CLAD: Robust Audio Deepfake Detection Against Manipulation Attacks with Contrastive Learning

Haolin Wu, Jing Chen, Ruiying Du, Cong Wu, Kun He,
Xingcan Shang, Hao Ren, Guowen Xu

***Abstract*—The increasing prevalence of audio deepfakes poses significant security threats, necessitating robust detection methods. While existing detection systems exhibit promise, their robustness against malicious audio manipulations remains underexplored. To bridge the gap, we undertake the first comprehensive study of the susceptibility of the most widely adopted audio deepfake detectors to manipulation attacks. Surprisingly, even manipulations like volume control can significantly bypass detection without affecting human perception. To address this, we propose CLAD (Contrastive Learning-based Audio deepfake Detector) to enhance the robustness against manipulation attacks. The key idea is to incorporate contrastive learning to minimize the variations introduced by manipulations, therefore enhancing detection robustness. Additionally, we incorporate a length loss, aiming to improve the detection accuracy by clustering real audios more closely in the feature space. We comprehensively evaluated the most widely adopted audio deepfake detection models and our proposed CLAD against various manipulation attacks. The detection models exhibited vulnerabilities, with FAR rising to 36.69%, 31.23%, and 51.28% under volume control, fading, and noise injection, respectively. CLAD enhanced robustness, reducing the FAR to 0.81% under noise injection and consistently maintaining an FAR below 1.63% across all tests. Our source code and documentation are available in the artifact repository (https://github.com/CLAD23/CLAD).**

***Index Terms*—Audio Deepfake Detection, Manipulation Attacks, Contrastive Learning**

I. Introduction

In recent years, deep learning has made remarkable progress in speech synthesis [1]–[3] and voice conversion [4]–[6]. These technologies advanced the creation of highly realistic and natural-sounding speech, which enhances user interaction in various applications. However, they also pose serious risks. Malicious attackers may use these technologies to generate high-quality audio deepfakes and then exploit them for phone scams [7], bypassing speaker recognition [8] and spreading disinformation on the web [9]. In response, researchers have developed various methods to detect audio deepfakes, including acoustic features such as spectral features [10], linear frequency cepstral coefficients (LFCC) [11] , and constant Q cepstral coefficients (CQCC) [12], as well as deep learning approaches such as RawNet2 [13], Res-TSSDNet [14] and AASIST [15]. These methods have shown impressive performance on large-scale public datasets like the ASVspoof dataset [16], [17].

Despite the impressive performance in detecting audio deepfakes, these methods' effectiveness and robustness in real-world scenarios have yet to be investigated, especially when faced with simple but natural audio manipulations. Most existing methods are evaluated using raw output of synthesis techniques, neglecting the possible manipulations applied to the audio by the adversary [11], [13]–[15]. While the latest ASVspoof 2021 dataset [17] introduces more realistic data subjected to telephony channel transmission and compression, these modifications are significantly limited, and fail to reflect the techniques a real attacker might employ to fool the detection system. Therefore, a thorough understanding of these methods' robustness against simple manipulations is essential to ensure their efficacy in mitigating deepfake threats.

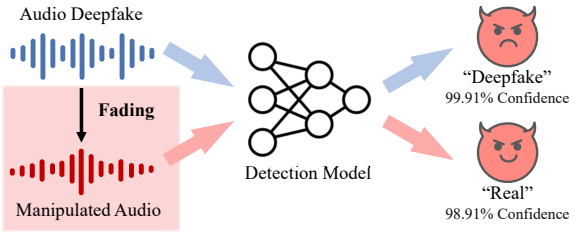


Fig. 1. Illustration of the manipulation attacks. Though detection model performs well on the original audio, we found that simple manipulations like fading could bypass it.

To bridge the gap, we perform the first systematic large-scale study on audio manipulation attacks in the audio deepfake detection task, where an attacker aims to deceive the detection system by simply manipulating the audio data in an imperceptible way. For example, the attacker may apply fading to the audio to bypass the detector, as shown in Fig. 1. There have been previous works [18]–[20] that use adversarial attacks to evade detection systems, these attacks are costly and often require prior knowledge of the target system. In contrast, the manipulation attack is more realistic and cheaper, posing a serious threat to detection systems. Specifically, we design 7 manipulations, i.e., noise injection, volume control, fading, time stretching, resampling, time shifting, echoes adding, and investigate the robustness of widely adopted detection methods against these manipualtions. We observe that these methods fail to counteract the manipulation attacks and the performance drops significantly under these attacks.

In this paper, we propose a Contrastive Learning-based Audio deepfake Detector (CLAD) to enhance the robustness against audio manipulation attacks. The key idea is to use contrastive learning to train a robust audio encoder that effectively differentiates audio deepfakes by learning intrinsic features.

By training the encoder to produce similar feature representations for the same audio under different augmentations, and dissimilar representations for different audios, the encoder can learn robust features that are more resistant to manipulation attacks. To further enhance the effectiveness of CLAD, we developed length loss, which compels the encoder to produce short feature vectors for real audios and long feature vectors for audio deepfakes. By clustering the features of real audios around the origin of the high-dimensional space and pushing the features of deepfakes away, the length loss effectively utilizes label information to improve the detection performance of CLAD. The key insight is that real audios exhibit more resemblance to one another than audio deepfakes, which are generated from distinct synthesis methods.

In summary, our paper makes the following contributions:

- We perform the first comprehensive study on the robustness of widely adopted audio deepfake detection methods against audio manipulation attacks and expose the significant threat of manipulation attacks. We find that these detection methods are vulnerable to simple manipulation attacks, such as volume control and fading.
- To the best of our knowledge, we present the first audio deepfake detection method that is robust to manipulation attacks, namely CLAD. To achieve this, we introduce contrastive learning, which learns a robust encoder to produce reliable features for different audio manipulations. To enhance the accuracy, we also design length loss as a new learning strategy to enhance the encoder.
- We conduct extensive experiments to evaluate the effectiveness of CLAD in defending against manipulation attacks under different scenarios. The results suggest that CLAD is effective and robust against manipulation attacks. e.g., achieving an overall FAR of less than 1.63% across all manipulations. CLAD also shows the significant improvements over existing methods.

II. Manipulation Attacks

A. Manipulations

To attack the audio deepfake detection systems, adversaries attempt to manipulate the audio with simple and common methods. We discuss many kinds of manipulation approaches as never before, some of which have received little attention in prior work [21], [22].

Noise injection applies additive noise to the audio signal, which is a common occurrence in real-world environments. We consider two typical noise types, gaussian white noise (WN) and environmental noise (EN). For our experimental evaluation, we consider the following environmental noises: wind, footsteps, breathing, coughing, rain, clock tick and sneezing. We control the strength of the noise by adjusting the signal-to-noise ratio (SNR).

Volume control (VC) scales the magnitude of a speech signal, while not changing the semantics of the speech signal, but only its loudness. Since human perception of loudness is logarithmic, humans are not sensitive to linear changes in volume. In contrast, the model takes the audio as a sequence of sampling values that are linearly related to the volume.

Fading (FD), including fade in and fade out, adds a smooth transition to the beginning and end of an audio, which makes the audios sound more natural. We apply five different fading shapes to the original signal, namely: linear, logarithmic, exponential, quarter sinusoidal and half sinusoidal.

Time stretching (TS) modifies the speed or duration of an audio signal without affecting its pitch. It is commonly used in audio signal processing to match the tempo of two songs, or to change the duration of an audio signal to fit a fixed length.

Resampling (RS) changes the sampling rate of an audio signal, a technique commonly employed in digital signal processing. This method converts signals from one sampling rate to another, ensuring compatibility between audio files, devices, or to decrease the size of an audio file.

Time shifting (SF) is a process to shift the audio in time domain, which can be used to delay or advance the audio signal by a certain amount of time. For human perception, regardless of how the audio is shifted, the auditory experience remains unchanged. However, for the model, the audio is a sequence of sampling values, the position of the sampling values do impact the model's performance.

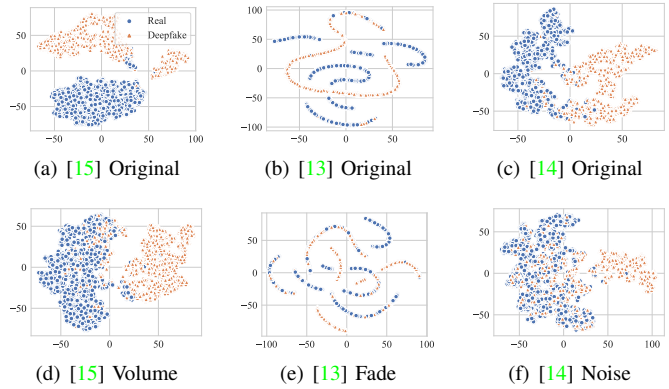
Echoes adding (EC) applies echoes to a speech signal by creating delayed and attenuated copies of the original signal and adding them together. Adding echoes makes the speech sound more reverberant, depending on the delay and attenuation factors, while not affecting human recognition of speaker and content.

