

Retrieval-Augmented Audio Deepfake Detection

Zuheng Kang*

Ping An Technology (Shenzhen) Co.,
Ltd.
Shenzhen, China
kangzuheng896@pingan.com.cn

Yayun He*

Ping An Technology (Shenzhen) Co.,
Ltd.
Shenzhen, China
heyayun097@pingan.com.cn

Botao Zhao

Ping An Technology (Shenzhen) Co.,
Ltd.
Shenzhen, China
zhaobotao204@pingan.com.cn

Xiaoyang Qu

Ping An Technology (Shenzhen) Co.,
Ltd.
Shenzhen, China
quxiaoyang343@pingan.com.cn

Junqing Peng

Ping An Technology (Shenzhen) Co.,
Ltd.
Shenzhen, China
pengjq@pingan.com.cn

Jing Xiao

Ping An Insurance (Group) Company
of China
Shenzhen, China
xiaojing661@pingan.com.cn

Jianzong Wang[†]

Ping An Technology (Shenzhen) Co.,
Ltd.
Shenzhen, China
jzwang@188.com

ABSTRACT

With recent advances in speech synthesis including text-to-speech (TTS) and voice conversion (VC) systems enabling the generation of ultra-realistic audio deepfakes, there is growing concern about their potential misuse. However, most deepfake (DF) detection methods rely solely on the fuzzy knowledge learned by a single model, resulting in performance bottlenecks and transparency issues. Inspired by retrieval-augmented generation (RAG), we propose a retrieval-augmented detection (RAD) framework that augments test samples with similar retrieved samples for enhanced detection. We also extend the multi-fusion attentive classifier to integrate it with our proposed RAD framework. Extensive experiments show the superior performance of the proposed RAD framework over baseline methods, achieving state-of-the-art results on the ASVspoof 2021 DF set and competitive results on the 2019 and 2021 LA sets. Further sample analysis indicates that the retriever consistently retrieves samples mostly from the same speaker with acoustic characteristics highly consistent with the query audio, thereby improving detection performance.

CCS CONCEPTS

• **Information systems** → **Clustering and classification.**

*Both authors contributed equally to this research

[†]Corresponding author: Jianzong Wang, jzwang@188.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

ICMR '24, June 10–14, 2024, Phuket, Thailand

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0619-6/24/06...\$15.00

<https://doi.org/10.1145/3652583.3658086>

KEYWORDS

audio deepfake, deepfake detection, retrieval-augmented detection, retrieval-augmented generation, LLM, voice conversion, text-to-speech

ACM Reference Format:

Zuheng Kang, Yayun He, Botao Zhao, Xiaoyang Qu, Junqing Peng, Jing Xiao, and Jianzong Wang. 2024. Retrieval-Augmented Audio Deepfake Detection. In *Proceedings of the 2024 International Conference on Multimedia Retrieval (ICMR '24)*, June 10–14, 2024, Phuket, Thailand. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3652583.3658086>

1 INTRODUCTION

Recent artificial intelligence (AI) techniques have enabled the generation of synthesized audio known as DeepFakes (DF) with increasing degrees of fidelity to natural human speech. Sophisticated DF generation techniques, such as text-to-speech (TTS) and voice conversion (VC) to mimic the timbre, prosody, and intonation of the speaker, can now generate audio perceptually indistinguishable from genuine recordings. However, the potential malicious use of such AI-synthesized speech has serious personal and societal implications, including disruption of automatic speaker verification (ASV) systems, propagating misinformation, and defaming reputations. As a result, the vulnerability of current audio-based communication systems to synthesized speech poses a serious threat. Therefore, the development of effective DF detection techniques is urgently needed to address this emerging risk for everyone of us.

Recent advances in artificial intelligence generation content (AIGC) techniques, such as DF generation, however, have made detecting DFs increasingly challenging. In newly organized competitions, such as the ASVspoof 2021 [27], ADD 2022 [28], 2023 [29], even the state-of-the-art (SOTA) DF detection systems perform poorly, with an equal error rate (EER) of over 10% [30], making them impossible to detect properly, and unsuitable for commercial deployment. This suggests that the rapid development of DF generation seems to have greatly outpaced the development of detection

technologies – the traditional detection methods are far from inadequate for identifying current out-of-domain [3] DF-generated samples. Building robust and reliable DF detection systems remains a critical issue for the research community that has not yet been fully resolved.

Recent decades have witnessed the emergence of various frameworks for detecting DF audio. The predominant frameworks utilize a pipeline consisting of a front-end feature extractor and a back-end classifier (discussed in § 2.1). Early works relied on hand-crafted features for DF detection with some success, such as Mel-frequency cepstral coefficients (MFCC) [10], linear-frequency cepstral coefficients (LFCC) [5, 17, 24], CQT [16, 32], and F0 sub-band feature [7]. However, these features exhibit limited performance due to the limited dataset used to train the robust model. More recent SOTA frameworks have leveraged the capabilities of self-supervised model feature extractors, such as wav2vec [1, 2, 18, 21, 25] and WavLM [2, 4, 9]. Moreover, Kawa et al. [13] utilize another powerful task-specific feature in deepfake detection using the pre-trained Whisper model [19]. Theoretically, by training on a large number of labeled or unlabeled bonafide samples in very large datasets, these feature extractor models could be particularly sensitive to unseen DF artifacts. Some alternative frameworks replace hand-crafted features with end-to-end trainable encoders. Notable examples include Jung et al.'s advanced graph attention network architecture AASIST [26], and Huang et al.'s discriminative frequency-based improvements to SincNet [12], both of which achieve competitive performance. As discussed by Sun et al. [20], vocoders employed in regular speech synthesis can introduce inconsistencies that reveal DFs. However, regardless of the methodology and architecture used, these frameworks rely solely on a single model to accomplish this challenging task, which may prove inadequate.

DF detection should be a knowledge-intensive task that also relies on vast external knowledge. To explain this, let's start with a story. In the identification of antique artifacts, fakes are often fabricated so realistically that it is difficult to determine their authenticity. Senior experts usually conduct a meticulous comparative analysis that includes material and textural attributes of many similar artifacts. With so many subtle factors to consider, relying solely on one's limited intelligence is likely to result in poor judgment. Similarly, relying only on a single model to detect deepfakes may be too challenging to make mistakes.

