algorithms

MDPI

*Article*

# Audio Anti-Spoofing Based on Audio Feature Fusion

**Jiachen Zhang [1,*], Guoqing Tu [1,*], Shubo Liu [2] and Zhaohui Cai [2]**

[1] Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[2] School of Computer Science, Wuhan University, Wuhan 430072, China
* Correspondence: zjcat2021@whu.edu.cn (J.Z.); tugq@whu.edu.cn (G.T.)

**Abstract:** The rapid development of speech synthesis technology has significantly improved the naturalness and human-likeness of synthetic speech. As the technical barriers for speech synthesis are rapidly lowering, the number of illegal activities such as fraud and extortion is increasing, posing a significant threat to authentication systems, such as automatic speaker verification. This paper proposes an end-to-end speech synthesis detection model based on audio feature fusion in response to the constantly evolving synthesis techniques and to improve the accuracy of detecting synthetic speech. The model uses a pre-trained wav2vec2 model to extract features from raw waveforms and utilizes an audio feature fusion module for back-end classification. The audio feature fusion module aims to improve the model accuracy by adequately utilizing the audio features extracted from the front end and fusing the information from timeframes and feature dimensions. Data augmentation techniques are also used to enhance the performance generalization of the model. The model is trained on the training and development sets of the logical access (LA) dataset of the ASVspoof 2019 Challenge, an international standard, and is tested on the logical access (LA) and deep-fake (DF) evaluation datasets of the ASVspoof 2021 Challenge. The equal error rate (EER) on ASVspoof 2021 LA and ASVspoof 2021 DF are 1.18% and 2.62%, respectively, achieving the best results on the DF dataset.

**Keywords:** deep learning; wav2vec 2.0; automatic speaker verification; deep-fake detection; ASVspoof Challenge



**Citation:** Zhang, J.; Tu, G.; Liu, S.; Cai, Z. Audio Anti-Spoofing Based on Audio Feature Fusion. *Algorithms* **2023**, *16*, 317. https://doi.org/10.3390/a16070317

Academic Editor: Yue Duan

Received: 15 May 2023
Revised: 23 June 2023
Accepted: 27 June 2023
Published: 28 June 2023



## 1. Introduction

In recent years, many researchers have proposed end-to-end speech synthesis systems based on deep learning, such as Tacotron2 [1], Deep Voice 3 [2], Transformer TTS [3], and FastSpeech 2 [4]. Synthesized speech has dramatically improved in human-likeness, realism, and naturalness. Speech synthesis software for different languages (such as Chinese, English, Vietnamese, etc.) has been made available to the public, with reduced barriers to entry and difficulty. The highly authentic synthetic speech segments mentioned will pose significant challenges to human auditory perception. Moreover, if these synthesized speech segments are maliciously exploited in authentication systems such as ASV, they can pose a grave security threat.

Training a traditional ASV system that is robust to attacks with synthetic speech requires a dataset that includes various types of synthetic speech generated by different algorithms. However, it is not practical to construct a training dataset that contains all possible attack scenarios, as attackers will always use unknown attack methods to target ASV systems. Therefore, when designing ASV systems, reliability in the face of various and unpredictable attacks [5] needs to be considered, and the generalization ability of ASV systems needs to be improved.

The ASVspoof Challenge has collected a large amount of speech segment data suitable for training ASV systems and constructed a public dataset [5–8] to improve the detection capability of ASV systems against unknown spoofing attacks. The task of the ASVspoof

2019 Challenge includes two scenarios: logical access (LA) and physical access (PA). The LA scenario consists of two attacks against ASV systems: speech synthesis and conversion. The PA scenario explicitly refers to the attack method of recording and replaying. The ASVspoof 2021 Challenge adds a deep-fake (DF) partition aimed at addressing the scenario of speech synthesis using deep-fake technology. This research mainly focuses on the LA and DF scenarios.

In recent years, more and more researchers have attempted to improve the detection capability of ASV systems by building deep learning models. Many studies [9–11] have proposed effective network architectures, such as using raw waveforms as input to obtain better representations. However, the robustness of the model is still limited by the insufficiency of the training data. To address this issue, some recent studies [12,13] have focused on using a pre-trained model at the front end of the existing model and using transfer learning to obtain more generalized feature representations.

However, most recent research has focused solely on the ASVspoof 2019 dataset, with very little attention given to ASVspoof 2021, particularly the DF partition. ASVspoof 2021 DF is a new task within ASVspoof, which extends the focus of the ASVspoof Challenge beyond ASV scenarios to detecting spoofed speech in non-ASV contexts. The DF task reflects scenarios in which attackers access victims' speech data, such as data published by victims on social media. Attackers may use public data and deep-fake technologies to generate synthetic speech with the voice of a victim and use such fake speech to engage in fraudulent activities, such as spreading false information, blackmailing, and causing social unrest and panic. They may also generate synthetic speeches of influential figures in social media or politics to spread fake news and sow discord or create artificial voices of ordinary individuals to deceive their family members of fraud. Additionally, attackers may generate harmful speech data to defame their victims. Therefore, developing better techniques for detecting synthetic speech generated using deep-fake technologies is an urgent problem that needs to be addressed.

The main contribution of this paper is designing a model with a lower equal error rate (EER) on both the ASVspoof 2021 LA and DF partitions. We use a pre-trained feature extraction model and a designed audio feature fusion mechanism for back-end classification.

The evaluation results on the ASVspoof 2021 LA and DF databases [8] show that our proposed model outperforms most of the previous works in terms of the minimum-detection cost function (min t-DCF) [14] and equal error rate (EER) evaluation criteria. In particular, it surpasses the best results mentioned in [15] on the DF dataset.