It is important to note that the manipulations described here are widely employed in audio processing. This simplifies the implementation and ensures that the resulting audio remains perceptually natural to humans. [1] While prior research has considered pitch shift, time masking, and frequency masking [21], [22], we opt not to incorporate these manipulations into our study due to their potential to introduce perceptually unnatural audio artifacts and affect human recognition of speaker identity and content, contradicting the purpose of audio deepfake.

B. Manipulation Attacks vs. Adversarial Attacks

Manipulation attacks are similar to adversarial attacks in that both aim to evade detection systems and achieve malicious purposes. However, manipulation attacks offer three key advantages. First, manipulation attacks are simple and do not require professional knowledge of machine learning. Second, manipulation attacks are computationally efficient. Adversarial attacks is based on optimization, making the generation of successful adversarial examples expensive. In contrast, the manipulations used in our work can be applied quickly to audio deepfakes. The third advantage is that manipulation attacks do not need any prior knowledge of the detection model. Traditional adversarial attacks require full information of the target model, while black box adversarial attacks struggle with transferability issues [20]. We believe these advantages make manipulation attacks more accessible to attackers, posing a serious threat to audio deepfake detection systems.

[1] Our manipulated audio samples are accessible at the following URL: https://sites.google.com/view/clad-demo/.



(a) [15] Original (b) [13] Original (c) [14] Original

(d) [15] Volume (e) [13] Fade (f) [14] Noise

Fig. 2. Visualization of origianl vs. manipulated features extracted by widely adopted detection models via t-SNE. (a) and (d) are features extracted by AASIST, (b) and (e) are features extracted by RawNet2, (c) and (f) are features extracted by Res-TSSDNet.

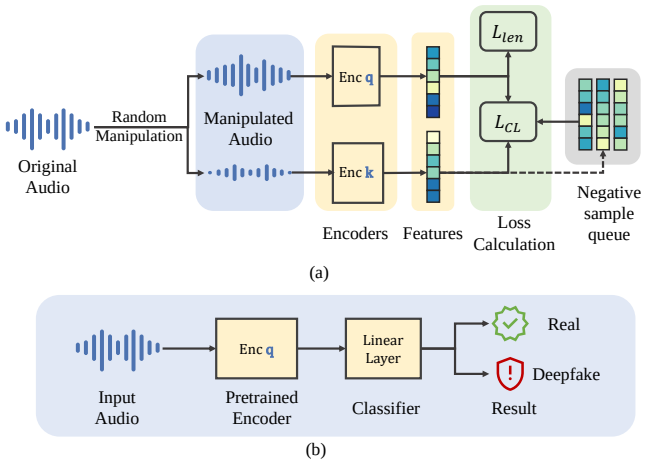


Fig. 3. Overview of CLAD. (a) illustrates the pretraining stage. (b) illustrates the downstream training stage.

III. Motivation

To expose the impact of manipulation attacks, we employ three common techniques: volume control, fading, and noise injection as illustrations to manipulate the audio deepfake samples. Fig. 2 visualizes the feature distribution extracted by three widely adopted models, AASIST [15], RawNet2 [13] and Res-TSSDNet [14] via t-distributed stochastic neighbor embedding (t-SNE). We can observe that compared with original distribution in Fig. 2(a-c), the average distance between manipulated audio deepfakes and real samples is significantly reduced in Fig. 2(d-f). Consequently, this leads to a notable degradation in the performance of the detection models when dealing with manipulated samples.

The underlying cause of this phenomenon is multifaceted. Firstly, existing detection methods have not accounted for the manipulations that could be potentially applied to the data. Secondly, these methods were predominantly trained using conventional supervised learning paradigms, thereby optimized to make predictions based on the specific examples they encountered during training. Manipulations cause the features extracted by these models changed significantly, resulting in the misclassification of the manipulated samples. This observation motivates us to design a robust model capable of withstanding potential manipulations. However, conventional supervised learning approaches often struggle to accommodate such variations due to their inherent limitations in adaptability.

In response to this challenge, we explore the potential of contrastive learning, a promising technique that empowers models to learn invariant and discriminative representations by distinguishing between similar and dissimilar examples based on their features. This enables models to capture the intrinsic characteristics of input audio, even in the presence of diverse manipulations. In contrast, conventional supervised learning approaches often struggle to handle such variations since they are tailored to specific instances and lack the inherent adaptability required. Motivated by these insights, we propose a contrastive learning-based detection model that learns more robust representations.

Furthermore, we observed that that models trained solely on contrastive learning extract features with poor clustering properties, making it challenging to achieve satisfactory performance in the detection task. Given the observation that real samples are more similar to each other than audio deepfakes, we introduce length loss which clusters real audio representations by controlling length of the feature vector. While contrastive learning focuses on the directionality of features, length loss targets their magnitude, thereby not only avoiding interference with the contrastive learning process but also complementing it, boosting the model's performance.

IV. CLAD Design

A. Overview

CLAD is constructed based on the contrastive learning framework presented in [23]. We enhance the framework with additional audio augmentations and length loss. The CLAD model for audio deepfake detection task consists of two main components: an encoder and a linear classifier. The training of the CLAD model is conducted in two phases: pre-training and downstream training, as s illustrated in Fig. 3.

During pre-training, our objective is to train a robust encoder that can produce representative features. Firstly, we randomly apply different manipulations to the input training data, resulting in two manipulated samples. We then feed the manipulated samples into the encoders to obtain the corresponding features. In the context of contrastive learning, samples augmented from the same audio are deemed positive pairs, while those from different audios are considered negative pairs. The encoder is trained to produce similar features for positive pairs and dissimilar features for negative pairs. This training paradigm facilitates the model in learning to generate consistent features for samples, even when subjected to different manipulations, while still maintaining discriminative features across different audios. To enhance the contrastive training, we employ a queue to store previous negative samples, as suggested in [23]. The contrastive loss is computed using the features output by the encoders and negative samples stored in the queue. Additionally, we introduce length loss to aggregate features for

real samples, thereby augmenting downstream performance. Further details regarding the contrastive loss and length loss are elaborated in following sections. The final loss function for pre-training is expressed as the weighted sum of the contrastive loss and length loss, as shown in Eq. 1.

$$\mathcal{L}_{pretrain} = \mathcal{L}_{CL} + \lambda \mathcal{L}_{len} \tag{1}$$

In the downstream training stage, we utilize the pre-trained encoder and add a linear classifier on top of it. We then train the model on the audio deepfake detection task using the labeled dataset to obtain the final detector.

Note that we do not specify the encoder architecture in CLAD. Instead, most deep learning-based end-to-end detection methods can be used as the encoder by removing the classification head. This makes CLAD a plug-and-play solution that can be easily integrated with existing detection methods to improve their robustness.

B. Contrastive Learning

Contrastive learning aims to learn consistent and discriminative representations by training a model to map similar instances closer together and dissimilar instances farther apart in a latent feature space. We employ contrastive learning to train the encoder to produce consistent representations under various manipualtions, enhancing the model's resistance against manipulation attacks.

While prior contrastive learning methods for audio [24], [25] primarily relied on the SimCLR framework [26], we opt for MoCo [23] because it allows for the use of a larger number of negative samples through the maintenance of a negative sample queue. This framework employs two different encoders, named query encoder and key encoder, which are the Enc q and Enc k shown in Fig. 3. The encoders have identical architecture, and the parameters of the key encoder are smoothly updated by using the parameters of the query encoder. [2] They take the different augmentations of input and produce corresponding features, q and k. The augmentations include all the manipulations discussed in Sec. II and are randomly applied during training. We train the encoder using contrastive loss to ensure that the generated feature k is as close as possible to feature q compared to the negative sample features in the queue. This enables the encoder to distinguish between different samples under manipulations. The contrastive loss is defined in Eq. 2.

$$\mathcal{L}_{CL} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(q_i^T k_i^+/\tau)}{\sum_{j=1}^{K} \exp(q_i^T k_j/\tau)} \tag{2}$$

where N is the batch size, K is the size of the negative sample queue, τ is the temperature parameter, and q_i and k_i^+ are the feature q and feature k of the i-th sample in the batch and k_j is the feature k of the j-th sample in the negative sample queue. After the contrastive loss is computed, we update the parameters of the query encoder with gradient descent. The parameters of the key encoder are updated by using the momentum update rule formulated in Eq. 3.

$$\theta_k = \mu\theta_k + (1-\mu)\theta_q \tag{3}$$

where θ_k and θ_q are the parameters of the key encoder and the query encoder, respectively, and μ is the momentum parameter. Finally, we enqueue the feature k in the batch to the negative sample queue and dequeue the earliest features in the queue to update the queue.

[2]The design of momentum update and negative sample queue aims to increase the negative sample size and stabilize the training process, thereby enhancing the model performance. We refer readers to [23] for more details.

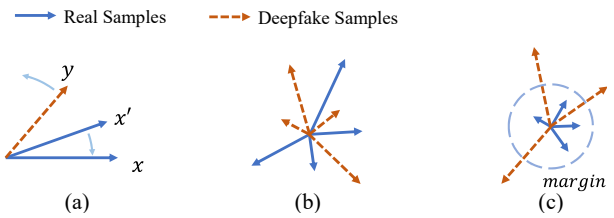


Fig. 4. Illustration of the motivation of length loss. (a) illustrates the training of contrastive loss. (b) illustrates the features extracted by contrastive loss trained encoder. (c) illustrates the features extracted by length loss and contrastive loss trained encoder.