Retrieval augmented generation (RAG) [8, 15] methodology provides a nice example for solving the problem of knowledge-intensive tasks (discussed in § 2.2). The RAG resolved the limitation of the single model by combining a pre-trained large language model (LLM) with an information retrieval system over a large knowledge database. Fundamentally, the RAG framework leverages significant supplementary real-time updated additional interpretable knowledge (with rapidly changing, proprietary data sources) to augment the limitations of a single model's knowledge, enabling it to provide a more reasoned answer. Complementary background information allows a single model to overcome its inherent knowledge gaps.

Similarly, the retrieval-augmented-based approach could provide the same benefits for DF detection. When analyzing suspect audio, the model could query a retrieval system to find many similar reference audio segments. These retrieved results would provide additional references to inform the deepfake detection, such as

typical artifacts for bonafide or synthetic spoofed examples. Then, the deepfake detector could integrate these retrieval results and the suspected audio into its decision process. This retrieval-augmented approach has several advantages. The model gains access to a much larger knowledge base beyond what can be encoded only in its model parameters. The retrieved results also provide supporting evidence for decisions, improving model performance. Furthermore, the system can update or modify its knowledge database for different detection tasks, since the model has learned to characterize a limited type of DF synthesizing methods. In summary, augmenting deepfake detectors with conditional retrieval of external data is an attractive direction. Developing DF detection methodology with a retrieval-augmented approach is worthy of further exploration. To implement the above-mentioned issue, we made the following contributions:

- We proposed a retrieval augmented detection (RAD) framework which innovatively retrieves similar samples with additional knowledge, and incorporates them into the detection process for improved detection performance.
- We extend the multi-fusion attentive classifier to integrate with RAD.
- Extensive Experiments show that our proposed method achieves state-of-the-art results on the ASVspoof 2021 DF set and competitive results on the 2019 and 2021 LA sets, demonstrating the effectiveness of the proposed retrieval-augmented approaches.

2 PRELIMINARY

2.1 Traditional Frameworks

(1) *Pipeline Framework.* The most common pipelines typically include separate components for feature extraction, and classification (Figure 1-1). Specifically, the front-end feature extractor first converts the raw speech signal x into a speech features y . These speech features are then passed to the back-end classifier, which analyzes the speech features that make a bonafide vs. spoof decision z . Such architectures leverage the capabilities of efficient hand-crafted features or semantically rich self-supervised pre-trained features to obtain highly informative speech representations. The back-end classifier then fully analyzes and mines these features, and makes the final prediction.

(2) *End-to-End Framework.* More advanced frameworks transform the feature extraction into a trainable encoder, forming an end-to-end architecture (Figure 1-2). Differently, instead of the feature extractor, the raw speech x is fed into a trainable encoder to produce speech representations y . To further improve performance, some approaches fuse multiple feature extractors or encoders, concatenating the resulting speech features and representations for deeper training.

2.2 Retrieval Augmented Generation

Prior to introducing our proposed approach, it is instructive to first provide background knowledge on the retrieval augmented generation (RAG) framework [15] to facilitate comparison with our method. The RAG framework consists of three main stages:

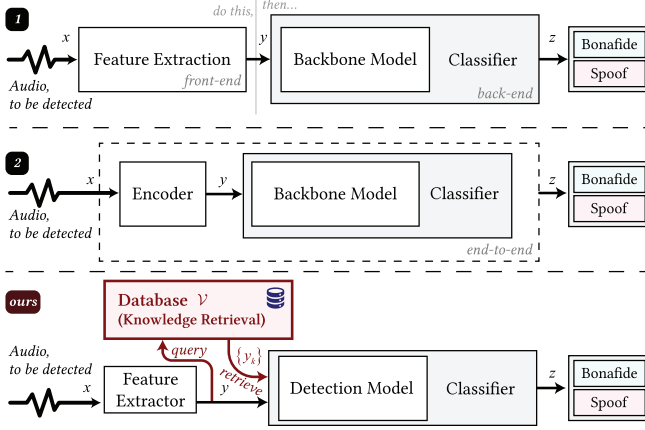


Figure 1: The overview of traditional frameworks, and our proposed framework for audio deepfake detection. (1) shows the pipeline framework. (2) shows the end-to-end framework. (ours) shows our proposed retrieval augmented-based detection (RAD) framework.

(1) *Build Knowledge Retrieval Database.* As shown in the stage 1 (blue section) of Figure 3-RAG, the plain text and format-rich text database x is partitioned into smaller chunks $\{x_n\}$ at n^{th} sample. These text chunks are then embedded into dense vector representations $\{v_n\}$ by a language model. In addition, each embedding v_n maintains an index that links it to its original text chunk x_n , allowing the retrieval of the original text content. Finally, these embeddings $\{v_n\}$ can be stored in a vector database \mathcal{V} that facilitates efficient similarity search and retrieval.

(2) *Retrieve Knowledge.* As shown in the stage 2 (red section) of Figure 3-RAG, the user’s query text \tilde{x}_q is embedded into a query embedding \tilde{v}_q using the same language model. This query embedding is then leveraged to perform a similarity search across the vector database \mathcal{V} containing all the document chunk embeddings. The top K most similar embeddings $\{\tilde{v}_k\}$ related to its document chunks $\{\tilde{x}_k\}$ are retrieved based on their semantic proximity to the query embedding \tilde{v}_q in the vector space. These most relevant K document chunks $\{\tilde{x}_k\}$ can be used as augmented contextual information to complement the original user query.

(3) *Get Results (Answer Generation).* As shown in stage 3 (green section) of Figure 3-RAG, the original user query \tilde{x}_q is concatenated with the retrieved document chunks $\{\tilde{x}_k\}$ to construct an expanded prompt p with its function \mathcal{P} . Where $p = \mathcal{P}(\tilde{x}_q, \{\tilde{x}_k\})$. This enriched prompt p , which contains both the initial query and relevant contextual information, is subsequently input to a large language model (LLM). The LLM analyzes the overall content and relationships within p to generate the final answer z .

3 METHODOLOGY

3.1 Self-supervised Feature with WavLM

The overall framework of our proposed method is illustrated in Figure 1. Unlike traditional methods (described in § 2.1), our proposed

framework leverages the state-of-the-art WavLM [4] feature extractor and incorporates an additional retrieval module after feature extraction to overcome performance bottlenecks. Specifically, we adopt a retrieval-augmented structure similar to RAG (described in § 2.2), which retrieves a few similar features from the bonafide samples and fuse them with the original test features before feeding into the detection model. By incorporating retrieved features highly similar to the test sample, our model can make much more reliable predictions through joint analysis.

