TO-Rawnet: Improving RawNet with TCN and Orthogonal Regularization for Fake Audio Detection

Chenglong Wang1,2, Jiangyan Yi2,3, Jianhua Tao3,4, Chu Yuan Zhang2,3, Shuai Zhang4, Ruibo Fu2, Xun Chen1

1University of Science and Technology of China, Hefei, China
2State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
3School of Artificial Intelligence, University of Chinese Academy of Sciences, China
4Department of Automation, Tsinghua University

chenglong.wang@nlpr.ia.ac.cn

Abstract

Current fake audio detection relies on hand-crafted features, which lose information during extraction. To overcome this, recent studies use direct feature extraction from raw audio signals. For example, RawNet is one of the representative works in end-to-end fake audio detection. However, existing work on RawNet does not optimize the parameters of the Sinc-conv during training, which limited its performance. In this paper, we propose to incorporate orthogonal convolution into RawNet, which reduces the correlation between filters when optimizing the parameters of Sinc-conv, thus improving discriminability. Additionally, we introduce temporal convolutional networks (TCN) to capture long-term dependencies in speech signals. Experiments on the ASVspoof 2019 show that the Our TO-RawNet system can relatively reduce EER by 66.09% on logical access scenario compared with the RawNet, demonstrating its effectiveness in detecting fake audio attacks.

Index Terms: ASVspoof, fake audio detection, end-to-end, orthogonal convolution

1. Introduction

In the field of fake audio detection, the use of standard hand-crafted feature is common [1, 2, 3, 4]. Linear frequency cepstrum coefficients (LFCC) are a benchmark feature for anti-spoofing tasks [5, 6]. LFCC uses linear filters instead of Mel filters, emphasizing high-frequency features over Mel frequency cepstral coefficients (MFCC). Constant Q cepstral coefficients (CQCC) are derived from a constant-Q transform (CQT) [7, 8], capturing frequency domain features effectively. Additional features, such as group delay gram [9], log power spectrum (LPS) [10], and cochlear filter cepstral coefficients instantaneous frequency (CFCCIF) [11], also perform well. However, standard features may smooth the speech spectrum, hindering the extraction of vital narrow-band speaker traits like pitch [12] and formants. In contrast, direct processing of raw waveforms enables the network to learn task-specific low-level embeddings.

Recently, researchers have shown increased interest in utilizing raw waveforms as system inputs [13, 14, 15, 16]. For example, Dinkel [17] proposes a raw waveform convolutional long short-term neural network (CLDNN) to enhance defense against unknown attacks. Another study [18] improves end-to-end fake audio detection by constructing ResWavegram from one-dimensional convolutions applied to the raw waveform. Moreover, SincNet [14] is a novel neural network architecture that enhances feature extraction using adjustable cutoff frequency parameters in bandpass filters based on Sinc functions, outperforming fixed-parameter Mel filters. Additionally, models like RawNet2 [15] and AASSIST [19] encode raw waveforms using Sinc functions. However, the flexibility of parameter settings may sometimes result in suboptimal local minima, and prior studies have not iteratively optimized Sinc-conv parameters [15, 19, 20].

Studies in image processing have demonstrated that the utilization of orthogonal learned filters can enhance model capacity, leading to improved feature expression and intra-class feature representation [21, 22, 23]. Building upon the work of [24], we enhance RawNet2 [25] by incorporating an orthogonal constraint. The convolution kernels are initialized as linear-scale Sinc filters using band-pass filtered Sinc functions. By enforcing the orthogonality property on the convolution kernels, we reduce correlation between filters. Additionally, convolutional neural networks (CNNs) face challenges in capturing long-term dependencies due to the constraint of kernel size. Drawing inspiration from [26], we employ a temporal convolution network (TCN) instead of Conv1d to expand the receptive field of the convolutional kernel. Our proposed method improves the discriminative power of RawNet2 by extracting more robust features from raw audio signals and capturing the complex temporal dynamics of speech signals. We evaluate the effectiveness of our approach on two benchmark datasets, demonstrating its superiority over other state-of-the-art systems for fake audio detection. Our contributions can be summarized as follows:

• We proposed a novel deep neural network architecture called TO-RawNet for fake audio detection. The model combines the advantages of orthogonal convolution and TCN to improve upon RawNet2. To our best knowledge, this is the first application of the combination of orthogonal convolution and TCN in the field of fake audio detection.

