

# Evaluation Framework for Multi-Channel Spoofing Detection Through Redesign of the ReMASC Corpus

Takuo Yamaguchi<sup>1</sup>, Sayaka Shiota<sup>1</sup>, and Naohiro Tawara<sup>2</sup>

<sup>1</sup> Tokyo Metropolitan University, Tokyo, Japan  
yamaguchi-takuo@ed.tmu.ac.jp, sayaka@tmu.ac.jp

<sup>2</sup> NTT, Inc., Kyoto, Japan  
naohiro.tawara@ieee.org

**Abstract.** In detecting replay spoofing attacks against voice-controlled systems such as smart speakers, the utilization of multi-channel information is crucial. ReMASC is the only large-scale multi-channel replay attack corpus created with multiple recording devices featuring different microphone array configurations, enabling comparison of detection performance across different device configurations. However, fair comparison is not possible because the composition ratios of recording conditions are inconsistent across recording devices. Additionally, since conditions other than speakers are known in the existing subsets, the independent analysis of each recording condition’s impact on performance is not feasible. In this study, we redesign ReMASC by introducing data cleaning to unify the composition ratios of recording conditions across recording devices and a subset splitting method that allows arbitrary recording conditions to be controlled as known or unknown. Furthermore, we conducted experiments based on the redesigned data splits and demonstrated that it is possible to quantitatively evaluate the impact of individual recording conditions and device configuration differences on detection accuracy.

**Keywords:** Spoofing detection · ReMASC corpus · Multi-channel audio · Dataset · Evaluation framework

## 1 Introduction

In recent years, voice-controlled systems represented by smart speakers have become increasingly widespread. From a personal information protection perspective, these voice-controlled systems often incorporate speaker verification for user authentication. On the other hand, there are concerns about replay spoofing attacks, where attackers present illegally recorded utterances of legitimate users to the system for speaker verification. However, rejecting replay spoofing through speaker verification alone is difficult, posing a serious challenge. Therefore, spoofing detection, which identifies whether input speech is bonafide or spoofed, is essential for voice-controlled systems.

By using signals from multiple microphones, it is possible to capture subtle fluctuations in sound-source position that are unique to bonafide speech and the presence or absence of noise from playback devices [12], enabling higher detection performance than conventional single-channel methods.

On the other hand, research related to spoofing detection still predominantly targets single-channel audio, and datasets recorded in multi-channel are limited. Among them, ReMASC (Realistic Replay Attack Microphone Array Speech Corpus) [2] is recognized as a representative corpus for large-scale multi-channel spoofing attack detection and is widely used [3, 6, 8, 9, 11, 13, 14]. ReMASC includes various attack conditions with four types of recording devices and multiple recording environments. However, ReMASC has the following problems:

1. Recording devices have inconsistent composition ratios across recording conditions, making it difficult to disentangle device characteristics from environmental factors and to make fair comparisons.
2. The spatial relationship between microphones and sound sources varies across training and evaluation data, making it difficult to assess the impact of each condition on system performance.
3. The absence of validation data makes it difficult to suppress overfitting and adjust parameters.

Additionally, since fair evaluation across recording devices is not possible and the impact of each recording condition cannot be analyzed individually, the reliability as an evaluation framework is limited.

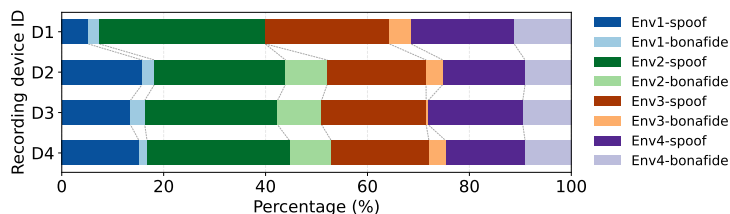
Therefore, in this study, we propose a new evaluation framework for spoofing attack detection using multi-channel information by redesigning ReMASC. Specifically, by providing data splits with unified conditions across recording devices through the redesign of ReMASC<sup>3</sup>, fair and reproducible experimental design and model evaluation under various recording conditions become possible, contributing as a research foundation for related fields. By evaluating existing methods using the redesigned subsets, we revealed that the performance differences across recording devices that have been reported are attributable to characteristics inherent to the recording devices rather than condition bias.