In the following section, we describe these modules in detail, including the WavLM feature extractor (described in § 3.1), the retrieval augmented mechanism (described in § 3.2), and the design of the detection model classifier (described in § 3.3). Meanwhile, since this framework is complex, we present a number of speed-up techniques (described in § 3.4) to greatly reduce the space and time complexity. To further improve performance, we also jointly optimize the WavLM feature extractor with the detection model in an end-to-end manner (described in § 3.1).

WavLM Feature Extraction. Recent advanced feature extractor WavLM [4] employs wav2vec 2.0 [1] as its backbone and is trained with larger real multilingual, multi-channel unlabeled speech data for much better performance. WavLM utilizes a masked speech denoising and prediction framework that artificially adds noise and overlapping speech to clean input audio before masking certain time segments, and the model should then predict the speech content of the original frames of these masked segments. This denoising process allows WavLM to learn robust representations that capture not only a variety of speech features but also the acoustic environment. In addition, WavLM performs excellently in a variety of downstream speech tasks such as automatic speech recognition (ASR), automatic speaker verification (ASV), and text-to-speech (TTS) with minimal fine-tuning. This suggests that WavLM already understands and is familiar with many high-level speech characteristics of bonafide audio, which are particularly appropriate for unseen DF-synthesized audio, since it often contains features that are very different from bonafide audio.

In the proposed framework, the feature extraction component utilizes the complete set of latent features $y \in \mathbb{R}^{L \times T \times F}$ from all layers of the WavLM encoder transformer when processing an input audio segment x . Where L is the number of WavLM encoder transformer layers, T is the number of frames, F is the dimension of features (same as WavLM feature size). This enables the model to leverage speech information encompassing low-level acoustic features as well as higher-level semantic abstractions extracted by the deeper layers.

WavLM Fine-Tuning. Since WavLM is trained only on bonafide audio during pre-training, it may not have exposure to spoofed samples, which potentially leads to incorrect classifications. Therefore, we first fine-tune the entire WavLM feature extractor in an

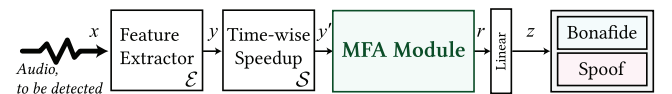


Figure 2: The baseline structure for fine-tuning.

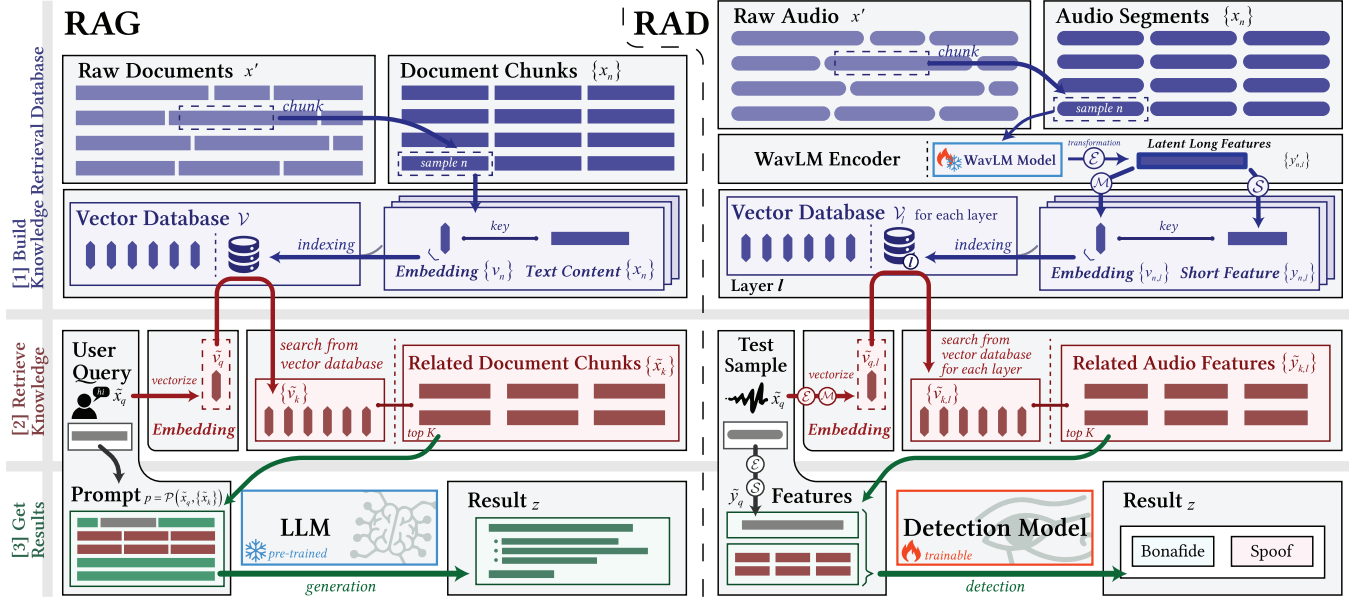


Figure 3: The overview of the RAG and RAD pipeline. Triangular edge rectangles represent vectors for retrieval databases. In RAG, long rectangles represent document chunks. In RAD, long rectangles with/without an outline represent long/short features, rounded edge rectangles represent audio segments.

end-to-end manner without the RAD framework. That is, shown in Figure 2, the speech is encoded into short features by a trainable WavLM model \mathcal{E} and time-wise speedup method \mathcal{S} , which is then encoded into intermediate representations by an MFA module. These representations are then classified as either bonafide or spoofed by a fully connected layer. By jointly optimizing the parameters, we obtain a fine-tuned WavLM model that can serve as an improved feature extractor in the subsequent RAD framework. In the subsequent RAD inference phase, we only leverage the fine-tuned WavLM and discard the back-end model.

3.2 Retrieval Augmented Detection

To address the performance limitations imposed by the detection bottleneck, we propose the retrieval augmented detection (RAD) framework. Similar to the RAG framework, the proposed RAD approach consists of three main stages, but with some procedural modifications compared to RAG.