• Compared to RawNet, experiments conducted on the ASVspoof 2019 dataset demonstrate that our TO-RawNet system can significantly reduce EER by 66.09% in the logical access scenario.

The structure of this paper is as follows: Section 2 provides an overview of related work. Section 3 details our proposed method. Experiments, results and discussions are reported in Section 4 and 5, respectively. Finally, we conclude the paper in Section 6.

2. Related Work

Recently, more and more researchers have been using raw waveform inputs directly in the field of fake audio detection.
SincNet [14] is a neural network architecture that is designed to operate directly on the raw waveform of audio signals. The first layer of SincNet consists of a bank of band-pass filters that are parametrized as Sinc functions, allowing for the extraction of useful features directly from the raw waveform. By using a constrained first layer with fewer learnable parameters, SincNet is able to learn a more meaningful filterbank structure, resulting in more meaningful output.

RawNet2 [25], another neural network architecture, also employs a bank of band-pass filters parametrized as Sinc functions to extract features from the raw waveform. The upper layers of RawNet2 consist of residual blocks and gated recurrent units (GRUs) [27], with the addition of end-to-end architectures based on learned features rather than knowledge-based and hand-crafted features. Studies have shown that using end-to-end architectures based on learned filters rather than knowledge-based and hand-crafted features has the potential to improve the performance of fake audio detection.

3. Proposed Methods

3.1. Orthogonal Convolution

This paper is based on the differentiable frontend of Sinc-conv, and aims to improve feature expressiveness and intra-class feature representation by using orthogonal convolutions and regularization constraints. The specific operational steps are as follows: as shown in Figure 1 (b), we view the convolution operation as a matrix-vector multiplication, where the kernel matrix \( M \) is generated by the convolution kernel \( K \). Using the linear property of the convolution operation, we adopt the Doubly Block-Toeplitz (DBT) matrix construction method to transform the convolution expression \( \text{Conv}(K, X) \) into a faster DBT matrix-vector representation, as shown below:

\[
Y = \text{Conv}(K, X) \iff y = Mx
\]

where \( M \) is the DBT matrix, and \( x \) and \( y \) represent the input and output tensors, respectively. The shape of the DBT matrix \( M \) is \( K \in \mathbb{R}^{(O \times 1) \times (I \times 1)} \). Where \( O \) and \( I \) are the output and input channels, and \( T_2 \) and \( T_1 \) are the feature map lengths of the output and input. And its rows need to be orthogonalized to reduce the correlation between filters. The orthogonality condition of the DBT matrix \( M \) is shown in equation (2):

\[
\left\langle M_{i_1', j_1'}, M_{j_2', i_2'} \right\rangle = \begin{cases} 1 & (i_1', j_1') = (j, i_2') \\ 0 & \text{else} \end{cases}
\]

where \( i_1' \) and \( j_2' \) represent two different filter positions, and \( i \) and \( j \) represent the corresponding row positions of these filters in the matrix. However, since \( M \) is highly structured and sparse, a more efficient method for orthogonal calculation was proposed in [24], as shown in equation (3):

\[
Y = \text{Conv}(K, \text{padding} = P, \text{stride} = S) = I_o \quad (3)
\]

where \( K \) represents the size of the convolution kernel, \( S \) represents the stride, and \( P = \left\lfloor \frac{K-1}{2} \right\rfloor \cdot S \) represents padding. \( I_o \) is a tensor, where the center is an \( n \times n \) identity matrix, and the rest is padded with zeros. By minimizing the difference between \( Z = \text{Conv}(K, \text{padding} = P, \text{stride} = S) \) and \( I_o \), a roughly orthogonal convolution can be obtained. The loss function of the orthogonal convolution can be expressed as:

\[
\min_K L_{\text{orth}} = \|Z - I_o\|^2_F \quad (4)
\]