C. Length Loss

The contrastive loss we used is based on the cosine similarity. However, by solely relying on the cosine similarity for calculating the contrastive loss, the model only learns to minimize the angles between feature vectors corresponding to augmented audios from the same source while increasing the angles for different audios, as depicted in Fig. 4 (a). After such training, we observed that the features extracted by the model remain scattered and exhibit poor clustering, as illustrated in Fig. 4 (b). In the context of audio deepfake detection, real speech samples tend to exhibit higher similarity to one another, whereas synthetic speech, generated using different algorithms, often poses challenges for clustering [27]. Building upon this observation, we propose clustering the feature vectors of real audio to enhance performance. To accomplish this, we introduce length loss that leverages the length information, which is not utilized in the cosine similarity loss calculation. Specifically, we decrease the length of feature vectors for real speech and increase it for audio deepfakes, leading to improved clustering of feature vectors for real speech, as demonstrated in Fig. 4 (c). The length loss is defined mathematically as Eq. 4.

$$\mathcal{L}_{len} = \frac{1}{N}\sum_{i=1}^{N} y_i w \|q_i\|_2 + (1-y_i)\max(margin - \|q_i\|_2, 0) \tag{4}$$

where N is the batch size, q_i denotes the feature of the i-th sample in the batch extracted by encoder $Enc\,q$, y_i is the label of the sample, w is the weight factor to assign different levels of importance to the different classes, since the audio deepfake detection dataset might be highly unbalanced. $margin$ is the margin used to control the degree of separation between the real audios and audio deepfakes.

In summary, length loss is designed to encourage the encoder to generate shorter feature vectors for real audios and longer ones for audio deepfakes. This is motivated by the observation that real speech samples generally exhibit more similarity compared to audio deepfakes generated by various algorithms. Besides, when integrated with the contrastive loss, which focuses on the directionality of feature vectors, the length loss provides an additional dimension of discriminative information. This combination of loss functions allows the model to leverage both magnitude and direction of feature vectors, resulting in improved performance in detection.

D. Downstream Training

After training the encoder with both the contrastive and length loss, we combine the encoder Enc q with a linear layer and train it by minimizing the cross-entropy loss for the downstream task. The linear classifier is a one-layer fully connected neural network that maps the encoder output to the classification score. We define the loss function as follows:

$$\mathcal{L}_{downstream} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \quad (5)$$

where N is the batch size, y_i is the label, p_i is the predicted probability of being real for the i-th sample in the batch.

V. Experimental Results

The objective of our evaluation is to answer the following Research Questions(RQs):

- RQ1: How do manipulation attacks degrade the performance of widely adopted audio deepfake detection systems?
- RQ2: How effective is the proposed CLAD model in detecting audio deepfakes under manipulation attacks?
- RQ3: What is the impact on the performance of the CLAD model when contrastive learning or length loss is removed?
- RQ4: How does the performance of the CLAD model change when facing unknown manipulations?

A. Experimental Settings

Baseline Models. We select RawNet2 [13], AASIST [15], Res-TSSDNet [14], and SAMO [28] as baseline models for audio deepfake detection. They are the most widely adopted and influential models in this field, selected as the baseline models for many challenges [17], [29], [30], and have been studied in recent works [31]–[33]. Furthermore, these models are accompanied by publicly available official implementations and pretrained model weights. To assess their performance against manipulation attacks, we utilize the pretrained model weights provided by the authors instead of training the models from scratch. For models with multiple versions, we select the one with the best performance reported.

Dataset. We used the Logical Access (LA) part of the ASVspoof 2019 dataset, which is commonly utilized by audio deepfake detection methods [13], [15], [34]–[37], including the baseline models. We followed baseline methods and trained our models exclusively on the training set of the ASVspoof 2019 LA to ensure a fair comparison. For evaluation, we used only the evaluation set of ASVspoof 2019 LA, excluding the other datasets such as ASVspoof 2021 LA and ASVspoof 2021 DF. This is due to the subpar performance of the pretrained baseline models on these datasets, making it challenging to demonstrate the impact of manipulation attacks.

Metrics. The evaluation metrics utilized in our study comprise the False Acceptance Rate (FAR), False Rejection Rate (FRR), F1 score, and Equal Error Rate (EER). FAR represents the proportion of deepfakes falsely classified as real audio, whereas FRR represents the opposite. Notably, FAR serves as a measure of the success rate of manipulation attacks and is therefore adopted as the primary metric for our evaluation. F1 score is computed as the harmonic mean of precision and recall. The EER corresponds to the point on the Detection Error Tradeoff (DET) curve where the FAR equals the FRR. Notably, the calculation of FAR, FRR, and F1 score necessitates a threshold, and for most evaluations, we adopt the threshold that yields the EER on the original data. Thus in our evaluation, FAR is also the EER on the original data.

Model hyperparameters. We take AASIST [15] model without the final fully connected layer as the encoder of our model to demonstrate the improvement of CLAD. The input audio is repeated or clipped to make the duration is fixed to 64600 samples. For pretraining stage, the model is trained with Adam optimizer using a learning rate of 0.0005, a weight decay of 0.0001, 150 epochs and a mini-batch size of 24. We also use the cosine annealing learning rate decay following the strategy of MoCo and AASIST. The queue size for contrastive learning is reduced to 6144, since large queue size will cause the queue to be filled with the same samples. The temperature τ and the momentum μ are set to 0.07 and 0.999, respectively. Concerning Length loss, we set the margin $margin$ to 4 and the weight w to 9. The weight of the length loss α is set to 2. The pretrain epochs is selected as 150 empirically. For downstream training stage, we train the model with Adam optimizer using a learning rate of 0.001, a weight decay of 0.0001, 10 epochs and a mini-batch size of 16.

Manipulation parameters To evaluate the impact of white noise, we varied the signal-to-noise ratio (SNR) with values of 15, 20, and 25dB. Regarding environmental noise, we added 7 types of noise from the ESC-50 dataset [38] at a constant SNR of 20dB. For volume control, we examined factors of 0.5 and 0.1. To investigate the effect of fading, we used a linear fade shape and varied the fade ratio with values of 0.1, 0.3, and 0.5. Additionally, we examined the influence of five fade shapes, each employing a consistent fade ratio of 0.5. To evaluate the impact of time stretching, we used an FFT length of 128 and varied the time stretching factor with values of 0.9, 0.95, 1.05, and 1.1. For resampling, we evaluated target resampling rates of 15,000, 15,500, 16,500, and 17,000 Hz, which correspond to -1,000, -500, +500, and +1,000 offsets to the original sampling rate of 16,000 Hz. We examined time shifting by considering shift lengths of 1,600, 16,000, and 32,000. To add echoes, we studied two parameters: delay and attenuation factor. We set the delay to 1,000 or 2,000 and the attenuation to 0.2 or 0.5. Additional information about the manipulation parameters is provided in the appendix.

TABLE I
THE FARS (%) AND F1 SCORES (%) OF DIFFERENT AUDIO DEEPFAKE DETECTION METHODS UNDER MANIPULATION ATTACKS.

Models	Types	None	Volume Control		White Noise			Environmental Noise[1]							Time Stretch			
			0.5	0.1	15dB	20dB	25dB	WD	FS	BR	CO	RA	CT	SN	1.1	1.05	0.95	0.9
RawNet2	FAR	4.60	2.78	**36.62**	**18.00**	10.09	7.44	4.02	2.95	3.74	4.30	8.68	7.65	3.37	0.28	0.44	0.33	0.16
	F1	81.08	86.90	**37.16**	**54.24**	67.42	73.38	82.86	86.34	83.74	81.99	70.46	72.88	84.94	96.43	95.78	96.24	96.95
AASIST	FAR	0.83	1.49	**9.12**	0.07	0.13	0.40	0.08	0.26	0.17	0.10	0.10	0.11	0.13	0.16	0.08	0.04	0.03
	F1	96.11	93.51	**71.25**	99.30	99.02	97.89	99.26	98.45	98.84	99.15	99.14	99.12	99.04	98.89	99.26	99.43	99.45
Res-TSSDNet	FAR	1.63	2.19	**10.11**	**51.28**	**39.23**	**29.80**	**36.68**	12.55	3.91	1.68	**40.37**	12.70	2.84	0.19	0.20	0.00	0.00
	F1	92.57	90.50	**68.74**	**30.56**	**36.49**	**43.03**	**38.05**	64.01	84.69	92.39	**35.84**	63.74	88.19	98.34	98.31	99.17	99.17
SAMO	FAR	1.09	3.11	**7.67**	0.22	0.77	1.40	0.38	0.29	0.58	0.12	0.58	0.32	0.46	0.00	0.01	0.00	0.00
	F1	94.94	87.56	**74.50**	98.51	96.22	93.74	97.83	98.19	97.01	98.94	96.98	98.09	97.48	99.44	99.43	99.45	99.45
CLAD	FAR	1.11	0.70	0.06	0.11	0.51	0.82	1.39	1.17	0.68	0.20	0.23	1.15	1.01	0.07	0.12	0.06	0.03
	F1	94.82	96.48	99.17	98.95	97.26	96.01	93.74	94.59	96.56	98.60	98.43	94.68	95.25	99.13	98.91	99.18	99.32