The structure of this paper is as follows. First, Section 2 analyzes the composition of the ReMASC dataset, and Section 3 details the data cleaning and new subset splitting methods. Section 4 discusses the impact of recording devices and environmental conditions on detection performance based on evaluation experiment results using the redesigned dataset, and finally Section 5 summarizes this research.

## 2 Composition of ReMASC

This section analyzes the data composition of ReMASC and the label composition ratios across recording devices, and describes the problems with existing subset splits.

<sup>3</sup> [https://github.com/shiotalab-tmu/remasc\\_curated\\_splits](https://github.com/shiotalab-tmu/remasc_curated_splits)



**Fig. 1.** Composition ratio of bonafide and spoofed speech by recording environment (Env) for recording devices D1–D4

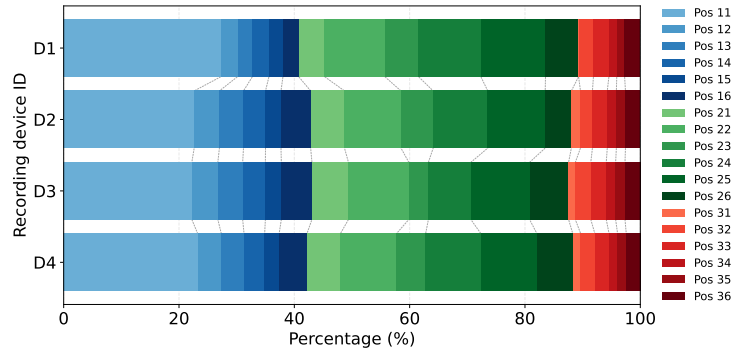
## 2.1 Recording Conditions of ReMASC

ReMASC is an audio corpus designed to improve the accuracy of spoofing detection for voice-controlled devices. To simulate the diverse environments in which voice-controlled devices operate, recordings were made under various conditions with different recording environments, playback devices, replay source recorders, speakers, position information, and recording devices for both bonafide and spoofed speech. The dataset contains 51 speakers: 50 humans and Text-to-speech (TTS). Four recording environments were employed: Env1 (outdoor), Env2 (quiet room), Env3 (room with background music), and Env4 (inside a car). Position information indicates the relative position between the recording device and the sound source with labeling schemes varying by environment. In Env1, near and far distance labels are assigned only to spoofed speech. In Env2, labels indicating the absolute position of the recording device within the room and the relative position of the sound source with respect to that recording device are assigned to all audio. In Env3, where both recording device and sound source positions are fixed, no positional labels are assigned. In Env4, labels indicating vehicle motion and the speaker’s position are assigned to bonafide speech, while sound source position labels are assigned to spoofed speech. Three types of replay source recorders, including TTS, were used. Four playback devices were used with labels assigned only to spoofed speech. Recording devices consist of four types of microphone arrays: D1 (2 channels), D2 (4 channels), D3 (6 channels), and D4 (7 channels).

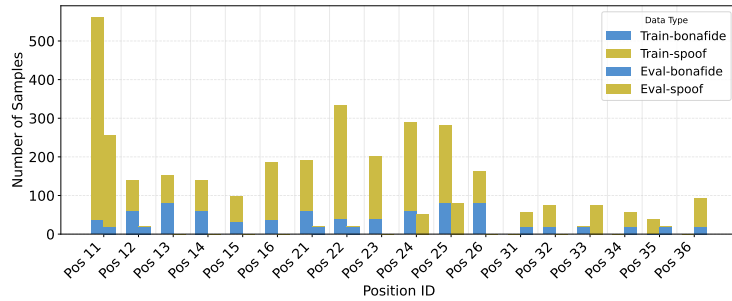
## 2.2 Problems with ReMASC

Although ReMASC comprises audio recordings under diverse conditions, it has aspects that render it unsuitable as an evaluation framework for detecting replayed spoofing attacks. Although multiple synthesizers are used for generation, containing elements such as multiple genders and dialects, the same speaker label is assigned, which may affect model performance analysis.