## 2. Methods

The model presented in this paper consists of two main components: wav2vec 2.0 and a feature fusion module. Our model architecture is illustrated in Figure 1. We first input the raw audio wave into wav2vec 2.0 to extract corresponding audio features. Then, the obtained audio features were embedded with positional information through an embedding module. The resulting hidden layer features were subsequently fed into the audio feature fusion module, which consists of two parts: remix and fusion. Finally, the output results were obtained through a fully connected (FC) layer.

### *2.1. Wav2vec 2.0*

Traditional detection methods often use carefully designed handcrafted features, such as MFCC, LFCC, CQCC, and other mathematically-based audio feature maps. These audio features can be easily and quickly obtained and contain valuable information.

In [16], the authors used LFCC and CQT as inputs to an end-to-end dual-branch network using a multitask learning method to detect different types of synthetic speech. Specifically, each branch added a forgery-type classifier as an additional task. Then, through adversarial training between the network and the forgery-type classifier, the learned features did not contain specific forgery information associated with a particular forgery type. Additionally, the convolutional block attention module (CBAM) was used to improve

the representation ability of the learned features. Using handcrafted features effectively reduces the number of parameters in the model and speeds up the training process, but it may decrease the model accuracy.

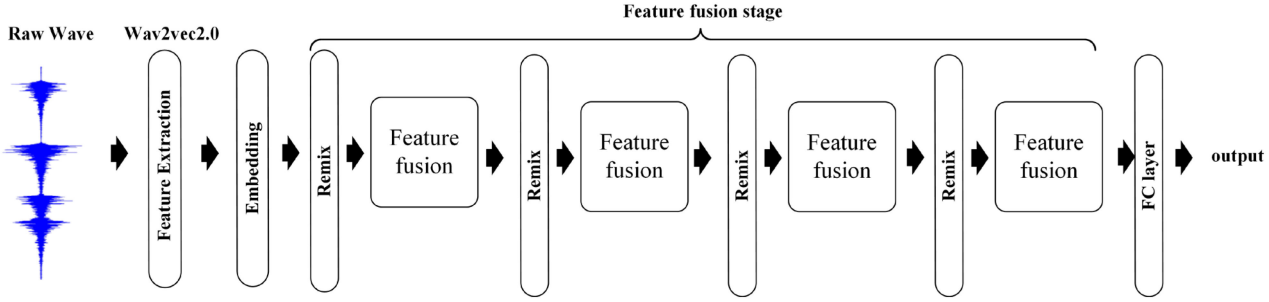


**Figure 1.** Overall framework diagram of our model. From the figure, it can be seen that the original waveform was input into wav2vec 2.0 for feature extraction. Then, the features obtained through embedding were input into the feature fusion stage. The feature fusion stage is composed of four repeated remix and feature fusion blocks. The final output was obtained through a fully connected (FC) layer.

With the continuous development of deep learning, a SincNet-based audio feature extraction method has emerged, which is an interpretable neural network structure designed to process temporal speech signals and directly learn more meaningful filters. Generally, for processing temporal signals, the extraction capability of the first-layer network is crucial because the effectiveness of the low-dimensional features extracted in the first layer is a prerequisite for the high-dimensional feature information learned by the higher layers of the network to be meaningful. In RawGAT-ST [10], for instance, the model directly operates on the raw waveform and uses a Sinc convolutional filter for front-end feature learning. This approach of utilizing neural networks to extract front-end features effectively improves the accuracy of the model.

However, both of the above feature extraction methods have certain limitations. Handcrafted features based on mathematical principles inevitably need to be aware of hidden information. Although neural network-based feature extraction can effectively extract hidden information, it is limited by the model parameters and training data size, leading to suboptimal performance.

The emergence of transfer learning can effectively solve the above problems. Using pre-trained models can significantly reduce training time, and ideal results can be achieved with a small amount of data. Therefore, more and more researchers are using pre-trained models to extract features [12,13,17,18] to improve model performance.

Wav2vec 2.0 [19] is a self-supervised learning framework proposed by Facebook AI, which introduces a transformer to replace RNN structures. The potent ability of the transformer makes wav2vec 2.0 much better than previous models. Although wav2vec 2.0 was initially used for automatic speech recognition (ASR), recent research, such as speech error detection [20,21], speaker recognition [22,23], emotion recognition [24], and deception detection [12,25], has shown that wav2vec 2.0 can also achieve good performance in other speech-related tasks. In this paper, we also used the wav2vec 2.0 model to perform synthetic speech recognition using the pre-trained model.

Since the pre-trained model only uses natural speech data (without synthetic speech data) for training, according to the research [12], fine-tuning with training data containing synthetic speech can potentially improve the model detection performance. To better adapt to downstream training tasks, we added a fully connected layer to the output of the wav2vec 2.0 encoder.

### 2.2. Remix

The well-known Swin-transformer [26] consists of four stages in patch merging and attention. Our model drew on this architecture, where the remix and feature fusion in the audio feature model correspond to patch merging and the transformer block in the Swin-transformer, respectively.

Patch merging in the Swin-transformer is a pooling-like operation that does not lose information. It consists of two steps: first, four adjacent patches are combined into one patch, with the channel number doubled to match the neural network; second, a 1D convolution is applied to the channel dimension to reduce the channel number.