Models	Types	Add Echoes[2]			Time Shift			Fade[3]							Resample			
		1k/ .2	1k/ .5	2k/ .5	1.6k	16k	32k	.5/ L	.3/ L	.1/ L	.5/ E	.5/ Q	.5/ H	.5/ Lo	-1k	-0.5k	+0.5k	+1k
RawNet2	FAR	3.55	1.48	2.71	4.08	3.99	2.14	15.57	7.61	4.79	**27.09**	7.88	**30.42**	4.58	3.82	3.92	3.87	3.67
	F1	84.34	91.62	87.14	82.66	82.92	89.16	57.71	72.97	80.50	**44.30**	72.31	**41.52**	81.14	83.48	83.15	83.33	83.95
AASIST	FAR	0.23	0.07	0.10	0.60	0.75	0.54	9.63	3.60	1.05	**23.53**	3.83	**31.22**	1.23	0.59	0.60	0.97	1.25
	F1	98.59	99.29	99.16	97.05	96.45	97.31	70.13	86.08	95.24	**49.15**	85.32	**42.17**	94.51	97.07	97.03	95.55	94.43
Res-TSSDNet	FAR	2.18	1.29	1.61	2.38	2.25	2.15	2.80	1.68	1.17	4.91	1.83	**11.55**	1.47	0.00	0.00	1.78	2.08
	F1	90.53	93.88	92.66	89.83	90.29	90.64	88.34	92.39	94.35	81.62	91.82	**65.87**	93.17	99.18	99.18	92.01	90.89
SAMO	FAR	0.15	0.04	0.07	0.69	1.89	1.58	**17.65**	10.34	2.16	**25.98**	11.07	**28.24**	4.53	0.38	0.43	1.19	1.55
	F1	98.78	99.28	99.17	96.56	91.89	93.05	**56.17**	68.51	90.89	**46.60**	67.05	**44.53**	83.01	97.84	97.64	94.55	93.16
CLAD	FAR	0.21	0.03	0.07	0.85	0.93	0.89	0.64	0.95	1.06	0.50	0.85	0.80	0.93	0.65	0.81	1.37	1.63
	F1	98.55	99.32	99.15	95.88	95.57	95.74	96.73	95.50	95.05	97.32	95.88	96.07	95.57	96.68	96.03	93.81	92.83

[1] The abbreviations WD, FS, BR, CO, RA, CT, SN stand for wind, footsteps, breathing, coughing, rain, clock tick and sneezing respectively.
[2] 1k/ .2 means that the delay is 1,000 samples and the attenuation factor is 0.2, and the rest is the same.
[3] .5/ L means the ratio is set to 0.5, and using linear fade shape. L, E, Q, H, Lo denote linear, exponential, quarter sinusoidal, half sinusoidal, logarithmic fade shapes respectively.

B. Manipulation Attacks Results (RQ1)

To address the first research question, we conducted a large-scale experiment to investigate the impact of manipulations on the performance of the baseline models. Only the audio deepfakes are manipulated to bypass the detection. The results are presented in the first four rows of Tab. I.

It is evident that all baseline models exhibit excellent performance in the absence of manipulations, with FARs of 4.60%, 0.83%, 1.63% and 1.09% for RawNet2, AASIST, Res-TSSDNet and SAMO, respectively. However, simple volume control significantly increases the FAR, reaching 36.69% for RawNet2 when the control factor is set to 0.1, with similar increases observed for other baselines. Regarding noise injection, RawNet2 is vulnerable to white noise, while Res-TSSDNet experiences severe degradation from white noise, wind noise, and rain noise, as indicated by an FAR as high as 51.28%. Interestingly, AASIST and SAMO performs better, with a lower FAR even compared to the original data, indicating its ability to identify noisy audio as potential deepfakes.

When facing time stretch manipulation, all models consistently yield better results, likely due to the noticeable artifacts introduced by the FFT and inverse FFT processes in time stretching, which are frequently observed in audio deepfakes. For echoes and timeshift, none of the baselines are affected much. Fading with half sinusoidal fade shape is a strong manipulation that consistently achieving high bypass performance with FARs of 30.43%, 31.23%, and 11.55%, 28.24% for RawNet2, AASIST, Res-TSSDNet and SAMO, respectively. Resampling does not show significant influence, consistent with findings in [21] who tested a narrower range. Regarding F1 scores, we observe similar trends as FARs.

To examine how manipulation attacks influence the model's prediction score, we take three representative manipualtion attack settings as examples and analyze the score distribution for the entire dataset output by three baselines. For better visualization, we adopt the approach used by [13], which employs the second element of the final linear layer output as the prediction score, which reflects the softmax output. The distribution of prediction scores for the selected baseline models under various manipulations is presented in Fig. 5. Notably, higher scores indicate higher level of confidence in the authenticity of the sample. We observe that though the original score distribution without manipulation for different models are different, they all share a same trait that the distribution of deepfakes and real audio can be easily separated. However, after manipulation, we observe that the score distribution for the deepfakes shift towards distribution of real audio. Consequently, with the original threshold, we could expect a high FAR for the manipulated samples.

In order to enhance the effectiveness of attacks, a straightforward and intuitive strategy for adversaries is to combine multiple potent manipulations. We assess the performance of baseline models against a combination of manipulation attacks. We employ a representative set of manipulations for evaluation, including: volume control (VC) with a factor of 0.1, white noise (WN) with a SNR of 15 dB, wind environmental noise(EN), time stretch(TS) with a stretching factor of 0.9, fading(FD) with a half sine shape and fade ratio of 0.5, and resampling(RS) to 17,000 Hz. Tab. II presents the results. Each row represents the first manipulation applied, and each column represents the second. The diagonal elements represent the performance of the baseline models under single manipulation attacks. We observe that RawNet2 and Res-TSSDNet exhibit a significantly higher FAR under combined attacks. In contrast, for AASIST and SAMO, the combination of manipulations does not result in a higher FAR.

To sum up, our evaluation indicates that all baseline models are vulnerable to manipulation attacks. Although the baseline models exhibit varying degrees of vulnerability, volume control, noise injection, and fading generally achieve higher

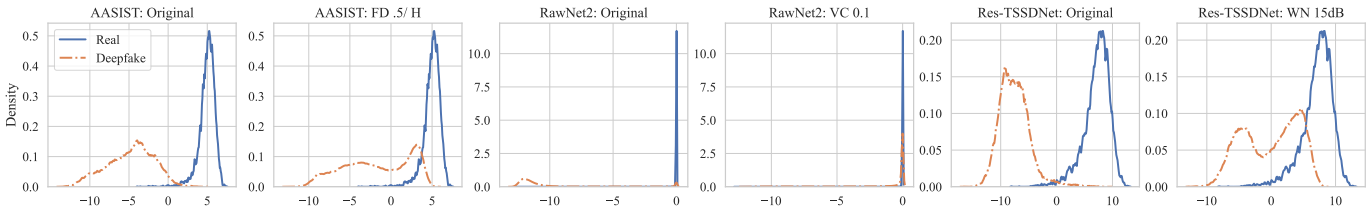


Fig. 5. The prediction score distribution of baseline model output for the whole dataset under various types of manipulations.

TABLE II
THE FARS (%) OF DIFFERENT AUDIO DEEPFAKE DETECTION METHODS UNDER COMBINATION OF MANIPULATION ATTACKS.

(a) RawNet2

	VC	WN	EN	TS	FD	RS
VC 0.1	36.69	86.47	59.9	7.87	33.57	34.07
WN 15dB	86.45	18.02	21.77	5.09	73.86	11.46
EN Wind	59.91	21.90	8.68	1.30	58.67	7.14
TS 0.9	7.89	6.53	1.67	0.16	1.31	0.09
FD .5/ H	33.57	71.7	56.02	1.01	30.43	27.35
RS +1k	34.07	20.03	9.14	0.09	26.02	3.67

(b) AASIST

	VC	WN	EN	TS	FD	RS
VC 0.1	9.15	12.64	10.61	0.45	20.71	9.83
WN 15dB	12.71	0.07	0.06	0.03	15.51	0.14
EN Wind	10.61	0.06	0.10	0.03	26.74	0.24
TS 0.9	0.45	0.03	0.03	0.03	0.31	0.03
FD .5/ H	20.71	0.96	3.28	0.59	31.23	30.57
RS +1k	9.83	0.16	0.23	0.07	32.14	1.25

(c) Res-TSSDNet

	VC	WN	EN	TS	FD	RS
VC 0.1	10.11	73.62	70.24	0.78	6.18	10.61
WN 15dB	73.63	51.28	54.53	0.00	49.17	52.57
EN Wind	70.24	54.43	40.37	0.00	43.00	42.12
TS 0.9	0.78	7.45	2.62	0.00	0.13	0.00
FD .5/ H	6.18	56.3	50.82	0.39	11.55	16.77
RS +1k	10.61	0.96	3.28	0.59	30.57	2.08

(d) SAMO

	VC	WN	EN	TS	FD	RS
VC 0.1	7.67	10.54	9.00	0.00	5.36	8.08
WN 15dB	10.46	0.22	0.15	0.00	21.48	0.27
EN Wind	9.00	0.15	0.38	0.00	26.48	0.79
TS 0.9	0.00	0.01	0.01	0.00	0.03	0.00
FD .5/ H	5.36	16.96	26.64	0.06	28.24	29.79
RS +1k	8.08	0.27	0.74	0.00	30.11	1.55

(e) CLAD

	VC	WN	EN	TS	FD	RS
VC 0.1	0.06	0.03	0.04	0.00	0.00	0.08
WN 15dB	0.02	0.12	0.05	0.01	0.33	0.38
EN Wind	0.00	0.00	0.23	0.00	0.19	0.31
TS 0.9	0.00	0.00	0.01	0.03	0.10	0.31
FD .5/ H	0.00	0.00	0.02	0.00	0.81	1.49
RS +1k	0.02	0.10	0.84	1.11	0.31	1.63

FARs. Furthermore, combining different manipulation techniques results in stronger attack performance for baselines like RawNet2 and Res-TSSDNet.