(1) *Build Knowledge Retrieval Database.* As shown in the stage 1 (blue section) of Figure 3-RAD, the bonafide audio dataset x' is segmented into smaller audio segments $\{x_n\}$ at the n^{th} sample. These audio segments can be encoded to latent long feature representations $\{y'_{n,l}\} \in \mathbb{R}^{N \times L \times T' \times F}$ by WavLM feature extractor $\mathcal{E}(\cdot)$, where T' is the time dimension of long features, N is the number of audio segments, and l indexes the encoder layer of WavLM. Subsequently, two operations are performed on the long features $\{y'_{n,l}\}$: (a) The features are embedded into dense vector representations $\{v_{n,l}\} \in \mathbb{R}^{N \times L \times F}$ via the mapping $\mathcal{M}(\cdot)$ in Equation 1, where each $v_{n,l}$ summarizes the time-wise features $y'_{n,l}$ by temporal averaging to eliminate the time dimension T' ; (b) The time dimension is

shortened to form short feature $\{y_{n,l}\} \in \mathbb{R}^{N \times L \times T \times F}$ for improved efficiency by the function $\mathcal{S}(\cdot)$ (details in § 3.4), where T is the time dimension of short feature.

$$v_{n,l} = \mathcal{M}(y'_{n,l}) = \frac{1}{T'} \sum_{t=1}^{T'} y'_{n,l}|_t. \quad (1)$$

Importantly, each embedding $v_{n,l}$ maintains an index linking to its original short feature $y_{n,l}$, enabling the retrieval of the original audio segment x_n for the source content. Finally, the collection of embeddings $\{v_{n,l}\}$ are stored in l vector databases \mathcal{V}_l to enable efficient similarity search and retrieval.

(2) *Retrieve Knowledge.* As shown in the stage 2 (red section) of Figure 3-RAD, a sample to be detected \tilde{x}_q can be embedded into a query embedding $\tilde{v}_{q,l} \in \mathbb{R}^{L \times F}$ by function \mathcal{E} , \mathcal{M} , and can be converted to short features $\tilde{y}_{q,l} \in \mathbb{R}^{L \times T \times F}$ by function \mathcal{E} , \mathcal{S} . This query embedding $\tilde{v}_{q,l}$ is then utilized to perform a similarity search across vector databases – for each layer l , there is a corresponding vector database \mathcal{V}_l to be searched. The top K most similar embeddings $\{\tilde{v}_{k,l}\} \in \mathbb{R}^{K \times L \times F}$ are retrieved, along with their associated short features $\{\tilde{y}_{k,l}\} \in \mathbb{R}^{K \times L \times T \times F}$. These K most relevant audio samples serve as references for detailed comparison with the sample to be detected. By analyzing the similarities and differences, the authenticity of the tested samples can be better determined.

(3) *Get Results (Sample Detection).* As shown in stage 3 (green section) of Figure 3-RAD, our proposed RAD framework requires the training of an additional detection model. This model accepts the samples to be tested as well as the most relevant retrieved samples. Specifically, the detection model is provided with the query short features $\tilde{y}_{q,l}$ and the top K similar short features $\{\tilde{y}_{k,l}\}$ to make the

Comparison	RAG	RAD	Full training / fine tuning for detection
Knowledge Updates	Updates directly to the latest retrieved knowledge base, without the need for frequent re-training to obtain the latest knowledge.	Knowledge is input implicitly. The ability of previously trained models for knowledge acquisition is yet to be verified.	Full training or retraining is needed for knowledge and data updates.
External Knowledge	Utilizing external resources, especially relevant or similar data from databases.	External knowledge can be acquired through retraining.	Only large datasets of high quality can improve performance.
Data Processing	Minimal data processing and manipulation is required.	Only similar samples can be traced back to specific data sources, offering greater interpretability and traceability.	Works as a black box with very low interpretability.
Interpretability	The results generated can be traced back to specific data sources, offering greater interpretability and traceability.	Only similar samples can be traced back to specific data sources, offering greater interpretability and traceability.	Works as a black box with very low interpretability.
Computational Resources	Database retrieval techniques are required for each generated or detection task, and external data sources require regular maintenance. These require additional computational resources.	Initially requires full training for specific tasks with high quality data.	Relying only on one model's judgment.
External Training	Zero-shot learning, without external training.	Initially requires full training for specific tasks with high quality data.	Only the model inference process will have low latency.
Latency Requirements	Data retrieval process leads to higher latency.	Careful comparison and review of retrieved similar samples with the tested samples may reduce detection errors, but this needs to be verified.	Relying on only one model may produce higher detection errors.
Hallucination/Detection Error	Less likely to hallucinate because each generated result is based on retrieved evidence.	The labels are compared, credible, and free of ethical and privacy concerns.	
Ethical and Privacy Issues	Content in external databases may have ethical and privacy concerns.		

Figure 4: Properties of RAG, RAD, and full training/fine-tuning for detection. Red text represents the focused attention, and green cells represent ideas that should be verified in this paper.

final decision z . Importantly, this detection model not only evaluates relevant samples, but also provides detailed comparisons with the most similar real samples. This additional contextual information helps to make more accurate judgments for DF detections.

Properties similar to RAG in RAD. Despite their different applications, with RAD optimized for detection and RAG for generation, given the similarities in structure and algorithms between RAG and RAD, it is likely that RAD also has the same advantages as RAG. Figure 4 provides a detailed summary of the key advantages and disadvantages of RAG, RAD, and full training / fine-tuning approaches. Although similarities and differences exist across these methods, three critical questions emerge as follows:

- **Question 1:** Does the RAD framework reduce detection errors?
- **Question 2:** Does updating external knowledge for the RAD framework further improve detection performance?
- **Question 3:** Can the retrieved audio samples be interpreted?

Research questions 1, 2 are verified in § 4.4, and question 3 is verified in § 4.5.

3.3 Detection Model

To apply RAD to DF detection, we extend the Multi-Fusion Attentive (MFA) classifier [9], named RAD-MFA, which combines the raw query input for detection and the retrieved similar bonafide samples to make comprehensive analysis for detections. Specifically, Figure 5 illustrates the overall structure of our proposed detection model, and Figure 5 shows the MFA sub-modules in detail.