The final training loss is as follows:

\[
L = L_{\text{task}} + \lambda L_{\text{orth}} \quad (5)
\]

Where \( L_{\text{task}} \) represents the loss of the classification task, and \( \lambda \) is the weight of the orthogonal regularization loss, and we set three different values in the experiment. Please refer to Algorithm 1 for the specific pseudocode implementation, where \( o_{\text{c}} \) represents output channels and \( i_{\text{c}} \) represents input channels.
Algorithm 1 Detailed Procedure of Orthogonal Convolution.

```python
function deconv_orth_dist(kernel, stride, padding):
    [o_c, i_c, kernel_size] = kernel.shape
    output = conv_1d(kernel, kernel, stride, padding)
    target = zeros((o_c, o_c, output.shape[-1]))
    center = floor_divide(output.shape[-1], 2)
    target[:, center] = eye(o_c)
    return norm(subtract(output, target))
```

3.2. Temporal Convolution Network

Inspired by TCN [25], we propose the dilated convolution block to extract features, as shown in Figure 1 (c). The residual block first employs batch normalization and the leaky ReLU activation function, followed by dilated convolution. Next, a 1x1 convolution is used to adjust the output channels to match the input channels. To expedite convergence and facilitate the training of deeper models, we incorporate a residual [28] path. Each block’s output serves as the input for the subsequent block. The dilation factor is doubled for each block up to a certain limit and then repeated (e.g., 1, 2, 4, …, 2^n). This exponential increase in dilation factor ensures that the model captures sufficient temporal contextual information for detecting fake audio. It enlarges the network’s receptive field and captures forgery traces in the entire speech with fewer stacked layers.

3.3. TO-Rawnet

Figure 1 (a) illustrates the architecture of our proposed TO-Rawnet system. First, the raw waveform is fed into the Sinc-conv layer with orthogonal regularization to produce a high-level speech representation. The orthogonal regularization helps reduce redundancy by enabling each filter to focus on distinct frequency components. Next, the high-level representations are fed into the residual module, which includes dilated convolutions with exponentially increasing receptive fields. This enables the network to effectively increase its perception range, allowing it to capture more global information from the input audio. Subsequently, we connect a GRU to extract an utterance-level representation, which is then fed into a softmax activation function to perform real/fake classification.

4. Experiments

4.1. Dataset

4.1.1. ASVspoof 2019 Challenge Dataset

ASVspoof 2019 LA [5] mainly has 19 spoofing attack algorithms (A01-A19), with two types of spoofing attacks: text to speech (TTS) and voice conversion (VC). The LA data set contains three subsets: the training set, the development set, and the evaluation set. Table 1 details the number of real and fake audio of the ASVspoof2019 LA dataset. The attack algorithms in the training and development sets overlap, while the evaluation set includes unseen spoofing attacks.

4.1.2. ASVspoof 2021 Challenge Dataset

ASVspoof 2021 LA [6] poses greater challenges than the previous versions. Although the training and development sets remain the same as those of ASVspoof 2019 LA database, the evaluation set is distinct. Specifically, the evaluation data for 2021 LA contains encoding and transmission artifacts that stem from actual telephony systems.

Table 1: The detailed information of ASVspoof2019 LA dataset and ASVspoof2021 LA dataset.