C. CLAD Performance (RQ2)

In this section, we evaluate CLAD's effectiveness against manipulation attacks and answer the second research question. Tab. I presents the results. We observe a slight increase in FAR from 0.83% to 1.11% for CLAD in the absence of manipulation. Notably, 0.83% represents the best result achieved by AASIST, with the authors reporting an average training result of 1.13% [15]. This increase in FAR is an acceptable trade-off, considering the overall robustness of CLAD. In the presence of white noise injection, the proposed CLAD model outperforms Res-TSSDNet, achieving an FAR of 0.12%, whereas Res-TSSDNet records a substantially higher FAR of 51.28%. Similarly, CLAD exhibits significant improvements over RawNet2 and AASIST when subjected to volume control and fading. Even for combinations of manipulations, CLAD maintains a lower FAR compared to using only a single manipulation, as depicted in Tab. II. In conclusion, CLAD demonstrates robustness against manipulation attacks by maintaining the highest FAR of 1.63% and the lowest F1 score of 92.82% among all tested manipulation attack scenarios.

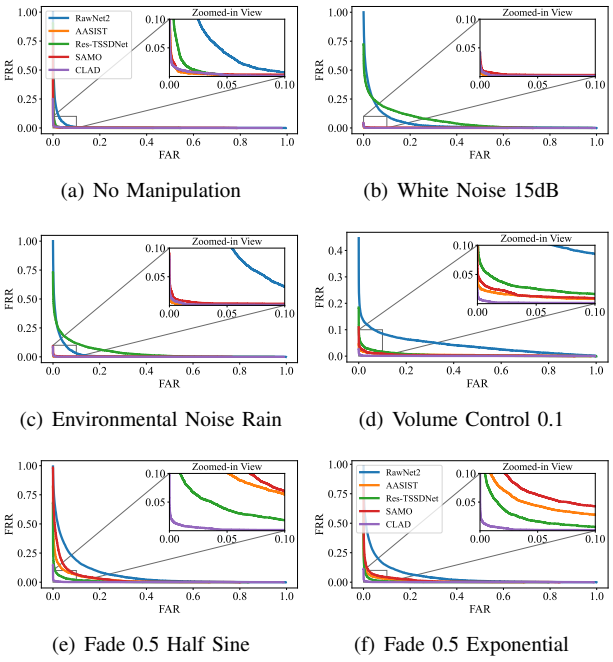
Additionally, we present the evaluation results using Detection Error Tradeoff (DET) curves, which offer a comprehensive visualization of the tradeoff between the False Acceptance Rate (FAR) and False Rejection Rate (FRR). Fig. 6 depicts the DET curves for different models across four manipulation attack settings. Curves positioned in the lower left quadrant indicate better detection performance. It can be observed that CLAD consistently performs excellently, while the baseline models exhibit varying degrees of performance degradation under certain manipulations.

It should be noted that CLAD can be integrated with existing end-to-end detection models by taking them as an encoder. In this study, we selected the AASIST architecture as the encoder for CLAD. To assess the impact of the encoder architecture, we conducted performance evaluations of CLAD using RawNet2 and Res-TSSDNet as alternative encoders. SAMO is based on the same architecture as AASIST, so we do not compare it here. Representative results are illustrated in Fig. 7, and full results can be found in Tab. III. The results demonstrate that the encoder architecture do influence the performance of CLAD. Specifically, the model employing an AASIST encoder consistently outperforms the alternative encoders across most experimental conditions. This observation aligns with the baseline model performance reported in Tab.I, suggesting that a better encoder architecture could enhance the performance of CLAD. Moreover, observing the results presented in Tab. III, it is noteworthy that the performance of CLAD with RawNet2 encoder still outperforms the performance of the original model under most manipulation attacks. We even trained a CLAD model with RawNet2 encoder with better performance on original data compared to the

TABLE III
THE FARS (%) OF MODEL WITH DIFFERENT ENCODER ARCHITECTURES UNDER ALL MANIPULATIONS

Encoder	None	Volume Control		White Noise			Environmental Noise							Time Stretch			
Architecture		0.5	0.1	15dB	20dB	25dB	WD	FS	BR	CO	RA	CT	SN	1.1	1.05	0.95	0.9
RawNet2	3.37	1.44	0.36	4.06	3.61	3.24	2.28	6.13	3.63	2.24	3.86	8.23	2.85	0.29	0.37	0.32	0.23
Res-TSSDNet	3.00	1.25	0.53	0.03	0.15	0.44	0.20	0.70	3.23	0.29	0.19	0.68	1.97	0.19	0.28	0.20	0.13
AASIST	1.11	0.70	0.06	0.12	0.52	0.78	1.39	1.17	0.68	0.20	0.23	1.15	1.01	0.08	0.12	0.06	0.03

Encoder	Add Echoes			Time Shift			Fade							Resample			
Architecture	1k/ .2	1k/ .5	2k/ .5	1.6k	16k	32k	.5/ L	.3/ L	.1/ L	.5/ E	.5/ Q	.5/ H	.5/ Lo	-1k	-0.5k	+0.5k	+1k
RawNet2	2.19	1.17	1.30	2.50	3.01	3.00	1.31	2.10	2.94	2.05	1.74	7.08	1.95	3.98	3.28	2.81	2.54
Res-TSSDNet	0.71	0.14	0.16	2.41	2.43	2.48	0.70	1.08	1.61	0.52	0.99	0.86	1.30	1.78	2.19	3.43	3.87
AASIST	0.21	0.03	0.07	0.85	0.93	0.89	0.64	0.95	1.06	0.50	0.85	0.81	0.93	0.65	0.81	1.38	1.63



(a) No Manipulation (b) White Noise 15dB

(c) Environmental Noise Rain (d) Volume Control 0.1

(e) Fade 0.5 Half Sine (f) Fade 0.5 Exponential

Fig. 6. DET curve of detection models under different manipulations.

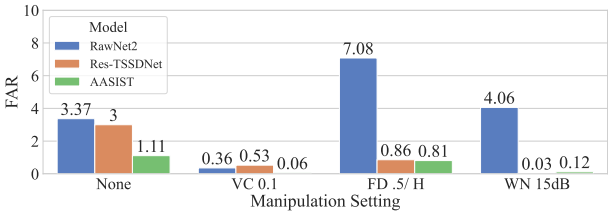


Fig. 7. The FARs (%) of models with differnet encoder architectures under different manipulation attacks

model released by the authors. Results here underscore the CLAD's effectiveness in improving the robustness of existing deepfake detection models. Fig. 8 presents an illustrative audio deepfake case alongside predictions generated by various detection methods under different manipulation settings for better comprehension.

D. Ablation Study (RQ3)

In order to systematically assess the key components of our approach, we conducted ablation experiments from two

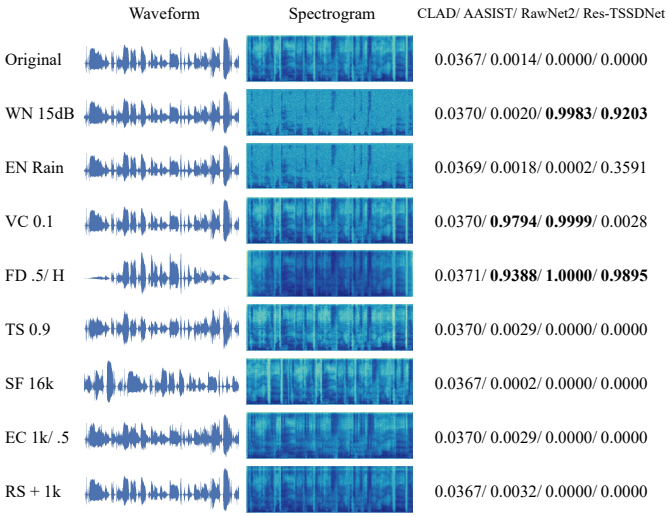


Fig. 8. Audio Deepfake examples under various manipulations, along with predicted probabilities of being real. Higher predictions indicate the model mistakenly identifies the audio as real.

perspectives and provide a comprehensive analysis in the following sections.