However, in our model, the input audio features were not partitioned into patches, so there were no adjacent parts to concatenate. Therefore, we evenly divided the entire audio feature into four patches in this module, and the audio features (T, F) were composed of two dimensions: timeframe (T) and feature dimension (F), as shown in Figure 2, denoted as $X_1$, $X_2$, $X_3$, and $X_4$. These four patches were then fused using addition, resulting in a patch size of (T/2, F/2). To ensure consistency between the input and output as well as feature fusion, we fed the features obtained through addition into a two-dimensional Conv1d, resulting in features with a size of (4, T/2, F/2). The input channel number of the two-dimensional convolution was 1, and the output channel number was 4. The resulting feature map (4, T/2, F/2) was finally reshaped to the same size as the input feature (T, F).

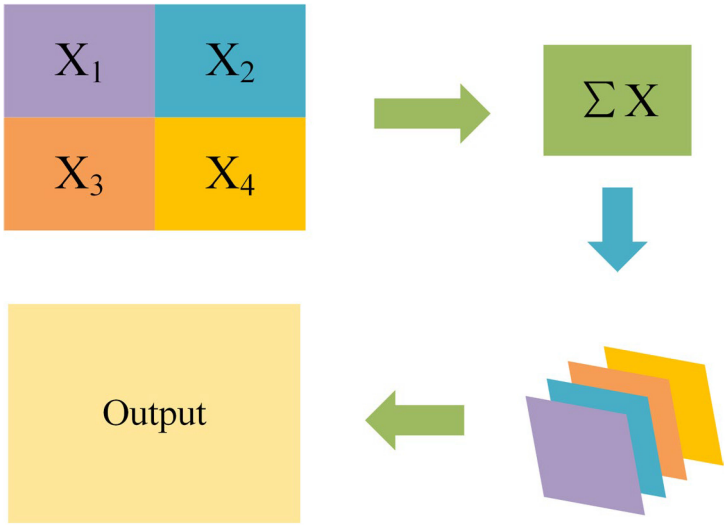


**Figure 2.** Schematic diagram of remix. We segmented and summed the features and then restored them through convolution and reshaping.

### 2.3. Feature Fusion

Google proposed a transformer in [27]. It relies solely on the attention mechanism and has made significant breakthroughs in NLP. Meanwhile, the transformer has also shown good performance in the computer vision (CV) area. Various variants of the transformer [26,28] based on the attention mechanism have continuously set new state-of-the-art records in CV, surpassing the SOTA models.

The original transformer was primarily based on the self-attention mechanism. The self-attention mechanism first multiplies the input vector by three matrices (representing three different weights) to obtain three new vectors (*QKV*). Then, the attention output is obtained using the dot product:

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{softmax}(\frac{\text{QK}^{\text{T}}}{\sqrt{\text{d}_\text{k}}})\text{V} \quad (1)$$

The attention mechanism used in the transformer structure is complex dot-product attention. Instead, a multi-head attention mechanism was designed to improve the perfor-

mance of the model. This mechanism divides the $QKV$ vectors into n segments along the last dimension and performs attention operations on each segment, separately. The results are then concatenated and passed through a fully connected layer to obtain the final output. Multi-head attention consists of n heads, where the weights for the $i$-th head are denoted as $W_{i,q}$, $W_{i,k}$, and $W_{i,v}$:

$$\text{head}_\text{i} = \text{Attention}(q \times W_i^q, k \times W_i^k, v \times W_i^v) \tag{2}$$

$$\text{Multihead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h}) W^O \tag{3}$$

Audio is a different kind of data from text sequences and images. In the past, when processing audio signals, most of the analyses would start from both the time domain and the frequency domain, and the audio features extracted by wav2vec2 also have two dimensions, timeframe and feature dimension. Therefore, after the pre-training model in wav2vec2 extracts the input audio, the feature map size is (T, F), where T represents the timeframe dimension, and F represents the feature dimension.

The original transformer would have focused on the information in the input feature map from only one perspective. This is due to the nature of natural language processing (NLP), where the input data are temporal. However, audio data are not only temporal, but frequency domain information is also important.

Transformer variants in the computer vision (CV) domain, such as those of [26,28], cannot effectively handle audio features with one less dimension. This is because the feature map of image data is three-dimensional, with height, width, and channels (H, W, C), whereas audio features are usually two-dimensional. Therefore, we cannot, in most cases, directly use these models for audio features.

Many studies [13,29] have chosen to adapt audio features to models in the CV domain by increasing the dimensionality of the features, for example, through convolution, so that two-dimensional audio features become three-dimensional and can thus be directly used with models in the CV domain. In most cases, good results have been achieved in this way.

In our model, however, we wanted to try to keep the dimensionality of the audio features the same. Therefore, we redesigned the attention blocks in the overall framework of the Swin-transformer. Unlike NLP, which is overly concerned with temporal information, we treated the timeframe and feature dimension equally.

Unlike other feature fusion approaches, we could not fuse feature maps from different inputs since we only had one feature map as our input. Inspired by [28], we took a single-input feature map and fused the output of the branch by feeding it into different branches. At the same time, we added part of the self-attention to the weight matrix as additional information.

Firstly, the input audio feature (T, F) (denoted as $X_T$) was linearly transformed into the corresponding $QKV$, marked as $Q_f$, $K_f$, and $V_f$, by multiplying with the weight matrix, $W_t$. Meanwhile, the audio feature was transposed to (F, T) (denoted as $X_F$) and similarly linearly transformed into $Q_t$, $K_t$, and $V_t$ using the weight matrix, $W_f$:

$$(Q, K, V) = X \times (W_q, W_k, W_v) \tag{4}$$

where $X$ represents the input audio feature matrix (including $X_T$ and $X_F$), and $W$ represents the corresponding weight matrix (including $W_T$ and $W_F$). The modified attention mechanism was then applied for feature fusion.