MFA Module. The MFA Module in our framework handles the test feature \tilde{y}_q and the retrieved features $\{\tilde{y}_{k,l}\}$. For conciseness, these features are denoted by y in Figure 5. Specifically, the MFA module is implemented through the following steps:

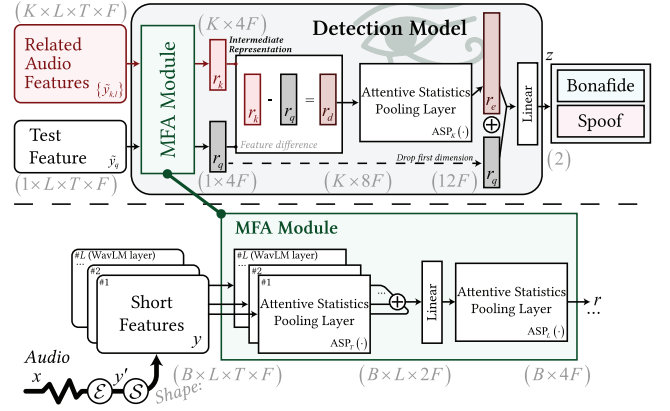


Figure 5: The structure of detection model architecture. \oplus denotes the concatenation. This process illustrates the 3rd get results stage of Figure 3-RAD in detail.

- (1) The input feature $y \in \mathbb{R}^{B \times L \times T \times F}$ is passed through L parallel time-wise attentive statistic pooling (ASP) layers (denoted as $ASP_T(\cdot)$) to eliminate the time dimension. Here, B denotes a virtual dimension.
- (2) The last outputs are concatenated and passed through a fully connected layer to transform the features to $\mathbb{R}^{B \times L \times 2F}$.
- (3) These outputs are then passed through a layer-wise ASP layer (denoted as $ASP_L(\cdot)$) to form the intermediate representation $r \in \mathbb{R}^{B \times 4F}$.

Extended RAD-based MFA. The RAD-MFA is implemented through the following steps:

- (1) The test feature $\tilde{y}_q \in \mathbb{R}^{1 \times L \times T \times F}$ and the retrieved features $\{\tilde{y}_{k,l}\} \in \mathbb{R}^{K \times L \times T \times F}$ are sent to the same MFA module, creating intermediate representations $r_q \in \mathbb{R}^{1 \times 4F}$ and $r_k \in \mathbb{R}^{1 \times 4F}$ respectively.
- (2) These two representations are used to form $r_d \in \mathbb{R}^{1 \times 4F}$ by taking their difference, $r_d = r_k - r_q$. We make a difference between two features with extremely similar timbre, which allows the discriminative model to pay more attention to other differential information, such as background noise.
- (3) This output is sent to a sample-wise ASP layer (denoted as $ASP_K(\cdot)$) to form the intermediate representation $r_e \in \mathbb{R}^{1 \times 8F}$.
- (4) r_e is concatenated with r_q , and sent to a fully connected layer to make the final decision.

Through this scheme, the RAD-based detection model can take into account numerous particularly similar bonafide samples and make comprehensive judgments on their contents and distributions. This enables the model to achieve more accurate detection results by accounting for many additional highly similar authentic cases.

3.4 Performance Optimization

In order to speed up the process of training and testing, two approaches are used for optimization.

Locally Stored Features. To speed up the training and testing process, we pre-compute and cache the WavLM features of all audio

segments in the knowledge retrieval database. Specifically, each raw audio segment x_n is passed through the WavLM model to extract the corresponding feature representation $y_{n,l}$. The retrieved feature $\{y_{n,l}\}$ are stored locally, and the indexes in the database only store pointers to these pre-extracted feature paths. For training, the entire pipeline operates on the pre-computed features rather than raw audio, with the cached test features, retrieved database features, and corresponding labels, making the training process extremely fast. For testing, this design allows efficient retrieval of audio contents without repetitively invoking the computationally expensive WavLM feature extraction each time.

Time-wise Speedup. The acoustic features extracted by WavLM have a frame-to-frame hop size of 20 ms and a large number of layers, resulting in an extremely large number of generated feature parameters that require a large amount of storage space. To solve the problem, we propose a method $\mathcal{S}(\cdot)$ to simplify features in the time dimension.

Let x denote an audio segment sample, the feature extracted by WavLM is called latent long features $y' \in \mathbb{R}^{T' \times F}$, which can then be transformed into short features $y \in \mathbb{R}^{T \times F}$ by the \mathcal{S} . A speedup parameter τ is introduced, which partitions the long feature y' along the time dimension as:

$$\text{partition}(y') = \underbrace{[[y'_1, \dots, y'_\tau], [y'_{\tau+1}, \dots, y'_{2\tau}], \dots, [\dots, y'_{T'}]]}_{T'/\tau \text{ partitions}}. \quad (2)$$

The speedup short feature y_t can then be derived by taking the average along the partitioned time dimension as:

$$y = \mathcal{S}(y') = [\text{mean}([\dots, y'_\tau]), \dots, \text{mean}([\dots, y'_{T'}])]. \quad (3)$$

Applying this speedup technique enables significant savings in storage space. However, there is an additional question that needs to be experimentally verified:

- **Question 4:** Does time-wise speedup method affect downstream DF detection performance?

This question is validated in § 4.4.

4 EXPERIMENTS

The following section describes the datasets and assessment metrics (described in § 4.1) used for all of the reported experimental work, as well as details of the reproducible implementation (described in § 4.2). The experimental results (described in § 4.3) will list the evaluation results compared to the existing SOTA. The ablation studies (described in § 4.4) will then focus on several experiments related to the **Four Research Questions** collected from § 3.2, 3.4, and why our proposed RAD framework could be effective.

4.1 Datasets and Metrics

ASVspoof 2019 LA Database. The ASVspoof 2019 [23] logical access (LA) dataset is comprised of bonafide and spoofed utterances generated using totally 19 different spoofing algorithms, including TTS, VC, and replay attacks. The dataset contains separate partitions for training, development, and evaluation. The training and development sets contain samples of 6 spoofing algorithms, while the evaluation set contains samples from 2 algorithms seen during training as well as 11 unseen spoofing algorithms not present in the

training data. The training set trains the model, the development set selects the best-performing model, and the evaluation set impartially evaluates the performance of the selected model. In addition, all bonafide samples will be used to build the retrieval database.

This experimental design aims to evaluate the generalization ability of the DF detection system against unknown spoofing attacks. Furthermore, the dataset may not contain complete bonafide recordings of all speakers that were impersonated in the spoofing dataset. The lack of target speaker data may limit the ability of the DF detection system to perform accurate speaker characteristics comparisons. To mitigate this issue, additional bonafide samples from impersonated speakers should be found to augment the available knowledge database for the retrieval system.