Model Variants. We consider four model variants to evaluate the components of CLAD, which are summarized in Tab. V. The ablation study results are presented in Tab. IV.

1) Vanilla. This is an AASIST model trained without contrastive learning and length loss. Compared with the baseline, it employs all manipulations as data augmentation.
2) CL. This represents our model trained without length loss.
3) LL. Refers to the Vanilla model trained with the inclusion of length loss as an additional loss function.
4) CLAD. Denotes our proposed method, which utilizes both contrastive learning and length loss.

Effectiveness of Contrastive Learning. Examining Tab. IV, it is evident that contrastive learning proves effective in enhancing the model's robustness against manipulation attacks. For models trained without contrastive learning, i.e., Vanilla and LL, we observe FAR values of 11.47% and 13.87% under fading and volume control, respectively. Even though these models are trained with all the manipulations, conventional supervised learning struggles to handle such diverse manipulations effectively. In contrast, CL and CLAD yield more consistent results across all manipulations. It is worth noting that the FAR of LL for the original dataset

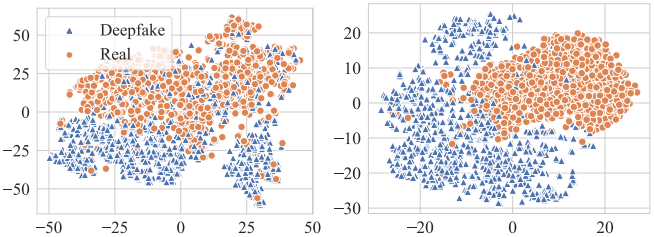
TABLE IV
THE FARS (%) OF MODEL VARIANTS IN ABLATION STUDY UNDER ALL MANIPULATIONS.

Models	None	Volume Control		White Noise			Environmental Noise							Time Stretch			
		0.5	0.1	15dB	20dB	25dB	WD	FS	BR	CO	RA	CT	SN	1.1	1.05	0.95	0.9
Vanilla	3.02	4.80	0.19	0.12	0.71	1.67	3.19	4.12	2.87	0.99	0.69	4.96	2.98	0.39	0.53	0.31	0.06
CL	4.36	1.09	0.03	2.52	4.20	4.61	4.20	4.23	4.10	1.50	3.01	5.89	3.43	0.79	0.90	0.88	0.67
LL	2.00	13.87	2.24	6.63	4.17	2.72	4.26	5.84	4.17	13.70	4.76	7.29	4.94	9.64	7.04	7.34	9.72
CLAD	1.11	0.70	0.06	0.12	0.52	0.78	1.39	1.17	0.68	0.20	0.23	1.15	1.01	0.08	0.12	0.06	0.03

Models	Add Echoes			Time Shift			Fade							Resample			
	1k/ .2	1k/ .5	2k/ .5	1.6k	16k	32k	.5/ L	.3/ L	.1/ L	.5/ E	.5/ Q	.5/ H	.5/ Lo	-1k	-0.5k	+0.5k	+1k
Vanilla	0.55	0.04	0.16	1.90	2.06	1.93	6.14	5.03	3.47	6.87	5.46	11.47	4.79	2.32	2.63	3.20	4.11
CL	1.85	0.76	1.02	3.73	3.79	3.87	1.82	3.33	4.08	1.25	2.79	1.95	3.19	3.44	3.97	4.94	5.88
LL	3.73	8.12	9.46	2.79	2.92	2.90	2.52	2.15	2.04	3.33	2.26	4.43	2.09	2.34	2.26	2.22	2.54
CLAD	0.21	0.03	0.07	0.85	0.93	0.89	0.64	0.95	1.06	0.50	0.85	0.81	0.93	0.65	0.81	1.38	1.63

TABLE V
THE MODEL VARIANTS IN ABLATION STUDY

Component	Vanilla	CL	LL	CLAD
Contrastive Learning	✗	✓	✗	✓
Length Loss	✗	✗	✓	✓



(a) w/o length loss (b) w/ length loss

Fig. 9. Visualization of the features extracted using the encoders trained without and with length loss.

stands at 4.36%, which is less favorable compared to other models. This discrepancy is attributed to the unsupervised nature of contrastive learning, which results in the model being insufficiently trained with label information.

Effectiveness of Length Loss. Length loss encourages the model to learn the clustering of real audio, thus making downstream training easier. As demonstrated in Tab. IV, models trained with length loss outperform those without it on the original data. LL achieves an FAR of 2.00%, while Vanilla achieves 3.02%. Notably, CLAD shows a substantial improvement compared to CL, with CLAD achieving a FAR of 1.11% compared to CL's 4.36%. Similar improvement can also be observed under various manipulations. Such enhancements can be attributed to the design of the length loss, which utilizes length of feature vectors and complements contrastive learning. To provide further insight into the impact of length loss, we visualize the feature vectors extracted by the encoder, both trained with and without length loss, in Fig. 9. The visualization illustrates that the encoder trained with length loss produces feature vectors that are more closely aggregated. This, in turn, enhances detection performance.

TABLE VI
THE FARS (%) OF MODELS TRAINED WITH PARTIAL MANIPULATIONS UNDER DIFFERENT MANIPULATION ATTACKS INCLUDING UNKNOWN MANIPULATION.

	VC	WN	EN	TS	EC	SF	FD	RS
None	1.49	2.61	1.96	2.03	1.76	1.26	2.24	2.11
VC 0.1	7.39	2.77	2.05	1.70	2.90	1.69	0.49	4.25
WN 15dB	0.50	1.12	0.16	0.52	0.40	0.19	0.37	0.36
EN Rain	0.80	1.65	0.59	0.93	0.83	0.58	0.84	0.90
TS 0.9	0.25	0.52	0.28	0.03	0.34	0.03	0.21	0.22
EC 1k/ .2	0.67	1.19	0.73	0.67	0.28	0.30	0.96	0.50
SF 16k	1.21	2.20	1.55	1.74	1.45	0.87	1.97	1.81
FD .5/ H	2.81	3.22	2.30	3.32	3.60	2.34	6.19	2.00
RS +1k	1.77	3.52	2.75	2.31	2.32	1.72	2.90	3.04

E. Unknown Manipulation Study (RQ4)

In this section, we assess the detection capability of CLAD for unknown manipulation attack methods, considering that the manipulation methods discussed in our paper may not encompass all methods used by attackers. Of the eight manipulation methods examined in our study, we trained eight CLAD models, each with one manipulation method removed. Subsequently, we assessed the performance of these models when confronted with unknown manipulation attacks. The representative results are presented in Tab. VI, each column represents a model trained without a specific manipulation.

We observed that volume control and fading are particularly strong attacks, as models trained without these manipulations performed not well, resulting in a FAR of 7.19% and 6.19%, respectively. However, for other manipulations, models trained without them still achieved good performance under unknown manipulation attacks. For instance, the model trained without time stretch achieved a low FAR of 0.03% under time stretch manipulation. Similar results were observed for time shift and echoes adding. Moreover, compared to baseline models, CLAD demonstrated improved performance against unknown manipulations in most scenarios. For example, although both trained without fading manipulation, our design enabled CLAD to reduce its FAR from 31.22% to 6.19% compared with baseline AASIST model. Therefore, we conclude that while strong manipulations like volume control and fading may lead to suboptimal results, CLAD is expected to outperform baseline models and achieve good performance against most unknown manipulation attacks.

F. Key Findings

- **Manipulation attacks pose a severe threat to deepfake detection models.** We found that all the baseline models we evaluated are vulnerable to particular manipulations. Among manipulations, volume control and fading demonstrate the most favorable attack performance. While volume control can be mitigated through straightforward techniques like input normalization, fading manipulations pose a significant challenge to current detection methods. Fading with a half sine fade shape demonstrates a notably high 25.36% average FAR across the four baseline models.
- **CLAD enhances model resilience against manipulations.** We successfully trained a robust audio deepfake detection model with AASIST encoder architecture, achieving a low 1.63% FAR under all manipulations. Furthermore, our experiments confirm that CLAD can serve as a plug-and-play module to enhance the robustness of existing deepfake detection models. Notably, CLAD trained with the RawNet2 encoder outperforms the model released by the authors, even in unmanipulated data.
- **Contrastive Learning and Length Loss are essential components of CLAD.** We found that contrastive learning trained model perform more consisitently under different manipulation attacks, indicating its improved robustness. Length loss is also an essential component of CLAD, complementing contrastive learning and further improving its performance. Moreover, our evaluation results suggest that intuitive data augmentation is not the optimal solution for countering manipulation attacks.
- **CLAD shows promise for superior performance against unknown manipulation attacks.** Our study reveals that CLAD demonstrates promising performance in the presence of unknown manipulation attacks. Even against strong manipulations like volume control and fading, it outperforms baseline models.