To better integrate the timeframe and feature dimension information, a transformation, as shown in Figure 3, was applied to the $V$ in each set of $Q$, $K$, and $V$. $V$ was input into a one-dimensional convolution, Conv1d. After passing through BatchNorm and GELU functions, it was passed through another one-dimensional convolution, Conv1d, to obtain the output, $V'$. Finally, we obtained the matrix $V$ required by the attention mechanism by performing a dot product between the transpose of $V'$ and the $V$ in another set of $QKV$.

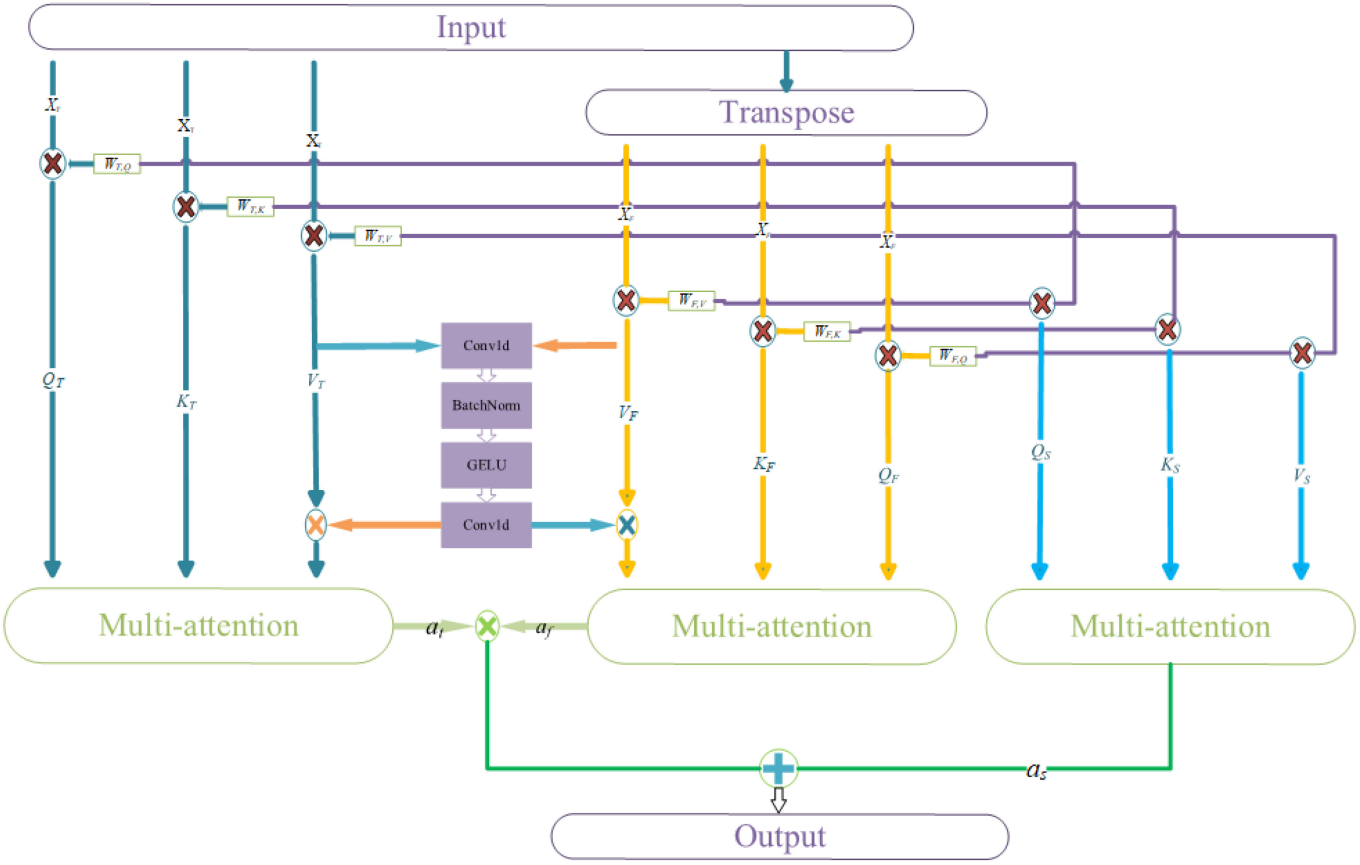


**Figure 3.** We fused $V$ from two different sets of $QKV$ and used the fused $V'$ for multi-attention. We also use the weights at linear transformation as input to the third multi-attention and its output as additional information.

The required attention matrix was obtained by inputting the obtained $V$ and the corresponding $Q$ and $K$ into the attention formula, denoted as $a_t$ and $a_f$, corresponding to the input $X_T$ and $X_F$, respectively.

Meanwhile, for the weight matrix, $W$, used in linear transformation, the following operations were performed:

(a) The size of the weight matrix $W_T$ corresponding to $X_T$ was (T, T). After linear transformation, $W'_T$ with a size of (T, 1) was obtained. Therefore, the three weight matrices, $W_{T,q}$, $W_{T,k}$, and $W_{T,v}$, corresponding to $X_T$, were obtained as $W'_{T,q}$, $W'_{T,k}$, and $W'_{T,v}$, respectively. Similarly, the three weight matrices, $W_{F,q}$, $W_{F,k}$, and $W_{F,v}$, corresponding to $X_F$, were obtained as $W'_{F,q}$, $W'_{F,k}$, and $W'_{F,v}$:

$$\begin{bmatrix} w_{1,1} & \cdots & \cdots & \cdots & w_{1,T} \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ w_{T,1} & \cdots & \cdots & \cdots & w_{T,T} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ \vdots \\ \vdots \\ a_T \end{bmatrix} = \begin{bmatrix} w\prime_1 \\ \vdots \\ \vdots \\ \vdots \\ w\prime_T \end{bmatrix}$$
$$\begin{bmatrix} w_{1,1} & \cdots & \cdots & \cdots & w_{1,F} \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ w_{F,1} & \cdots & \cdots & \cdots & w_{F,F} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ \vdots \\ b_F \end{bmatrix} = \begin{bmatrix} w\prime_1 \\ \vdots \\ \vdots \\ \vdots \\ w\prime_F \end{bmatrix} \quad (5)$$