Next, there is the issue that composition ratios of recording conditions differ across recording devices. Figure 1 shows the composition ratio of recording environments (Env) for each recording device, and the ratio of bonafide speech



**Fig. 2.** Position information labels in Env2 for recording devices (D1–D4)



**Fig. 3.** Number of data items by relative position labels between recording device and playback device included in training and evaluation data for Env2 recorded with D2

to spoofed speech (spoo) in each environment. As shown in Figure 1, the composition ratios of the recording environments differ across recording devices. Furthermore, even within the same recording environment, the ratio of bonafide to spoofed speech is inconsistent across recording devices. This causes problems in performance comparison across recording devices. The composition ratios of both recording environments and bonafide/spoofed speech differ across devices, making it impossible to separate device characteristics from differences in recording conditions. Furthermore, Fig. 2 shows the composition ratio of position information (Pos) in Env2 for each recording device. The figure shows that, even within the same environment, the composition of positional information varies across recording devices. This further prevents isolating device characteristics from differences in the composition of recording conditions.

Another problem is that the publicly available subsets do not allow independent analysis of the impact of various recording conditions on identification accuracy. The subsets are train and eval, with speaker labels mutually exclusive between them. However, conditions other than speakers—such as position information and recording environment—share values across both subsets. Figure 3

illustrates this issue using Env2 audio from recording device D2, showing the distribution of bonafide and spoofed (spoo) files by position label across train and eval. While Pos 11 contains both bonafide and spoo in both subsets (known), other positions exhibit inconsistent patterns: some appear in only one subset (unknown), while others show mixed patterns where only bonafide or spoo is known. Because known and unknown labels are mixed in recording conditions other than speakers, it is impossible to distinguish whether identification failure is due to unknown speakers or unknown conditions.

Finally, since the subsets consist only of train and eval, verification during training is difficult, hindering overfitting suppression and parameter optimization. Additionally, when using a subset of the training data for validation, the specific samples used are not documented, preventing reproducible and valid model evaluation.

### 3 Reconstruction of ReMASC

This section explains the reconstruction policy for ReMASC, which consists of two stages: data cleaning and splitting algorithms, designed to address the problems described in Section 2.2.

#### 3.1 Data Cleaning

To standardize the composition ratios of recording conditions across recording devices, cleaning processing is applied to the audio data in ReMASC. Each audio data is assigned six types of recording conditions: audio type (bonafide/spoo), recording environment, playback device, replay source recorder, speaker, and position information. Let the label sets for each recording condition be  $\mathcal{L}_1, \dots, \mathcal{L}_M$ , then all condition combinations are represented by the Cartesian product  $\mathcal{C} = \mathcal{L}_1 \times \dots \times \mathcal{L}_M$  ( $M = 6$ ). For fair comparison across recording devices, data must be consistent across all conditions. Therefore, TTS audio and D1 data were excluded, due to the issues in Section 2.2 and D1’s lack of Env2 bonafide data (Fig. 1), respectively.

After excluding TTS audio and D1 data, data cleaning was performed using the following procedure to further align the data volume for each condition combination. First, for each condition combination  $c \in \mathcal{C}$ , the number of data items  $n_{c,d}$  for recording device  $d$  was aggregated, and the minimum number  $k_c = \min(n_{c,d})$  was set as the adopted number of data items for that condition. Next, for all recording devices,  $k_c$  items of data belonging to combination  $c$  were randomly extracted. However, since condition combinations with extremely small amounts of data lack statistical reliability, combinations where  $k_c < \tau$  were excluded from the analysis (in this work,  $\tau = 10$ ). Finally, the cleaned dataset  $\mathcal{D}'$  consists of recording devices D2, D3, and D4, and as shown in Table 1, the number of data items and bonafide/spoo ratio across recording devices are equalized.

This processing enables evaluation of performance differences originating from recording devices, independent of data volume bias.

**Table 1.** Number of utterances by recording device before and after data cleaning

Recording Device	Original		After Cleaning	
	Bona	Spoof	Bona	Spoof
D1	1,473	6,873	0	0
D2	2,452	8,212	2,017	5,005
D3	2,159	7,941	2,017	5,005
D4	2,365	8,311	2,017	5,005

### 3.2 Subset Splitting Policy

In this section, splitting data into train, dev, and eval to create one set ( $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{eval}}$ ) is called a “data split.” The data splits proposed are of two types: (1) fully-closed split where all conditions are known in training, validation, and evaluation data, and (2) partially-open split where only specific conditions are unknown. In both data splits, adjustments are made so that the data volume ratio is as close as possible to  $|\mathcal{D}_{\text{train}}|:|\mathcal{D}_{\text{dev}}|:|\mathcal{D}_{\text{eval}}| = 3 : 1 : 1$  (target ratio) and the ratio of bonafide to spoof is equal across subsets.