VI. Related Work

A. Contrastive Learning

Contrastive learning is a popular form of self-supervised learning [39], [40] that encourages augmentations (views) of the same input to have more similar representations than augmentations of different inputs. Earlier studies [23], [26] illustrates the effectiveness of contrast learning by showing that contrastive learning can even yield better results than supervised learning in some cases. Although contrastive learning has been a great success in the field of computer vision [41]–[43], it has not received enough attention in the audio domain. Saeed et al. [24] proposed COLA, a self-supervised pre-training approach for learning a general-purpose representation of audio. However, COLA does not discuss the downstream tasks of audio deepfake detection. Guan et al. [25] introduced contrastive learning into the anomalous sound detection and showed its effectiveness. In conclusion, existing work on contrast learning is not sufficient for the audio domain, and as far as we know, our work is the first to apply contrast learning to the field of audio deepfake detection.

B. Audio Deepfake Detection

The rise of audio deepfakes has posed significant security threats, prompting a surge of interest in developing robust detection methods [30], [44], [45]. Earlier approaches [11], [12], [46] were mainly based on handcrafted features and machine learning-based classifiers. In 2021, Tak et al. [13] proposed a milestone model which is based on an end to end speaker verification model, RawNet2 [47]. Remarkably, RawNet2 demonstrated state-of-the-art (SOTA) performance without the need for hand-crafted features. Its pre-trained models quickly gained popularity within the field. After that, Jung et al. [15] proposed a heterogeneous stacking graph attention layer that models artefacts spanning heterogeneous temporal and spectral intervals, and achieved SOTA performance on the ASVspoof 2019 dataset. Blue et al. [48] leverage fluid dynamics to estimate the human vocal tract arrangement during speech production and use it to identify deepfakes. This approach, however, needs accurate phoneme timestamps, which are seldom accessible in practical scenarios.

Recently, instead of solely focusing on benchmark datasets performance, researchers have begun considering broader aspects of audio deepfake detection. Recognizing the limitations of existing models on non-English data, Ba et al. [31] proposed domain adaptation strategies for cross-lingual detection. Zhang et al. [49], [50] investigated continual learning in audio deepfake detection, which is an effective way to help detection models adapt to new attack types. Zhang et al. [33] and Singh et al. [51] investigated the challenges in detecting audio deepfakes in compressed form.

However, most of these approaches do not consider the scenarios where the attacker manipulate the speech. A notable exception is DeepSonar [21], which evaluates the robustness of the model and performs a small-scale experiment of manipulation attacks. In this work, we conduct a larger scale experiment on ASVSpoof 2019 evaluation set (containing 71,237 samples) with more types of manipulation attacks.

VII. Discussion

Intuitive Defense. An intuitive defense against manipulation attacks is denoising, as noise injection is one of manipulations. However, our preliminary experiment with Wiener filtering does not yield satisfactory outcomes. While modern speech enhancement methods have potentials, we are concerned that enhancements artefact might trigger false alarms in the detection model. We leave further investigation into this matter to future research.

Diverse Datasets. In this study, we opt to evaluate the baselines using the parameters provided by the authors to make the results more convincing. However, this choice limited our dataset to the ASVspoof 2019 dataset, as the pretrained models did not perform well on other publicly available datasets. With the advancement of the field, we anticipate that future models will exhibit improved generalization and adaptable to a wider range of datasets, enabling evaluations on a more diverse dataset in the future.

VIII. Conclusion

In this paper, we expose the threat of manipulation attacks against audio deepfake detection systems. It does not require prior knowledge of the system, making it simple for deepfake creators to degrade the detection system performance. To address this issue, we conduct a large-scale evaluation of widely adopted models on the ASVspoof dataset under manipulation attacks and find that existing SOTA models experience severe performance degradation when faced with manipulation attacks. To mitigate the impact of manipulation attacks, we propose contrastive learning-based detection approach and design length loss to train a model to produce robust representations against manipulations. Experimental results show that CLAD achieves significantly better performance than the widely adopted models under manipulation attacks.

References

[1] Y. Wang, R. J. Skerry-Ryan *et al.*, "Tacotron: Towards End-to-End Speech Synthesis," in *INTERSPEECH 2017*, F. Lacerda, Ed. ISCA, 2017, pp. 4006–4010.

[2] R. Huang, Z. Zhao *et al.*, "ProDiff: Progressive Fast Diffusion Model for High-Quality Text-to-Speech," in *Proceedings of the 30th ACM International Conference on Multimedia (MM)*. Lisboa Portugal: ACM, October 2022, pp. 2595–2605.

[3] Y. A. Li, C. Han *et al.*, "StyleTTS 2: Towards Human-Level Text-to-Speech through Style Diffusion and Adversarial Training with Large Speech Language Models," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems (NeurIPS) 2023*, 2023.

[4] Y. A. Li, C. Han, and N. Mesgarani, "Styletts-VC: One-Shot Voice Conversion by Knowledge Transfer From Style-Based TTS Models," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 920–927.

[5] H. Tang, X. Zhang *et al.*, "Avqvc: One-Shot Voice Conversion By Vector Quantization With Applying Contrastive Learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 4613–4617.

[6] J. Li, W. Tu, and L. Xiao, "Freevc: Towards High-Quality Text-Free One-Shot Voice Conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[7] S. Khatsenkova, "Audio deepfake scams: Criminals are using ai to sound like family and people are falling for it," 2023. [Online]. Available: https://www.euronews.com/embed/2231732

[8] E. Wenger, M. Bronckers *et al.*, ""Hello, It's Me": Deep Learning-Based Speech Synthesis Attacks in the Real World," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, New York, NY, USA, 2021, pp. 235–251.

[9] Z. Khanjani, G. Watson, and V. P. Janeja, "Audio deepfakes: A survey," *Frontiers Big Data*, vol. 5, 2022.

[10] E. A. AlBadawy, S. Lyu, and H. Farid, "Detecting AI-Synthesized Speech Using Bispectral Analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[11] G. Lavrentyeva, S. Novoselov *et al.*, "STC Antispoofing Systems for the ASVspoof2019 Challenge," in *INTERSPEECH 2019*. ISCA, September 2019, pp. 1033–1037.

[12] M. Todisco, H. Delgado, and N. Evans, "Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, Sep. 2017.

[13] H. Tak, J. Patino *et al.*, "End-to-End anti-spoofing with RawNet2," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6369–6373.

[14] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards End-to-End Synthetic Speech Detection," *IEEE Signal Processing Letters*, vol. 28, pp. 1265–1269, 2021.

[15] J.-w. Jung, H.-S. Heo *et al.*, "AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6367–6371.

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

[17] J. Yamagishi, X. Wang *et al.*, "ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection," in *Proc. ASVspoof Challenge workshop*, 2021, pp. 47–54.

[18] Y. Zhang, Z. Jiang *et al.*, "Black-Box Attacks on Spoofing Countermeasures Using Transferability of Adversarial Examples," in *INTERSPEECH 2020*. ISCA, October 2020, pp. 4238–4242.

[19] S. Liu, H. Wu *et al.*, "Adversarial Attacks on Spoofing Countermeasures of Automatic Speaker Verification," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 312–319.

[20] M. Panariello, W. Ge *et al.*, "Malafide: a novel adversarial convolutive noise attack against deepfake and spoofing detection systems," in *INTERSPEECH 2023*. ISCA, Aug. 2023, pp. 2868–2872.

[21] R. Wang, F. Juefei-Xu *et al.*, "DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices," in *Proceedings of the 28th ACM International Conference on Multimedia (MM)*. Seattle WA USA: ACM, October 2020, pp. 1207–1216.

[22] H. Hojjati and N. Armanfard, "Self-Supervised Acoustic Anomaly Detection Via Contrastive Learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 3253–3257.

[23] K. He, H. Fan *et al.*, "Momentum Contrast for Unsupervised Visual Representation Learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 9726–9735.

[24] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive Learning of General-Purpose Audio Representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Toronto, ON, Canada: IEEE, June 2021, pp. 3875–3879.

[25] J. Guan, F. Xiao *et al.*, "Anomalous Sound Detection Using Audio Representation with Machine ID Based Contrastive Learning Pretraining," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[26] T. Chen, S. Kornblith *et al.*, "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR, November 2020, pp. 1597–1607.

[27] Y. Zhang, F. Jiang, and Z. Duan, "One-Class Learning Towards Synthetic Voice Spoofing Detection," *IEEE Signal Process. Lett.*, vol. 28, pp. 937–941, 2021.

[28] S. Ding, Y. Zhang, and Z. Duan, "SAMO: Speaker Attractor Multi-Center One-Class Learning For Voice Anti-Spoofing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Jun. 2023, pp. 1–5.

[29] J.-w. Jung, H. Tak *et al.*, "SASV 2022: The First Spoofing-Aware Speaker Verification Challenge," in *INTERSPEECH 2022*. ISCA, Sep. 2022, pp. 2893–2897.

[30] J. Yi, R. Fu *et al.*, "ADD 2022: the first Audio Deep Synthesis Detection Challenge," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9216–9220.

[31] Z. Ba, Q. Wen *et al.*, "Transferring Audio Deepfake Detection Capability across Languages," in *Proceedings of the ACM Web Conference (WWW) 2023*. Austin TX USA: ACM, Apr. 2023, pp. 2033–2044.