(b) ${W'}_{T,q}$ and ${W'}_{F,q}$ were multiplied to obtain the matrix $Q_s$ of size (T, F). Similarly, $K_s$ and $V_s$ were obtained:

$$\begin{bmatrix} w\prime_1 \\ \vdots \\ w\prime_T \end{bmatrix} \begin{bmatrix} w\prime_1 & \dots & w\prime_F \end{bmatrix} = \begin{bmatrix} w\prime_{1,1} & \dots & \dots & w\prime_{1,F} \\ \vdots & & & \vdots \\ w\prime_{T,1} & \dots & \dots & w\prime_{T,F} \end{bmatrix} \tag{6}$$

(c) The attention calculation was performed on $Q_s$, $K_s$, and $V_s$ to obtain the corresponding matrix $a_s$:

$$a_s = \text{Multihead}(Q_s, K_s, V_s) \tag{7}$$

Finally, three attention matrices, $a_t$, $a_f$, and $a_s$, were obtained, and the transpose matrix of $a_t$ was dot-multiplied with $a_f$, and the result was then added to $a_s$ to obtain the final output matrix:

$$\text{attention} = (a_f \times {a_t}^T + a_s) \tag{8}$$

### *2.4. Data Augmentation*

Data augmentation (DA) alleviates the issue of insufficient data in deep learning, which was initially widely used in the image domain and then extended to NLP, achieving success in many tasks. DA can reduce overfitting, thus improving generalization [30–33], and is particularly effective in LA scenarios involving large amounts of encoding, transmission, and acquisition equipment. Although many DA solutions have been proposed, such as in [34] and SpecMix [35], they are only suitable for detection models operating on 2D front-end representations.

The model here used RawBoost DA [33] tools, a data augmentation and enhancement method used to design more robust models that directly takes raw waveform inputs, meeting the needs of the model. RawBoost has developed various data augmentation methods. Mode 1 is the processing of data through convolutional noise, mode 2 processes the data through impulse noise, mode 3 processes the data through colored additive noise, and mode 4 is where the three algorithms of modes 1, 2, and 3 are concatenated together for data processing. Mode 5, conversely, is the data processing of two algorithms of modes 1 and 2 in series. Based on [13], we conducted experiments on our model using modes 3 and 5.

## 3. Experimental Setup and Details

The ASVspoof database consists of three partitions for evaluating logical access (LA), physical access (PA), and deep-fake (DF), with the DF partition being newly added in ASVspoof 2021. The model experiments were based on the ASVspoof partitions.

### *3.1. Dataset and Evaluation Criteria*

The ASVspoof 2019 LA database corresponds to a synthesized deception attack scenario, which includes genuine and spoofed speech data generated by 17 different TTS and VC systems. The speech data in the ASVspoof 2019 LA dataset is based on the VCTK corpus, including natural speech from 107 speakers: 46 males and 61 females. The natural speech all has the same recording configuration without channel and background noise interference. The LA partition is divided into three subsets: training, development, and evaluation, with each subset consisting of genuine and spoofed speech. Six different TTS/VC algorithms are used to generate spoofed speech in the training and development subsets, while thirteen different TTS/VC algorithms are used to generate spoofed speech in the evaluation subset.

ASVspoof 2021 LA aims to reduce the gap between ideal laboratory conditions and expected field conditions. It includes a collection of genuine and spoofed speech transmitted through various telephone systems, including IP telephony (VoIP) and public switched

telephone networks (PSTN). Speech data may undergo a series of unknown changes when transmitted across different telephone systems, which can result in artefacts not only from forgery but also from coding and transmission. Therefore, the challenge in the 2021 LA task is to investigate the robustness against harmful changes caused by artefacts resulting from compression, packet loss, and variations in bandwidth, transmission facilities, and bit rates. The speech data are sourced from the ASVspoof 2019 LA evaluation database, which originates from the VCTK database. The spoofing experiments are generated using 1 of 13 different VC, TTS, or hybrid spoofing attack algorithms.

ASVspoof 2021 DF extends the detection of spoofed speech to non-ASV scenarios. The DF evaluation data collect genuine and spoofed speech segments generated using different lossy codecs. The evaluation source data are taken from the ASVspoof 2019 LA evaluation set and other sources, and the ASVspoof 2021 DF evaluation data are generated using over 100 different spoofing algorithms.

According to the rules of the ASVspoof Challenge, the model in this paper was trained on the ASVspoof 2019 LA training and development sets and evaluated on the ASVspoof 2021 LA and ASVspoof 2021 DF evaluation subsets. The evaluation criteria used the tools provided by ASVspoof, including the equal error rate (EER) and the minimum-detection cost function (min t-DCF).

### *3.2. Experimental Details*

All model implementations in this paper were conducted using the PyTorch framework, explicitly utilizing the wav2vec 2.0 XLS-R (0.3B) model. Raw audio was input into the wav2vec 2.0 XLS-R model in approximately 4-second segments containing 64,600 sampling points, resulting in a feature representation size of (201, 1024). After embedding, the resulting feature size was (256, 512) and was input into an audio feature fusion module.

A classification token (CLS) was added to the feature embedding to improve the classification accuracy. The CLS was then passed through a linear layer to obtain the necessary classification output.