Additionally, since some recording conditions exist only in spoof, there is a problem that bonafide data cannot be split when performing data splitting with those recording conditions as unknown conditions. Therefore, along with data cleaning, labels that exist only in spoof were also randomly assigned to bonafide in advance.

**Fully-closed Split.** In fully-closed split, data is split so that the same conditions appear in all subsets of training, validation, and evaluation. This allows model performance to be evaluated while excluding the impact of unknown conditions on identification performance, since the model does not encounter unknown conditions. Specifically, data sets matching condition combination  $c \in \mathcal{C}$  from  $\mathcal{D}'$  were randomly allocated to  $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{eval}}$  according to the split ratio of 3:1:1. This enables fair comparison across recording devices and evaluation of baseline performance excluding the influence of unknown conditions while matching the recording conditions included in each subset.

**Partially-open Split: Splitting Algorithm.** In partially-open split, data is split so that only specific recording conditions are unknown. In this case, the utterance count ratio and bonafide/spoof ratio between each subset may deviate from the target ratio depending on how the condition labels are split. Therefore, an algorithm is needed that searches for data splits with only specific recording conditions unknown while keeping the error from the target ratio as small as possible.

Let the set of labels for the recording condition to be made unknown be  $\mathcal{L}$ , and consider splitting this label set into three mutually exclusive subsets ( $L_{\text{train}}, L_{\text{dev}}, L_{\text{eval}}$ ). At this time, when the number of label types  $|\mathcal{L}|$  for the condition to be made unknown is large, the number of combination candidates becomes

enormous, or when  $|\mathcal{L}| = 2$ , it becomes impossible to realize a mutually exclusive split into three subsets. Therefore, in this study, we decided to separate the data splitting algorithm depending on the number of label types for the condition to be made unknown as follows: (1) When the number of types is large: Heuristic search method, (2) When the number of types is small: Complete enumeration combination method, (3) When the number of types is 2: Binary split method. Below, we describe the details of each splitting algorithm.

*When the Number of Label Types is Large.* When the number of label types  $|\mathcal{L}|$  is large, it is not practical to enumerate all label splits and select the optimal split. Therefore, we aim to obtain only a finite number of label splits by appropriately limiting the number of label combinations. We propose a heuristic search method that combines a label combination candidate generation algorithm to limit the number of combinations to be searched and a data split selection algorithm to retain appropriate data splits. The procedure for the candidate generation algorithm is shown in Algorithm 1, and the procedure for the data split selection algorithm is shown in Algorithm 2, respectively.

The label combination candidate generation algorithm (Algorithm 1) randomly splits the label set  $\mathcal{L}$  for the recording condition to be made unknown into three parts to generate three label set subsets ( $L_{\text{train}}, L_{\text{dev}}, L_{\text{eval}}$ ), and constructs the data subsets ( $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{eval}}$ ) corresponding to those label subsets. At this time, constraint condition  $\mathcal{K}$  is introduced to reject conditions where the data volume after label splitting is clearly insufficient, in order to prevent combinatorial explosion. Also, for the obtained data split, error  $e$  is calculated using the CalculateError function, which calculates the error considering the ratio of data volume per subset to the target ratio and the bonafide/spoof ratio. The label combination candidate combined with this error  $e$  becomes one element of the candidate set with error  $\mathcal{P}_{\text{cand}}$ . The search is repeated until the predetermined search time limit  $T_{\text{limit}}$  is reached. This algorithm allows preparation of a set of label split candidates.

The subsequent data split selection algorithm (Algorithm 2) selects appropriate data split candidates from  $\mathcal{P}_{\text{cand}}$  generated by Algorithm 1 according to the target. The procedure involves sequentially selecting appropriate candidates from  $\mathcal{P}_{\text{cand}}$  and adding them to the selected data split set  $\mathcal{S}_{\text{final}}$ . For this purpose, there is a function SelectCandidate that selects appropriate candidates. SelectCandidate performs the process of selecting candidates that satisfy the diversity criterion  $\mathcal{M}$  from the current unselected candidates  $\mathcal{P}_{\text{remain}}$ . Here, diversity criterion  $\mathcal{M}$  is a criterion that lowers the evaluation of candidates whose data volume ratio per subset or bonafide/spoof ratio deviates from the target ratio, or whose label composition is too similar compared to already selected combination candidates. This series of processes is repeated until the target number of data split sets is reached, as shown in Algorithm 2. This allows construction of subset groups corresponding to label splits with small errors from the target ratio and low mutual correlation.