ASVspoof 2021 LA Database. The LA and DF evaluation subsets from the ASVspoof 2021 [27] challenge present intentionally more difficult spoofing detection tasks compared to the 2019 LA data, including more unseen attacks, and both encoding and transmission distortions in the LA set, as well as the unseen coding and compression artifacts in the DF set. According to the challenge guidelines, since no new training or development data was released for the ASVspoof 2021 challenge, model training, and evaluation were limited only to the ASVspoof 2019 database. The entire subset of ASVspoof 2021 LA and DF were used for model testing.

VCTK Database. All bonafide samples of ASVspoof are a very small subset of the VCTK dataset. To mitigate the issue of insufficient bonafide samples per speaker, as the datasets used originate from the VCTK dataset, we expand our database by using additional samples that exclude all samples already present in our experimental data. This allows us to increase the number of bonafide samples available for each speaker in the retrieval task.

Metrics. We evaluated our proposed model on these datasets using two standard metrics: the minimum normalized tandem detection cost function (min t-DCF) [14] and pooled equal error rate (EER). The min t-DCF measures the combined (tandem) performance of the ASV systems, and the EER reflects the independent DF detection capability.

4.2 Implementation Details

Data Processing. The original audio recordings in the database are segmented into clips of 4 seconds in length. Audio recordings over the 4-second duration are truncated to 4 seconds. For audio recordings shorter than 4 seconds, the clips are padded to the 4-second length by repeating the recording. No additional processing such as voice activity detection (VAD) is applied to the audio samples prior to segmentation. The audio segments are first encoded into short features using a WavLM model, and are then shortened into more compact short features through a time-wise speedup model with parameter $\tau = 10$.

Vector Database. In order to store and query embeddings from vector databases more conveniently, we created L databases to store the audio feature vectors extracted at each layer of the WavLM model. When performing a database retrieval, the query audio is converted to query embedding, and the top 10 most similar WavLM short features will be retrieved.

Table 1: Comparison with other anti-spoofing systems in the ASVspoof 2019 LA evaluation set, reported in terms of pooled min t-DCF and EER (%).

System	Configuration	min t-DCF	EER(%)
Hua et al. [11]	DNN+ResNet	0.0481	1.64
Zhang et al. [31]	FFT+SENet	0.0368	1.14
Ding et al. [6]	SAMO	0.0356	1.08
Tak et al. [22]	RawGAT-ST	0.0335	1.06
Jung et al. [26]	AASIST	0.0275	0.83
Huang et al. [12]	DFSincNet	0.0176	0.52
Fan et al. [7]	f0+Res2Net	0.0159	0.47
Guo et al. [9]	WavLM+MFA	0.0126	0.42
Ours	WavLM+RAD-MFA	0.0115	0.40

Model Training. The front-end feature extractor utilized in this work is WavLM. During fine-tuning of the front-end WavLM, the Adam optimizer is employed with a learning rate of $3e-6$ and a batch size of 4. For training the MFA, the batch size is changed to 32 and the learning rate is $3e-5$. All experiments were performed utilizing two NVIDIA GeForce RTX 3090 GPUs. Each model configuration is trained for approximately 30 epochs.

4.3 Experimental Results

To demonstrate the superior performance of our proposed method over existing approaches, we compare our proposed method to recent SOTA methods.

Results on ASVspoof 2019 LA evaluation set. The experimental results in Table 1 compare the performance of our proposed methods to existing approaches on the ASVspoof 2019 LA evaluation dataset. Our method achieves an EER of 0.40% and a min t-DCF of 0.0115 which is the best reporting result, demonstrating the effectiveness and superiority of our proposed method. Notably, although Guo et al. [9] utilizes a similar WavLM feature extractor and MFA network, our proposed RAD framework improves its performance, overcoming the limitations of single-model approaches.

In our analysis, the RAD framework first retrieves the most similar audio samples, which are likely from the same speaker, and then performs careful comparisons between these samples and the test sample. For the detection model, it only needs to consider the differences between the two, rather than relying on fuzzy prior knowledge for detection. In contrast, our proposed method is more robust. Specifically, by focusing on fine-grained differences rather than generalized knowledge, our method can more accurately distinguish more detailed information.

Results on ASVspoof 2021 DF evaluation set. We further test our model on the ASVspoof 2021 LA and DF evaluation set, results are shown in Table 2. In the DF subset, our method achieves SOTA performance on the DF subset with an EER of 2.38%. In the LA subset, we obtain an EER of 4.89%, which is also quite a competitive performance, but still better than the baseline system [9] without RAD. Further analysis and ablation studies are needed to a fully

Table 2: Comparative results of our proposed method with other systems in the ASVspoof 2021 LA and DF evaluation set with pooled EER (%).

System	Configuration	LA	DF
Fan et al. [7]	f0+Res2Net	3.61	–
Doñas et al. [18]	wav2vec2+ASP	3.54	4.98
Wang et al. [25]	wav2vec2+LGF	6.53	4.75
Tak et al. [21]	wav2vec2+AASIST	0.82	2.85
Fan et al. [7]	WavLM+MFA	5.08	2.56
Ours	WavLM+RAD-MFA	4.83	2.38

Table 3: Ablation studies on ASVspoof 2021 DF dataset for the effectiveness of each component with pooled EER (%). -L and -S: large and small. ft: fine-tuning. Just Difference is the r_e (denoted in Figure 5, without r_q) directly connected to the fully connected layer for classification.

Ablation	Configuration	Pooled EER(%)
Full Framework	–	2.38
w/o RAD	Baseline (Figure 2)	2.90
w/o VCTK	ASVspoof 2019 only	2.54
w/o WavLM-L	WavLM-S	9.15
w/o ft	WavLM-S	9.62
	WavLM-L	4.98
Variation Structure	Just Difference	2.49

characterize the advantages of our proposed method on each component.