<table>
<thead>
<tr>
<th>Set</th>
<th>Genuine # utterance</th>
<th>Spoofed # utterance</th>
<th>Total # utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>2,580</td>
<td>22,800</td>
<td>25,380</td>
</tr>
<tr>
<td>Dev</td>
<td>2,548</td>
<td>22,296</td>
<td>24,844</td>
</tr>
<tr>
<td>Eval(2019 LA)</td>
<td>7,355</td>
<td>64,578</td>
<td>71,933</td>
</tr>
<tr>
<td>Eval(2021 LA)</td>
<td>18,452</td>
<td>163,114</td>
<td>181,566</td>
</tr>
</tbody>
</table>

4.2. Experimental Setup

The audio sampling rate is 16kHz. To form batches, we standardized the duration of the raw waveform input to approximately 4 seconds (64600 samples) by either truncating longer utterances or concatenating shorter ones. The Sinc-conv layers have a filter length of l = 129, a stride of d = 1, and utilize n = 128 filters. We used fixed linear-scale Sinc filters. To enhance the performance of our model, we utilized six residual-blocks architecture that consists of 12 dilated convolution blocks with varying dilation factors, where the highest dilation factor is 32. In order to prevent over-fitting and under-fitting during the training process, we experimented with different channel combination configurations to determine the optimal combination. The number of channels in the first two residual blocks and the last four residual blocks are set to (32, 64), (128, 256), and (256, 512), respectively. We named them small (S), medium (M), and large (L), in that order. To further improve the discriminative power of our model, we employed FMS independently for each residual-block output. This technique enhances the most informative filter outputs and improves the overall accuracy of the model. To aggregate frame-level representations into an utterance-level representation, we utilized a GRU layer with 1024 hidden nodes. The output of the GRU layer is passed through a softmax activation function, which produces two-class predictions, i.e., real or fake. We propose an orthogonal regularization loss in this paper and set three different weights λ for this loss: 0.05, 0.1, and 0.2.

To train the model, we use the Adam optimizer with a learning rate of 5×10^{-5}. We set the batch size to 32. The model is trained for 150 epochs. The training set is used to train the model, the development set is used to select the model with the best performance, and finally, the evaluation set is used for evaluation. The results of the ASVspoof2021 competition suggest that data augmentation can reduce overfitting and improve generalization [6, 29, 30]. To this end, in our experiments on ASVspoof2021, we utilized data augmentation techniques. Specifically, we employed the open-source tool RawBoost for performing data augmentation in the LA task. We added linear and nonlinear convolutional noise and impulsive signal-dependent additive noise in the LA database.

In this work, in order to evaluate the results of different fake audio detection systems, the equal error rate (EER) [31] is used as the evaluation metric.

5. Results and Discussion

5.1. Ablation Experiments

Table 2 reveals that the hyperparameters perform better when λ is set to 0.1. As a result, for subsequent experiments, we have kept λ fixed at 0.1. The Orth-RawNet-M based model consis-

---

1https://github.com/TakHemlata/RawBoost-antispoofing
Table 2: Orth-RawNet based models tested on the ASVspoof2019 LA evaluation set. Orth-RawNet-S, Orth-RawNet-M and Orth-RawNet-L with different values of $\lambda$. Results are the average (best) obtained from three runs of each experiment with different random seeds.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\lambda$</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orth-RawNet-S</td>
<td>0.05</td>
<td>4.39 (4.15)</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>3.82 (3.63)</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>4.51 (4.43)</td>
</tr>
<tr>
<td>Orth-RawNet-M</td>
<td>0.05</td>
<td>3.86 (3.57)</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>3.19 (3.06)</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>3.52 (3.36)</td>
</tr>
<tr>
<td>Orth-RawNet-L</td>
<td>0.05</td>
<td>3.78 (3.65)</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>3.66 (3.59)</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>3.94 (3.73)</td>
</tr>
</tbody>
</table>

Table 3: The ablation experiments on the ASVspoof2019 LA and ASVspoof2021 LA evaluation sets are represented by EER1 and EER2, respectively. Results are the average (best) obtained from three runs of each experiment with different random seeds.