*When the Number of Label Types is Small.* When the number of label types  $|\mathcal{L}|$  for the condition to be made unknown is small, it may not be possible to satisfy

---

**Algorithm 1** Heuristic Search Method Step 1: Label Combination Candidate Generation
 

---

**Require:** Cleaned data  $\mathcal{D}'$ , target label set  $\mathcal{L}$   
**Require:** Constraint conditions for splitting  $\mathcal{K}$   
**Require:** Search time limit  $T_{\text{limit}}$   
**Ensure:** Data split candidate set with error  $\mathcal{P}_{\text{cand}}$

- 1:  $\mathcal{P}_{\text{cand}} \leftarrow \emptyset$
- 2: **while** search time  $< T_{\text{limit}}$  **do**
- 3:    $(L_{\text{train}}, L_{\text{dev}}, L_{\text{eval}}) \leftarrow$  Randomly split label set  $\mathcal{L}$  into 3 based on constraint  $\mathcal{K}$
- 4:    $\forall s \in \{\text{train}, \text{dev}, \text{eval}\} : \mathcal{D}_s \leftarrow \{x \in \mathcal{D}' \mid x \text{ is labeled as } l, l \in L_s\}$
- 5:    $e \leftarrow \text{CalculateError}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{eval}})$
- 6:    $\mathcal{P}_{\text{cand}} \leftarrow \mathcal{P}_{\text{cand}} \cup \{((\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{eval}}), e)\}$
- 7: **end while**
- 8: **return**  $\mathcal{P}_{\text{cand}}$

---



---

**Algorithm 2** Heuristic Search Method Step 2: Data Split Selection
 

---

**Require:** Candidate split set  $\mathcal{P}_{\text{cand}}$  (output of Alg. 1), number of adopted data split sets  $N$   
**Require:** Diversity criterion  $\mathcal{M}$   
**Ensure:** Adopted data split set  $\mathcal{S}_{\text{final}}$

- 1:  $\mathcal{S}_{\text{final}} \leftarrow \emptyset$
- 2: **while**  $|\mathcal{S}_{\text{final}}| < N$  **do**
- 3:    $\mathcal{P}_{\text{remain}} \leftarrow \mathcal{P}_{\text{cand}} - \mathcal{S}_{\text{final}}$
- 4:    $c_{\text{next}} \leftarrow \text{SelectCandidate}(\mathcal{P}_{\text{remain}}, \mathcal{S}_{\text{final}}, \mathcal{M})$
- 5:   **if**  $c_{\text{next}}$  is None **then**
- 6:     **break**
- 7:   **end if**
- 8:    $\mathcal{S}_{\text{final}} \leftarrow \mathcal{S}_{\text{final}} \cup \{c_{\text{next}}\}$
- 9: **end while**
- 10: **return**  $\mathcal{S}_{\text{final}}$

---

the target ratio of 3:1:1 for data volume in subsets train, dev, and eval. Therefore, when the number of label types is small, a target ratio is set individually and all combinations satisfying that ratio are enumerated. This procedure is called the complete enumeration combination method. As shown in Algorithm 3, the procedure generates all label split candidates that split the label set  $\mathcal{L}$  into three mutually exclusive subsets  $(L_{\text{train}}, L_{\text{dev}}, L_{\text{eval}})$  to satisfy the target allocation ratio  $r_{\text{train}} : r_{\text{dev}} : r_{\text{eval}}$  for each subset. Then, for each label split, data with the corresponding labels is extracted to construct subsets.

*When Only Two Label Types Exist.* When only two label types exist for the condition to be made unknown, it is not possible to construct a split that is mutually unknown to all three subsets. In that case, one label is assigned to eval and the other label is assigned to train and dev. This policy is called the **binary split method**. For allocation to train and dev, random splitting is performed based on the predetermined ratio  $|D_{\text{train}}| : |D_{\text{dev}}|$  while considering factors such as the ratio of utterance counts.

