. CSTR VCTK Corpus: English multi-speaker corpus for CSTR Voice Cloning Toolkit (version 0.92). University of Edinburgh. The Centre for Speech Technology Research (CSTR).
Yamagishi, J., Wang, X., Todisco, M., Sahidullah, M., Patino, J., Nautsch, A., Liu, X., Lee, K.A., Kinnunen, T., Evans, N., Delgado, H., 2021. ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection. In: Proc. ASVspoof Challenge Workshop. pp. 47–54.
Yan, Y., Tan, X., Li, B., Zhang, G., Qin, T., Zhao, S., Shen, Y., Zhang, W.-Q., Liu, T.-Y., 2021. Adaspeech 3: Adaptive text to speech for spontaneous style. In: Proc. Interspeech. pp. 4668–4672.
Zhang, Y., Li, Z., Lu, J., Hua, H., Wang, W., Zhang, P., 2023. The impact of silence on speech anti-spoofing. IEEE/ ACM Trans. Audio, Speech, Lang. Process. 31, 3374–3389.
Zhang, Y., Lv, Z., Wu, H., Zhang, S., Hu, P., Wu, Z., Lee, H.-y., Meng, H., 2022. MFA-conformer: Multi-scale feature aggregation conformer for automatic speaker verification. In: Proc. Interspeech. pp. 306–310.
Zhao, Y., Huang, W.-C., Tian, X., Yamagishi, J., Das, R.K., Kinnunen, T., Ling, Z.-H., Toda, T., 2020. Voice Conversion Challenge 2020 — Intra-lingual semi-parallel and cross-lingual voice conversion —. In: Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020. pp. 80–98.
Zhu, G., Jiang, F., Duan, Z., 2021. Y-vector: Multiscale waveform encoder for speaker embedding. In: Proc. Interspeech. pp. 96–100.
Zhu, J.-Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proc. CVPR. pp. 2223–2232.
Ziyin, L., Hartwig, T., Ueda, M., 2020. Neural networks fail to learn periodic functions and how to fix it. In: Proc. NeurIPS. pp. 1583–1594.