The model utilized four back-end classification layers, each containing a feature mixing module and n feature fusion modules. The value of n for each of the four layers was 3, 3, 9, and 3, respectively.

During training, the model utilized the standard Adam optimizer with a fixed learning rate of 0.0001. The weighted cross-entropy loss was used to mitigate the impact of data imbalance between natural and synthetic speech in the training dataset, with a weight ratio of 9:1. The batch size was set at 16. All models were trained for 50 epochs on a single GeForce Tesla v100.

### *3.3. Simple Back-End Classification Model*

In our model, some modifications were made to the self-attention mechanism. To verify that our changes to the self-attention mechanism were effective compared to the original self-attention mechanism, we chose to replace the modified self-attention mechanism in the feature fusion blocks (as shown in Equation (8)) with a simple multi-head attention mechanism, while the remaining components remained unchanged:

$$\text{attention} = \text{Multihead}(Q, K, V) \tag{9}$$

where $Q$, $K$, and $V$ are the $Q_f$, $K_f$, and $V_f$, corresponding to $X_F$, as mentioned in Section 2.3.

## 4. Results

This section is divided into five subsections. The first presents the results of our model on the ASVspoof 2021 LA partition and DF partition using different Rawboost modes. The second and third subsections compare the results of our model with other models on the ASVspoof 2021 LA and DF partitions. The model in the second subsection does not use the wav2vec2 pre-training model, while the model in the third subsection uses

the wav2vec2 pre-training model. The fourth subsection shows the results for the simple back-end classification model mentioned in Section 3.3, and the fifth subsection shows the results obtained with the wav2vec2 pre-training model and Rawboost, but without the back-end classification model.

### *4.1. Data Augmentation Results*

We report the experimental results of our model on both LA and DF partitions using two data augmentation techniques of Rawboost (mode 3 and mode 5), which are presented in Table 1. From the experimental results, it can be seen that on the LA partition, Rawboost with mode 5 achieved better results, while on the DF partition, Rawboost with mode 3 achieved better results. Therefore, we used the optimal results corresponding to the respective partitions in the subsequent experimental results.

**Table 1.** Our model obtained these results using two different Rawboost data augmentation methods, mode 3 and mode 5, on LA and DF partitions, respectively.

| Partition | EER (%) | Min t-DCF |
|---|---|---|
| LA (3) | 3.71 | 0.2880 |
| LA (5) | 1.18 | 0.2171 |
| DF (3) | 2.62 | |
| DF (5) | 4.72 | |

### *4.2. Comparison with Models That Do Not Use the Pre-Trained Model*

As shown in Table 2, in this study, we used the experimental results provided by the ASVspoof 2021 LA partition to compare our model with the baseline models, including Rawnet2 [36], LFCC-LCNN, CQCC-GMM, and LFCC-GMM. Among these models, LFCC-LCNN had the best performance, with an EER of 8.90%, while the worst model, LFCC-GMM, had an EER of 21.13%. Our proposed model showed significant improvements compared to LFCC-LCNN and LFCC-GMM, indicating good performance on the ASVspoof 2021 LA partition.

**Table 2.** The results of our model and other studies on the ASVspoof 2021 LA partition.

| Model | EER (%) | Min t-DCF |
|---|---|---|
| SSL_Anti-spoofing [13] | 0.82 | 0.2066 |
| ours | 1.18 | 0.2171 |
| T23 [8] | 1.32 | 0.2177 |
| wav2+FF layer [37] | 3.54 | 0.2780 |
| fusion [38] | 4.66 | 0.2882 |
| wav2+LLGF [12] | 7.18 | 0.3590 |
| LFCC-LCNN | 8.90 | 0.3152 |
| Rawnet2 [36] | 9.49 | 0.4192 |
| AASIST [29] | 11.47 | 0.5081 |
| CQCC-GMM | 15.80 | 0.4948 |
| LFCC-GMM | 21.13 | 0.5836 |

AASIST [29] is a model that performed well in the ASVspoof 2019 LA partition. It applies heterogeneous graphs for synthetic speech recognition and has the characteristics of few parameters and a high recognition accuracy. Due to the excellent performance of AASIST in the ASVspoof 2019 LA partition, many current studies, including SASV [39],

use AASIST as the baseline. However, the EER of AASIST on the ASVspoof 2021 LA task was 11.47%, which is much worse than our model's performance.

According to the results on the official website of ASVspoof 2021 LA, the best result was achieved by team T23 [8], with 1.32%, and our model also outperformed this result.

### *4.3. Comparison with Models Using the Pre-Trained Model*

Since our model uses the wav2vec2 pre-training model, we compared it with other models [12,37] that also use wav2vec2. We used the same wav2vec2 pre-trained model for feature extraction as these models. We differed in using our respective back-end classification modules and different data augmentation techniques. As can be seen from the results in Table 2, the model in [12] had an EER of 7.18% on the ASVspoof 2021 LA partition, and that in [37] had an EER of 3.54% on the ASVspoof 2021 LA partition, both of which are higher than our EER of 1.18%. As can be seen in Table 3, the model in [37] had an EER of 4.98% on the ASVspoof 2021 DF partition, which is also higher than ours of 2.62%. Therefore, our back-end classification module showed an improvement compared to the other back-end classification modules.

**Table 3.** The results of our model and other studies on the ASVspoof 2021 DF partition.

| Model | EER (%) |
|---|---|
| ours | 2.62 |
| SSL_Anti-spoofing | 2.85 |
| wav2+FF layer [37] | 4.98 |
| Rawnet2 [36] | 6.10 |
| T23 [8] | 15.64 |

SSL_Anti-spoofing [13] is a model proposed in 2022, which is a synthetic speech recognition model that also uses Rawboost data augmentation techniques and wav2vec2 pre-trained models. Our model differs from it in the back-end used. According to the latest literature on ASVspoof 2021 [15], it is the best-performing model for both LA and DF partitions. No model has so far been able to surpass it in effectiveness. Our model also failed to reach its performance on the ASVspoof 2021 LA partition.