**Algorithm 3** Data Splitting by Complete Enumeration Combination Method**Require:** Cleaned data  $\mathcal{D}'$ , label set of recording condition to be made unknown  $\mathcal{L}$ **Require:** Allocation ratio for each subset  $r_{\text{train}}, r_{\text{dev}}, r_{\text{eval}}$ **Ensure:** Data split set  $\mathcal{S}_{\text{total}}$ 

- 1:  $\mathcal{S}_{\text{total}} \leftarrow \emptyset$
- 2:  $\mathcal{P} \leftarrow$  Set of all label splits that divide label set  $\mathcal{L}$  into three mutually exclusive subsets  $(L_{\text{train}}, L_{\text{dev}}, L_{\text{eval}})$  at ratio  $r_{\text{train}} : r_{\text{dev}} : r_{\text{eval}}$
- 3: **for** each split  $(L_{\text{train}}, L_{\text{dev}}, L_{\text{eval}}) \in \mathcal{P}$  **do**
- 4:    $\forall s \in \{\text{train}, \text{dev}, \text{eval}\} : \mathcal{D}_s \leftarrow \{x \in \mathcal{D}' \mid x \text{ is labeled as } l, l \in L_s\}$
- 5:    $\mathcal{S}_{\text{total}} \leftarrow \mathcal{S}_{\text{total}} \cup \{(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{dev}}, \mathcal{D}_{\text{eval}})\}$
- 6: **end for**
- 7: **return**  $\mathcal{S}_{\text{total}}$

**Table 2.** Applied algorithms and settings for each condition in partially-open split

Unknown Condition	$ \mathcal{L} $	Split	Constraint $\mathcal{K}$ and Settings
Recording Env.	4	Algo. 3	$r_{\text{train}}:r_{\text{dev}}:r_{\text{eval}} = 2:1:1$
Playback Device	4	Algo. 3	$r_{\text{train}}:r_{\text{dev}}:r_{\text{eval}} = 2:1:1$
Unauth. Rec. Dev.	2	Binary	$ D_{\text{train}} : D_{\text{dev}}  = 4:1$
Speaker	50	Algo. 1,2	$r_{\text{train}}:r_{\text{dev}}:r_{\text{eval}} = 3:1:1, e_{\text{th}} \leq 0.6, \text{Jaccard} \geq 0.3$
Position (Env1)	2	Binary	$ D_{\text{train}} : D_{\text{dev}}  = 4:1$
Position (Env2)	18	Algo. 1,2	$ L_{\text{train}}  \geq 6$
Position (Env4)	7	Algo. 1,2	$ L_{\text{train}}  \geq 3$

**Partially-open Split: Application to Each Condition.** This section shows the specific settings and processing procedures for applying the splitting algorithms explained in the previous section to each recording condition included in ReMASC and actually constructing partially-open splits. Table 2 summarizes the number of labels  $|\mathcal{L}|$ , the adopted splitting policy algorithm, and constraint condition  $\mathcal{K}$  for each recording condition set as an unknown condition.

*Recording Environment.* The label set  $\mathcal{L}$  for recording environment consists of four types: Env1, Env2, Env3, and Env4 ( $|\mathcal{L}| = 4$ ). Since the number of labels is small, the complete enumeration combination method was applied, and subsets were generated with constraint condition  $\mathcal{K}$  set to  $r_{\text{train}}:r_{\text{dev}}:r_{\text{eval}} = 2:1:1$ . As a result, the final number of data split sets created was 12.

*Playback Device.* There are five types of playback device labels in total, but the fifth type, the in-vehicle speaker, is included only in Env4. Therefore, in this split, Env4 data was excluded, and data splitting was performed using four types of playback device labels. Under this setting, the complete enumeration combination method was applied as the splitting algorithm, and subsets were generated with constraint condition  $\mathcal{K}$  set to  $r_{\text{train}}:r_{\text{dev}}:r_{\text{eval}} = 2:1:1$ . As a result, the final number of data split sets created was 12.

*Replay source recorder.* Since there are only two types of replay source recorder labels, the binary split method was applied. In this case, the number of data split sets is 2. The data allocation ratio for train and dev was set to  $|D_{\text{train}}|:|D_{\text{dev}}| = 4:1$ .