4.4 Ablation Study

Ablation Study on Different Components. The ablation study presented in Table 3 summarizes the results obtained by evaluating different configurations and components of the proposed system on the ASVspoof 2021 DF subset. Specifically, the impact of the RAD framework, WavLM-L feature extractor, fine-tuning of the feature extractor, incorporation of additional VCTK datasets for data retrieval, and the structure of the detection network were analyzed. The experiments were conducted using a time-wise speedup parameter $\tau = 10$. System performance was assessed using the pooled EER expressed as a percentage. The key observations are summarized as follows (line-by-line explanation from Table 3):

- (1) The full system reaches the SOTA performance with a pooled EER of 2.38%.
- (2) Removing the proposed RAD framework for similar sample retrieval increases the pooled EER to 2.90%. This validates the effectiveness of the RAD framework, which answers the research **Question 1**. However, this result is slightly higher than that of Guo et al. [9], which may be due to different parameter settings and time-wise speedup operations.

Table 4: Ablation study of the effect of different time-wise speedup parameter τ on DF detection performance of ASVspoof 2021 DF dataset, using pooled EER (%).

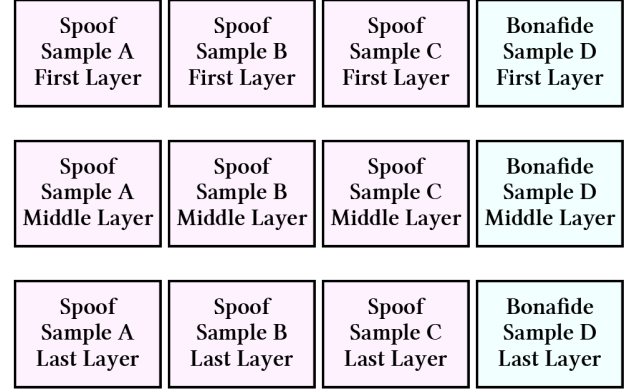
Speedup ($\tau =$)	5	10	20
Original	4.68	4.98	5.45
Fine-tune	2.36	2.38	2.54

- (3) Excluding the supplementary VCTK dataset slightly increases the pooled EER to 2.64%, indicating that updating the knowledge with additional related data could improve the detection performance, which answers the research *Question 2*.
- (4) Replacing WavLM-L with WavLM-S significantly increases the pooled EER to 9.15%, highlighting the importance of the feature extractor in the overall framework.
- (5) Without fine-tuning, the EER rises drastically to 9.62% and 4.98% for WavLM-S and WavLM-L respectively. This observation clearly highlights the positive influence of fine-tuning in enhancing DF detection performance, since fine-tuning combines spoofed data instead of using bonafide data alone, thereby improving the discriminatory capability of DF samples.
- (6) We also tried the variation structure of removing the r_q branch and directly connecting r_e (denoted in Figure 5) to the classifier slightly increases the pooled EER to 2.49%, suggesting that not only the difference of the feature, but also the original feature play the role for performance improvement.

Effect of Time-wise Speedup Parameter. Table 4 examines whether time-wise speedup affects the performance of DF detection. We tested the original and the fine-tuned WavLM-Large feature extractor on the ASVspoof 2021 DF with τ is 5, 10, 20, reporting by the pooled EER. It is difficult to test under $\tau < 5$ due to very high computational costs and storage consumption, which need to be addressed in future work. Before fine-tuning, the performance varies greatly across τ : the smaller τ is, the better the performance, but the much higher computational cost and storage consumption. After fine-tuning, the gap narrows, suggesting that optimization can reduce the impact of time-wise speedup operation. Overall, the time-wise speedup operation will affect the performance, but not too much, which answers the research *Question 4*. Taking these factors into account, we finally chose $\tau = 10$. However, we still need better ways to reduce storage and computation, which is an open problem that needs to be investigated.

4.5 Sample Analysis

The retrieval samples shown in Figure 6 with clickable audio to hear, offer insights into the factors influencing successful detection of spoofing artifacts. This figure presents 4 test samples, comprising 3 spoofed and 1 bonafide audio, along with the 2 to 3 most similar samples. Extracts were taken from three layers (initial, middle, final) of a WavLM-L model. Despite quality defects in the spoofed test samples, the system appeared to retrieve samples from the same speaker identities. This suggests the system might rely strongly on speaker-discriminative features, potentially providing an approximate answer to the research *Question 3*. However, although

**Figure 6: Examples of retrieved samples (click on the figure to hear the sound).**

initial layer retrievals corresponded to the same speakers, middle and final layer results may differ in speaker identity. Our analysis implies that the shallower layers of the model may focus on timbral and quality-based features, whereas the deeper layers capture more abstract semantic information. Nevertheless, these explanations remain brief and qualitative, lacking rigorous argumentation, which could be an interesting area for future exploration. In summary, the retrieved samples have the greatest similarity at the feature level, providing insights for the successful detection of the spoofing artifacts. After retrieval, a careful comparison of the retrieval results with the test sample will be a key factor in the performance improvement.

5 CONCLUSIONS

In this work, we proposed a novel retrieval-augmented detection (RAD) framework that leverages retrieved samples to enhance deepfake detection performance. We also extend the multi-fusion attentive classifier by integrating it with our proposed RAD framework. Extensive experiments demonstrate state-of-the-art results of the RAD framework on the ASVspoof 2021 DF dataset and competitive performance on the 2019 and 2021 LA datasets. The consistent improvements achieved on multiple datasets highlight the potential of RAD as a new paradigm for DF detection. The ablation study reveals RAD and retrieval-augmented generation (RAG) share similar properties in improving detection performance. Additionally, the retrieved samples are usually from the same speaker, suggesting potential interpretability. In conclusion, this work opens promising research avenues into retrieval-based augmentation techniques to enhance performance for detection tasks. By breaking the reliance on a single model, RAD provides a new perspective that utilizes more available information to overcome performance limitations and advance DF detection techniques.

ACKNOWLEDGMENTS

Supported by the Key Research and Development Program of Guangdong Province (grant No. 2021B0101400003) and the Corresponding author is Jianzong Wang (jzwang@188.com).