<table>
<thead>
<tr>
<th>Methods</th>
<th>EER1</th>
<th>EER2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RawNet</td>
<td>4.66</td>
<td>5.31</td>
</tr>
<tr>
<td>Orth-RawNet-S</td>
<td>3.82 (3.63)</td>
<td>5.02 (4.86)</td>
</tr>
<tr>
<td>Orth-RawNet-M</td>
<td>3.19 (3.06)</td>
<td>4.81 (4.73)</td>
</tr>
<tr>
<td>Orth-RawNet-L</td>
<td>3.66 (3.59)</td>
<td>4.62 (4.55)</td>
</tr>
<tr>
<td>TCN-RawNet-S</td>
<td>3.43 (3.37)</td>
<td>3.26 (3.13)</td>
</tr>
<tr>
<td>TCN-RawNet-M</td>
<td>2.86 (2.62)</td>
<td>5.08 (4.87)</td>
</tr>
<tr>
<td>TCN-RawNet-L</td>
<td>3.25 (3.14)</td>
<td>5.12 (4.96)</td>
</tr>
<tr>
<td>TO-RawNet-S</td>
<td>1.97 (1.86)</td>
<td>4.05 (3.84)</td>
</tr>
<tr>
<td>TO-RawNet-M</td>
<td>1.58 (1.23)</td>
<td>3.70 (3.58)</td>
</tr>
<tr>
<td>TO-RawNet-L</td>
<td>2.56 (2.37)</td>
<td>3.93 (3.78)</td>
</tr>
</tbody>
</table>

Table 4: Performance comparison of the proposed methods to some known single systems on the ASVspoof2019 LA evaluation set.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Front-end</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQCC-GMM (Baseline1) [34]</td>
<td>CQCC</td>
<td>9.57</td>
</tr>
<tr>
<td>LFCC-GMM (Baseline2) [34]</td>
<td>LFCC</td>
<td>8.09</td>
</tr>
<tr>
<td>ST-RawNet2 [25]</td>
<td>Raw waveform</td>
<td>3.64</td>
</tr>
<tr>
<td>S2-RawNet2 [25]</td>
<td>Raw waveform</td>
<td>5.13</td>
</tr>
<tr>
<td>S3-RawNet2 [25]</td>
<td>Raw waveform</td>
<td>4.66</td>
</tr>
<tr>
<td>Resnet18-OC-softmax [32]</td>
<td>LFCC</td>
<td>2.19</td>
</tr>
<tr>
<td>MCG-Res2Net50 [33]</td>
<td>CQT</td>
<td>1.78</td>
</tr>
<tr>
<td>AASIST [19]</td>
<td>Raw waveform</td>
<td>1.13</td>
</tr>
<tr>
<td>TO-RawNet (ours)</td>
<td>Raw waveform</td>
<td>1.58</td>
</tr>
<tr>
<td>Orth-AASIST (ours)</td>
<td>Raw waveform</td>
<td>1.02</td>
</tr>
</tbody>
</table>

6. Conclusions

We propose a new end-to-end fake speech detection system named TO-RawNet, which has two new contributions: (i) using orthogonal regularization to constrain the learning process of filters, thereby improving the ability of feature expression and intra-class feature representation; (ii) introducing TCN to capture long-term dependencies in time-series data. Compared to RawNet, our TO-RawNet system reduces the EER by 66.09% in logical access scenarios. Furthermore, we apply the orthogonal regularization technique to the SOTA single-system AASIST and observe performance improvement, verifying the generalizability of orthogonal regularization. In the future, we will verify the performance of TO-RawNet across datasets and further improve its performance on backend models.

7. Acknowledgments

This work is supported by the National Key Research and Development Plan of China (No.2020AAA0140003), the National Natural Science Foundation of China (NSFC) (No.61831022, No.U21B2010, No.62101553, No.61971419, No.62006223, No.62276259, No.62201572, No.62206278), Beijing Municipal Science & Technology Commission, Administrative Commission of Zhongguancun Science Park No.Z211100004821013, Open Research Projects of Zhejiang Lab (NO. 2021HK0A0B06).
8. References