*Speaker.* There are 49 types of speaker labels (Spk40 was eliminated in data cleaning). Complete enumeration of all combinations is not practical, so a heuristic search method was used. Constraint condition  $\mathcal{K}$  was set to  $r_{\text{train}}:r_{\text{dev}}:r_{\text{eval}} = 3:1:1$ , and data split candidates were generated. The procedure for diversity criterion  $\mathcal{M}$  is as follows: (1) Randomly select a data split candidate from the data split candidate set. (2) Calculate error  $e_{\text{utt}}$  for the selected data split candidate as the sum of absolute differences between the utterance count ratio per subset and the target ratio. (3) Similarly, calculate error  $e_{\text{b/s}}$  as the sum of absolute differences between the bonafide/spoof utterance count ratio per subset and the target ratio. (4) If both error  $e_{\text{utt}}$  and error  $e_{\text{b/s}}$  are 0.6 or less, calculate the Jaccard distance per subset with already selected data split candidates, and adopt data splits where the Jaccard distance is 0.3 or more for all subsets. This procedure was repeated until the number of data split sets reached 10.

*Position Information.* Position information labels have different meanings and numbers of label types for each recording environment, so the splitting policy needs to be changed for each environment. Below, specific settings for each environment are shown.

**Env1:** Position information labels consist of only two types, “near distance” and “far distance,” so the binary split method was applied. In this case, the number of data split sets is 2. The data allocation ratio for train and dev was set to  $|D_{\text{train}}|:|D_{\text{dev}}| = 4:1$ .

**Env2:** Position information labels are defined as 18 types in total by the product of 3 installation positions of the recording device and 6 combinations of distance and angle of the sound source relative to the device. Since the number of labels is large and exhaustive search is difficult, the heuristic search method was used. Constraint condition  $\mathcal{K}$  was set to require that the train subset contains at least 6 types of labels ( $|L_{\text{train}}| \geq 6$ ). Diversity criterion  $\mathcal{M}$  was defined with the following procedure: (1) Calculate error  $e_{\text{utt}}$  and  $e_{\text{b/s}}$  for each data split candidate in the same way as when the speaker is the unknown condition. (2) Prepare new data split candidates by keeping only those where both errors are 0.011 or less. (3) For those candidates, calculate the Jaccard distance per subset with all already selected data split candidates, and adopt the data split that minimizes the average Jaccard distance across the 3 subsets. This procedure was repeated until the number of data split sets reached 10.

**Env3:** Since position information is fixed to one pattern, this was excluded from partially-open splits regarding position.

**Env4:** Bonafide speech is assigned running state labels and seat positions 1–6, while spoofed speech is assigned only seat positions 0–6, and the number of labels does not match. Therefore, the heuristic search method was applied only for seat positions 1–6 common to both. Constraint condition  $\mathcal{K}$  was set

to require that the train subset contains at least 3 positions ( $|L_{\text{train}}| \geq 3$ ) and that subsets consisting only of label 0 are not generated. Diversity criterion  $\mathcal{M}$  was defined with the following procedure: (1) Calculate error  $e_{\text{utt}}$  and  $e_{\text{b/s}}$  in the same way as when the speaker is the unknown condition. (2) For the data split candidate with the smallest  $e_{\text{utt}} + e_{\text{b/s}}$  among the candidates, calculate the Jaccard distance per subset with already selected data split candidates, and adopt data splits where the Jaccard distance is 0.3 or more for all subsets. This procedure was repeated until the number of data split sets reached 10.

## 4 Experiments

Using the re-designed, composition-matched subsets, we evaluate existing multi-channel spoofing detection methods to test whether reported device gaps are truly device-driven and to quantify condition effects via fully-closed vs. partially-open splits.

### 4.1 Experimental Conditions

In this experiment, we adopted mch-SSL-AASIST [13] as a spoofing detection method utilizing multi-channel input. This method inputs multi-channel acoustic features extracted by a pre-trained model based on self-supervised learning (SSL) into an AASIST backend [4] for identification, and high performance has been reported in prior studies.

For training, the train subsets of each data split defined in the previous section were used, and for evaluation, the eval subsets were used. The dev set was used for early-stopping monitoring, and training was terminated when the Equal Error Rate (EER) for the dev subset did not improve for 20 consecutive epochs. The final model adopted was the model from the epoch with the minimum EER on the dev subset.

For the SSL frontend, pre-trained wav2vec2.0 XLSR (0.3B) [1] was used, and the backend was trained from random initialization. Also, the parameters of the SSL frontend were updated simultaneously with the backend. Since the recording devices in ReMASC (D2, D3, D4) have different numbers of channels, and the model structure changes accordingly, independent models were trained for each recording device. Other detailed training parameters were set to the same settings as in [13].