REFERENCES

- [1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances In Neural Information Processing Systems* 33 (2020), 12449–12460.
- [2] Zexin Cai and Ming Li. 2024. Integrating frame-level boundary detection and deepfake detection for locating manipulated regions in partially spoofed audio forgery attacks. *Computer Speech & Language* 85 (2024), 101597.
- [3] Jinggang Chen, Junjie Li, Xiaoyang Qu, Jianzong Wang, Jiguang Wan, and Jing Xiao. 2024. GAIA: Delving into Gradient-based Attribution Abnormality for Out-of-distribution Detection. *Advances in Neural Information Processing Systems (NIPS)* 36 (2024).
- [4] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Micheal Zeng, and Furu Wei. 2021. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing* 16 (2021), 1505–1518.
- [5] Steven Davis and Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, 4 (1980), 357–366.
- [6] Sivan Ding, You Zhang, and Zhiyao Duan. 2022. SAMO: Speaker Attractor Multi-Center One-Class Learning For Voice Anti-Spoofing. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), 1–5.
- [7] Cunhang Fan, Jun Xue, Jianhua Tao, Jiangyan Yi, Chenglong Wang, Chengshi Zheng, and Zhao Lv. 2024. Spatial reconstructed local attention Res2Net with F0 subband for fake speech detection. *Neural Networks* (2024), 106320.
- [8] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-Augmented Generation for Large Language Models: A Survey.
- [9] Yinlin Guo, Haofan Huang, Xi Chen, He Zhao, and Yuehai Wang. 2024. Audio Deepfake Detection With Self-Supervised Wavlm And Multi-Fusion Attentive Classifier. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 12702–12706.
- [10] Ameer Hamza, Abdul Rehman Rehman Javed, Farkhund Iqbal, Natalia Kryvinska, Ahmad S Almadhor, Zunera Jalil, and Rouba Borghol. 2022. Deepfake audio detection via MFCC features using machine learning. *IEEE Access* 10 (2022), 134018–134028.
- [11] Guang Hua, A. Teoh, and Haijian Zhang. 2021. Towards End-to-End Synthetic Speech Detection. *IEEE Signal Processing Letters* 28 (2021), 1265–1269.
- [12] Bingyuan Huang, Sanshuai Cui, Jiwu Huang, and Xiangui Kang. 2023. Discriminative Frequency Information Learning for End-to-End Speech Anti-Spoofing. *IEEE Signal Processing Letters* 30 (2023), 185–189.
- [13] Piotr Kawa, Marcin Plata, Michal Czuba, Piotr Szymański, and Piotr Syga. 2023. Improved DeepFake Detection Using Whisper Features. *International Speech Communication Association (Interspeech)* abs/2306.01428 (2023).
- [14] Tomi H. Kinnunen, Kong-Aik Lee, Héctor Delgado, Nicholas W. D. Evans, Massimiliano Todisco, Md. Sahidullah, Junichi Yamagishi, and Douglas A. Reynolds. 2018. t-DCF: a Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification. (2018).
- [15] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [16] Xu Li, Xixin Wu, Hui Lu, Xunying Liu, and Helen Meng. 2021. Channel-wise gated res2net: Towards robust detection of synthetic speech attacks. *International Speech Communication Association (Interspeech)* (2021).
- [17] Anwei Luo, Enlei Li, Yongliang Liu, Xiangui Kang, and Z Jane Wang. 2021. A capsule network based approach for detection of audio spoofing attacks. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6359–6363.
- [18] Juan M. Martín-Donas and Aitor Álvarez. 2022. The Vicomtech Audio Deepfake Detection System Based on Wav2vec2 for the 2022 ADD Challenge. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), 9241–9245.
- [19] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. (2023), 28492–28518.
- [20] Chengzhe Sun, Shan Jia, Shuwei Hou, and Siwei Lyu. 2023. AI-Synthesized Voice Detection Using Neural Vocoder Artifacts. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2023), 904–912.
- [21] Hemlata Tak, Massimiliano Todisco, Xin Wang, Jee weon Jung, Junichi Yamagishi, and Nicholas W. D. Evans. 2022. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. *Speaker Odyssey Workshop* (2022).
- [22] Hemlata Tak, Jee weon Jung, Jose Patino, Madhu R. Kamble, Massimiliano Todisco, and Nicholas W. D. Evans. 2021. End-to-End Spectro-Temporal Graph Attention Networks for Speaker Verification Anti-Spoofing and Speech Deepfake Detection. *ASVspoof 2021 Workshop-Automatic Speaker Verification and Spoofing Countermeasures Challenge* (2021).
- [23] Massimiliano Todisco, Xin Wang, Ville Vestman, Md. Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas W. D. Evans, Tomi H. Kinnunen, and Kong-Aik Lee. 2019. ASVspoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In *International Speech Communication Association (Interspeech)*.
- [24] Xin Wang and Junich Yamagishi. 2021. A comparative study on recent neural spoofing countermeasures for synthetic speech detection. *International Speech Communication Association (Interspeech)* (2021).
- [25] Xin Wang and Junichi Yamagishi. 2022. Investigating self-supervised front ends for speech spoofing countermeasures. *The Speaker and Language Recognition Workshop* abs/2111.07725 (2022).
- [26] Jee weon Jung, Hee-Soo Heo, Hemlata Tak, Hye jin Shim, Joon Son Chung, Bong-Jin Lee, Ha jin Yu, and Nicholas W. D. Evans. 2021. AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks. *International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), 6367–6371.
- [27] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al. 2021. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. In *ASVspoof 2021 Workshop-Automatic Speaker Verification and Spoofing Countermeasures Challenge*.
- [28] Jiangyan Yi, Ruiho Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Ye Bai, Cunhang Fan, et al. 2022. ADD 2022: the first audio deep synthesis detection challenge. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 9216–9220.
- [29] Jiangyan Yi, Jianhua Tao, Ruiho Fu, Xinrui Yan, Chenglong Wang, Tao Wang, Chu Yuan Zhang, Xiaohui Zhang, Yan Zhao, Yong Ren, Leling Xu, Jun Zhou, Hao Gu, Zhengqi Wen, Shan Liang, Zheng Lian, Shuai Nie, and Haizhou Li. 2023. ADD 2023: the Second Audio Deepfake Detection Challenge. *ArXiv* abs/2305.13774 (2023).
- [30] Jiangyan Yi, Chenglong Wang, Jianhua Tao, Xiaohui Zhang, Chu Yuan Zhang, and Yan Zhao. 2023. Audio Deepfake Detection: A Survey. *ArXiv* abs/2308.14970 (2023).
- [31] Yuxiang Zhang, Wenchao Wang, and Pengyuan Zhang. 2021. The Effect of Silence and Dual-Band Fusion in Anti-Spoofing System. In *International Speech Communication Association (Interspeech)*.
- [32] Pedram Abdzadeh Ziabary and Hadi Veisi. 2021. A countermeasure based on cqt spectrogram for deepfake speech detection. In *2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*. IEEE, 1–5.