However, the proposed model yielded highly satisfactory results on the ASVspoof 2021 DF partition, exhibiting an EER of merely 2.62%, which was slightly higher, by 0.23%, than the SSL_Anti-spoofing model and outperformed all the models mentioned in [15], as displayed in Table 3.

### *4.4. Results of the Simple Back-End Classification Model*

The results in Table 4 are the experimental results of the simple back-end we mentioned in Section 3.3. The Rawboost mode is optimal for both the LA and DF partitions. As seen from Table 4, the EER of the model was 1.58% on the LA partition and 3.18% on the DF partition. Neither performed better than our modified model. However, the results of the simple back-end model outperformed some of the other back-end models [12,37].

**Table 4.** Results of the simple back-end classification model on LA and DF partitions.

| Partition | EER (%) | Min t-DCF |
|---|---|---|
| LA (5) | 1.58 | 0.2286 |
| DF (3) | 3.18 | |

### *4.5. Results of a Model without Back-End Classification*

Table 5 shows the experimental results obtained without using the back-end classification of our design. Since no back-end was used, the features extracted by the pre-trained

model would be directly passed through an averaging pooling layer (AdaptiveAvgPool2d) to obtain the final output.

**Table 5.** Results of a model without back-end classification on LA and DF partitions.

| Partition | EER (%) | Min t-DCF |
|---|---|---|
| LA (5) | 24.84 | 0.8038 |
| DF (3) | 16.91 | |

The results showed that the effect of not using the back-end was much worse than that using the back-end.

## 5. Conclusions

In this work, we modified the self-attention mechanism to make it more suitable for speech tasks by incorporating features from different dimensions. The results showed that our improvement in the self-attention mechanism was effective. We utilized self-supervised pre-training and data augmentation to enhance the performance of the proposed model. Our model achieved promising results on both ASVspoof 2021 LA and DF datasets, especially in the DF partition, where it outperformed the current state-of-the-art model. This study fills the gap in the research on the ASVspoof 2021 DF dataset. However, due to the use of pre-trained models and self-attention mechanisms, our model has a large number of parameters, which may not be suitable for resource-limited scenarios. Future work will focus on reducing the model size, retaining its effectiveness, and making it more widely applicable.

**Author Contributions:** Conceptualization, J.Z. and G.T.; methodology, J.Z. and S.L.; software, J.Z.; validation, J.Z. and G.T.; formal analysis, J.Z. and G.T.; investigation, G.T. and Z.C.; resources, G.T. and Z.C.; data curation, J.Z.; writing—original draft preparation, J.Z. and G.T.; writing—review and editing, J.Z.; visualization, J.Z. and S.L.; supervision, G.T. and S.L.; project administration, G.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** All data included in this study are available upon request by contacting the corresponding author.

**Conflicts of Interest:** We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

## References

1. Shen, J.; Pang, R.; Weiss, R.J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerrv-Ryan, R.; et al. Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, Calgary, AB, Canada, 15–20 April 2018; pp. 4779–4783.
2. Ping, W.; Peng, K.; Gibiansky, A.; Arik, S.O.; Kannan, A.; Narang, S.; Raiman, J.; Miller, J. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv* **2017**, arXiv:171007654.
3. Li, N.; Liu, S.; Liu, Y.; Zhao, S.; Liu, M. Neural speech synthesis with transformer network. In Proceedings of the AAAI Conference on Artificial Intelligence 2019, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6706–6713.
4. Ren, Y.; Hu, C.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; Liu, T.-Y. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv* **2020**, arXiv:200604558.
5. Nautsch, A.; Wang, X.; Evans, N.; Kinnunen, T.H.; Vestman, V.; Todisco, M.; Delgado, H.; Sahidullah, M.; Yamagishi, J.; Lee, K.A. ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech. *IEEE Trans. Biom. Behav. Identity Sci.* **2021**, *3*, 252–265. [CrossRef]
6. Wu, Z.; Kinnunen, T.; Evans, N.; Yamagishi, J.; Hanilçi, C.; Sahidullah, M.; Sizov, A. ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge. In Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 6–10 September 2015.