[32] N. M. Müller, P. Czempin *et al.*, "Does Audio Deepfake Detection Generalize?" in *INTERSPEECH 2022*. ISCA, 2022, pp. 2783–2787.

[33] J. Zhang, X. Yi, and X. Zhao, "A Compressed Synthetic Speech Detection Method with Compression Feature Embedding," in *INTERSPEECH 2023*. ISCA, Aug. 2023, pp. 5376–5380.

[34] Y. Wen, Z. Lei *et al.*, "Multi-Path GMM-MobileNet Based on Attack Algorithms and Codecs for Synthetic Speech and Deepfake Detection," in *INTERSPEECH 2022*. ISCA, Sep. 2022, pp. 4795–4799.

[35] X. Wang and J. Yamagishi, "Investigating Self-Supervised Front Ends for Speech Spoofing Countermeasures," in *The Speaker and Language Recognition Workshop (Odyssey 2022)*. ISCA, June 2022, pp. 100–106.

[36] E. Conti, D. Salvi *et al.*, "Deepfake Speech Detection Through Emotion Recognition: A Semantic Approach," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8962–8966.

[37] T.-P. Doan, L. Nguyen-Vu *et al.*, "BTS-E: Audio Deepfake Detection Using Breathing-Talking-Silence Encoder," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Jun. 2023, pp. 1–5.

[38] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia (MM)*. ACM Press, 2015, pp. 1015–1018.

[39] X. Liu, F. Zhang *et al.*, "Self-Supervised Learning: Generative or Contrastive," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[40] L. Jing and Y. Tian, "Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[41] M. Kang and J. Park, "ContraGAN: Contrastive Learning for Conditional Image Generation," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato *et al.*, Eds., 2020.

[42] S. Park, J. Lee *et al.*, "Fair Contrastive Learning for Facial Attribute Classification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 10 379–10 388.

[43] P. Wang, K. Han *et al.*, "Contrastive Learning Based Hybrid Networks for Long-Tailed Image Classification," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 943–952.

[44] J. Yi, J. Tao *et al.*, "ADD 2023: the Second Audio Deepfake Detection Challenge," in *Proceedings of the Workshop on Deepfake Audio Detection and Analysis co-located with 32th International Joint Conference on Artificial Intelligence (IJCAI 2023), Macao, China, August 19, 2023*, vol. 3597, 2023, pp. 125–130.

[45] J. Frank and L. Schönherr, "WaveFake: A Data Set to Facilitate Audio Deepfake Detection," in *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks 1*, 2021.

[46] M. E. Ahmed, I.-Y. Kwak *et al.*, "Void: A fast and light voice liveness detection system," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2685–2702.

[47] J.-w. Jung, S.-b. Kim *et al.*, "Improved RawNet with Feature Map Scaling for Text-Independent Speaker Verification Using Raw Waveforms," in *INTERSPEECH 2020*. ISCA, Oct. 2020, pp. 1496–1500.

[48] L. Blue, K. Warren *et al.*, "Who Are You (I Really Wanna Know)? Detecting Audio DeepFakes Through Vocal Tract Reconstruction," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

[49] X. Zhang, J. Yi *et al.*, "Do You Remember? Overcoming Catastrophic Forgetting for Fake Audio Detection," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*. PMLR, Jul. 2023, pp. 41 819–41 831, iSSN: 2640-3498.

[50] ——, "What to Remember: Self-Adaptive Continual Learning for Audio Deepfake Detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, pp. 19 569–19 577, Mar. 2024.

[51] A. K. Singh Yadav, Z. Xiang *et al.*, "ASSD: Synthetic Speech Detection in the AAC Compressed Domain," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

Haolin Wu received the B.E. degree in Information Security from from Wuhan University, Hebei, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, Hubei, China. His research interest lies in the area of multimedia and machine learning security and privacy.

Jing Chen received the Ph.D. degree in computer science from Huazhong University of Science and Technology, Wuhan, China. He is the deputy Dean of the School of Cyber Science and Engineering at Wuhan University. His research interests are in the areas of network security, cloud security, and mobile security. He has published more than 100 research papers in many international journals and conferences, including USENIX Security, ACM CCS, INFOCOM, IEEE TDSC, IEEE TIFS, IEEE TMC, IEEE TC, IEEE TPDS, IEEE TSC, etc. He acts as a reviewer for many conferences and journals, such as IEEE INFOCOM, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing and IEEE/ACM Transactions on Networking.

Ruiying Du received the B.S., M.S., and Ph.D. degrees in computer science in 1987, 1994 and 2008, from Wuhan University, Wuhan, China. She is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. Her research interests include network security, wireless network, cloud computing and mobile computing. She has published more than 80 research papers in many international journals and conferences, such as TPDS, USENIX Security, CCS, INFOCOM, SECON, TrustCom, NSS.

Cong Wu is currently a Research Fellow at Cyber Security Lab, Nanyang Technological University, Singapore. He received Ph.D. degree at School of Cyber Science and Engineering, Wuhan University in 2022. His research interests include AI system security and Web3 security. His leading research outcomes have appeared in USENIX Security, ACM CCS, IEEE TDSC, TMC.

Kun He received his Ph.D. from Wuhan University, Wuhan, China. He is currently an associate professor with Wuhan University. His research interests include cryptography and data security. He has published more than 30 research papers in various journals and conferences, such as TIFS, TDSC, TMC, USENIX Security, CCS, and INFOCOM.

Xingcan Shang received the B.S. degree in computer science and technology from Central South University, Hunan, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China. His research interest include artificial intelligence security.

Hao Ren is currently a research associate professor at the Sichuan University. He was a research fellow at Nanyang Technological University from Jul. 2022 to Feb. 2024, at The Hong Kong Polytechnic University from Aug. 2021 to Jun. 2022. He received his Ph.D. degree in Dec. 2020 from the University of Electronic Science and Technology of China. He was a visiting Ph.D. student at the University of Waterloo from Dec. 2018 to Jan. 2020. His research outcomes appeared in major conferences and journals, including WWW, ACM ASIACCS, ACSAC, IEEE TCC, and IEEE Network. He won the Best Paper Award from IEEE BigDataSecurity 2023. His research interests include data security and privacy, applied cryptography and privacy-preserving machine learning.

Guowen Xu is currently a Postdoc at City University of Hong Kong. He obtained his Ph.D. degree from University of Electronic Science and Technology of China in 2020. His research focuses on applied cryptography and privacy-preserving deep learning, resulting in over 80 publications in reputable security conferences/journals including IEEE S&P, ACM CCS, ICML, NeurIPS, ICLR, CVPR, TDSC and TIFS. He was recognized as one of Stanford World's Top 2% Scientists in 2023. Currently, He serves as the Associate Editor for IEEE TIFS, IEEE TNSM, Lead Guest Editor of ACM TAAS, and hold the title of Distinguished Reviewer for ACM TWEB. Moreover, he has had the privilege of participating as a TPC member for esteemed conferences such as ICML (Area Chiar 2024), ACSAC 2024, NeurIPS 2023, CVPR 2023, WWW 2022, AAAI 2022-2023, KSEM 2022, and ICC 2024.

APPENDIX

MANIPULATION DETAILS

In the following section, we provide a description of the manipulation process and define the parameters used. For more detailed information, readers are encouraged to refer to our implementation(https://github.com/CLAD23/CLAD).

In the case of noise injection, we generate noise and adjust its amplitude to achieve a desired signal-to-noise ratio (SNR). A higher SNR corresponds to a lower level of added noise. In the context of environmental noise injection, we empirically identified the most effective environmental noise source from the ESC-50 dataset.

Volume control involves multiplying the amplitude of the audio waveform by a specified factor. The factor dictates the extent and direction of volume adjustment, whether it entails an increase or decrease, and the magnitude of the change.

Fading involves applying a specific fade shape to attenuate portions of audio at both the beginning and end. The fade shape defines a mask which is applied as a multiplier to the audio's amplitude. The fade ratio determines the extent of audio fading. For instance, a ratio of 0.5 implies that half of the audio at the beginning and half at the end will be faded. The maximum selectable ratio is also 0.5. Detailed information regarding the fade shape definition can be found in the PyTorch implementation.

Time stretching is performed using the official PyTorch implementation. This process alters the audio signal's duration while preserving its pitch. The FFT length is utilized in the FFT calculations involved. The time stretching factor specifies the ratio between the length of the stretched audio and the original audio. A factor less than 1 speeds up the audio.

Resampling is conducted using the official PyTorch implementation. This operation adjusts the audio signal's sample rate to a specified target rate. The target resampling rates defines the sample rate after resampling. When resampled audio is played at the original sample rate, both pitch and duration are affected.

Time shifting involves shifting the audio signal in the time domain by moving all samples forward or backward. The shift length specifies both the direction of audio signal movement and the number of samples shifted.

Echoes adding creates a delayed and attenuated copy of the original audio and adds them together. The delay parameter determines the echo's delay in samples, and the attenuation factor represents the number by which the echo's amplitude is multiplied.

All manipulations were implemented to process the raw waveform and output the raw waveform.