7. Kinnunen, T.; Sahidullah, M.; Delgado, H.; Todisco, M.; Evans, N.; Yamagishi, J.; Lee, K.A. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. In Proceedings of the 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017.
8. Yamagishi, J.; Wang, X.; Todisco, M.; Sahidullah, M.; Patino, J.; Nautsch, A.; Liu, X.; Lee, K.A.; Kinnunen, T.; Evans, N.; et al. ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection. *arXiv* **2021**, arXiv:210900537.
9. Ma, Y.; Ren, Z.; Xu, S. RW-Resnet: A novel speech anti-spoofing model using raw waveform. *arXiv* **2021**, arXiv:210805684.
10. Tak, H.; Jung, J.; Patino, J.; Kamble, M.; Todisco, M.; Evans, N. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. *arXiv* **2021**, arXiv:210712710.
11. Hua, G.; Teoh, A.B.J.; Zhang, H. Towards end-to-end synthetic speech detection. *IEEE Signal Process. Lett.* **2021**, *28*, 1265–1269. [CrossRef]
12. Wang, X.; Yamagishi, J. Investigating self-supervised front ends for speech spoofing countermeasures. *arXiv* **2021**, arXiv:211107725.
13. Tak, H.; Todisco, M.; Wang, X.; Jung, J.; Yamagishi, J.; Evans, N. Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation. *arXiv* **2022**, arXiv:220212233.
14. Kinnunen, T.; Lee, K.A.; Delgado, H.; Evans, N.; Todisco, M.; Sahidullah, M.; Yamagishi, J.; Reynolds, D.A. t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification. *arXiv* **2018**, arXiv:180409618.
15. Liu, X.; Wang, X.; Sahidullah, M.; Patino, J.; Delgado, H.; Kinnunen, T.; Todisco, M.; Yamagishi, J.; Evans, N.; Nautsch, A.; et al. ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild. *arXiv* **2022**, arXiv:221002437. [CrossRef]
16. Ma, K.; Feng, Y.; Chen, B.; Zhao, G. End-to-End Dual-Branch Network Towards Synthetic Speech Detection. *IEEE Signal Process. Lett.* **2023**, *30*, 359–363. [CrossRef]
17. Ilyas, H.; Javed, A.; Malik, K.M. Avfakenet: A Unified End-to-End Dense Swin Transformer Deep Learning Model for Audio-Visual Deepfakes Detection. *Appl. Soft Comput.* **2023**, *136*, 110124. [CrossRef]
18. Eom, Y.; Lee, Y.; Um, J.S.; Kim, H. Anti-spoofing using transfer learning with variational information bottleneck. *arXiv* **2022**, arXiv:220401387.
19. Baevski, A.; Zhou, Y.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 12449–12460.
20. Xu, X.; Kang, Y.; Cao, S.; Lin, B.; Ma, L. Explore wav2vec 2.0 for Mispronunciation Detection. In Proceedings of the Interspeech 2021, Brno, Czech Republic, 30 August–3 September 2021; pp. 4428–4432.
21. Peng, L.; Fu, K.; Lin, B.; Ke, D.; Zhang, J. A Study on Fine-Tuning wav2vec2. 0 Model for the Task of Mispronunciation Detection and Diagnosis. In Proceedings of the Interspeech 2021, Brno, Czech Republic, 30 August–3 September 2021; pp. 4448–4452.
22. Vaessen, N.; Van Leeuwen, D.A. Fine-tuning wav2vec2 for speaker recognition. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 7967–7971.
23. Fan, Z.; Li, M.; Zhou, S.; Xu, B. Exploring wav2vec 2.0 on speaker verification and language identification. *arXiv* **2020**, arXiv:201206185.
24. Pepino, L.; Riera, P.; Ferrer, L. Emotion recognition from speech using wav2vec 2.0 embeddings. *arXiv* **2021**, arXiv:210403502.
25. Xie, Y.; Zhang, Z.; Yang, Y. Siamese Network with wav2vec Feature for Spoofing Speech Detection. In Proceedings of the Interspeech 2021, Brno, Czech Republic, 30 August–3 September 2021; pp. 4269–4273.
26. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2021, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), London, UK, 4–9 December 2017.
28. Cui, Y.; Jiang, C.; Wang, L.; Wu, G. Mixformer: End-to-end tracking with iterative mixed attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022, New Orleans, LA, USA, 18–24 June 2022; pp. 13608–13618.
29. Jung, J.; Heo, H.-S.; Tak, H.; Shim, H.; Chung, J.S.; Lee, B.-J.; Yu, H.-J.; Evans, N. AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 22–27 May 2022; pp. 6367–6371.
30. Lavrentyeva, G.; Novoselov, S.; Tseren, A.; Volkova, M.; Gorlanov, A.; Kozlov, A. STC antispoofing systems for the ASVspoof2019 challenge. *arXiv* **2019**, arXiv:190405576.
31. Cáceres, J.; Font, R.; Grau, T.; Molina, J.; SL, B.V. The Biometric Vox system for the ASVspoof 2021 challenge. In Proceedings of the ASVspoof 2021 Workshop, Online, 16 September 2021.
32. Das, R.K. Known-unknown data augmentation strategies for detection of logical access, physical access and speech deepfake attacks: ASVspoof 2021. In Proceedings of the ASVspoof 2021 Workshop, Online, 16 September 2021.
33. Tak, H.; Kamble, M.; Patino, J.; Todisco, M.; Evans, N. Rawboost: A Raw Data Boosting and Augmentation Method Applied to Automatic Speaker Verification Anti-Spoofing. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 6382–6386.
34. Park, D.S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E.D.; Le, Q.V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv* **2019**, arXiv:190408779.
35. Kim, G.; Han, D.K.; Ko, H. SpecMix: A mixed sample data augmentation method for training with time-frequency domain features. *arXiv* **2021**, arXiv:210803020.

36. Tak, H.; Patino, J.; Todisco, M.; Nautsch, A.; Evans, N.; Larcher, A. End-to-End anti-spoofing with RawNet2. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2021, Toronto, ON, Canada, 6–11 June 2021; pp. 6369–6373.
37. Martín-Doñas, J.M.; Álvarez, A. The Vicomtech Audio Deepfake Detection System Based on Wav2vec2 for the 2022 ADD Challenge. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 9241–9245.
38. Cohen, A.; Rimon, I.; Aflalo, E.; Permuter, H.H. A study on data augmentation in voice anti-spoofing. *Speech Commun.* **2022**, *141*, 56–67. [CrossRef]
39. Jung, J.; Tak, H.; Shim, H.; Heo, H.-S.; Lee, B.-J.; Chung, S.-W.; Kang, H.-G.; Yu, H.-J.; Evans, N.; Kinnunen, T. SASV challenge 2022: A spoofing aware speaker verification challenge evaluation plan. *arXiv* **2022**, arXiv:220110283.