As preprocessing, all audio data was downsampled to 16 kHz, and as in [3, 6, 9, 13], the first 1 second of each utterance was extracted and used as input. EER was used as the evaluation metric. Training and evaluation were performed for both fully-closed split and partially-open split, and for the latter, the average EER across multiple subsets is reported.

### 4.2 Experimental Results

Table 3 shows the experimental results for each split condition. First, comparing the conventional ReMASC subset (original), where the speaker is the un-

known condition, with the partially-open split for speaker created in this study, in the original split, the EER across recording devices was relatively close at 7–11%, whereas in the new partially-open split, the performance difference between devices became significantly larger. This is thought to be because the bias in recording conditions included in the conventional split made the performance differences between devices unclear. By unifying the composition ratios, the true performance differences attributable to characteristics inherent to each recording device were clearly observed.

Next, from the results of the partially-open split regarding position information, it can be seen that the relationship between recording device differences and spatial generalization performance differs significantly between quiet environments (Env1, Env2) and the in-vehicle environment (Env4). In Env1 and Env2, differences in EER were observed between devices, suggesting that generalization performance to unknown position conditions depends on device-specific characteristics, such as array shape. Particularly in Env2, despite containing diverse position conditions, D2 and D3 consistently showed low EER, suggesting that these devices are relatively robust to position variations. On the other hand, in Env4, all devices showed high EER, with performance degradation occurring that cannot be explained by the diversity of position conditions. Since devices that performed well in Env2 also uniformly degraded in performance, it is considered that broadband noise and road noise specific to in-vehicle environments were the main factors, rather than position differences, causing the performance degradation.

Comparing the degree of influence of each condition, the fully-closed split showed extremely high accuracy with an average of 2.25%, while all partially-open splits showed performance degradation, with the largest degradation of 29.62% average occurring when the recording environment was unknown. This suggests that the recording environment is the most significant factor in spoofing detection.

To further investigate how the type of unseen recording environment affects performance, we analyzed the results for each subset under the recording-environment partially-open splits. We found that when the evaluation set consisted of Env4, the EER consistently exceeded 39%, indicating a pronounced performance degradation. This finding suggests that the domain mismatch is particularly severe in the in-vehicle environment. Overall, these results indicate that robustness to unseen environments can be improved by incorporating Env4-like conditions via data augmentation and by applying noise superposition.

### 4.3 Discussion

From the results of this study, it became evident that the recording environment is the most dominant factor in multi-channel spoofing detection. In particular, in in-vehicle environments, reverberation, broadband noise, and road noise vary simultaneously, which renders features based on sound source position unstable. This finding suggests that the spatial consistency exploited by existing methods, which assume microphone arrays can be severely degraded by environmental

**Table 3.** EER (%) of mch-SSL-AASIST for each split condition. Average values calculated for all subset splits for each recording device and macro average values for all recording devices are shown.

Split Type	Unknown Condition	EER by Recording Device (%)			
		D2	D3	D4	Avg.
Fully-closed	–	2.26	0.99	3.50	2.25
Partially-open	Recording Env.	28.70	27.51	32.66	29.62
	Playback Device	6.88	0.72	14.40	7.33
	Position (Env1)	8.31	6.10	13.81	9.41
	Position (Env2)	2.33	0.13	13.89	5.45
	Position (Env4)	17.27	15.74	18.26	17.09
	Source Rec. Device	10.27	6.80	9.90	8.99
	Speaker	8.67	4.59	12.26	8.51
Original [13]	Speaker	7.6	10.6	8.3	8.8

noise, can be severely degraded by environmental noise. Consequently, future model designs should explicitly consider environmental normalization and robustness to noise.

Furthermore, by unifying the composition ratios of recording conditions and introducing partially-open splits, this study enabled factor separation that was difficult to achieve with the conventional ReMASK setup. The result confirmed that the proposed framework functions as an evaluation methodology capable of independently analyzing the impact of individual recording conditions on system performance. This framework is applicable not only to spoofing detection but also to a wide range of audio-related tasks, such as far-field speaker recognition [7] and speech quality estimation [5, 10], where multiple recording conditions are intricately intertwined.

On the other hand, this study has several limitations. First, since certain conditions were excluded during data cleaning, the overall coverage of the ReMASC dataset was reduced. In particular, the exclusion of recording device D1 resulted in incomplete preservation of device diversity. Second, although this study primarily focused on mch-SSL-AASIST, a more comprehensive comparison that includes beamforming-based methods [3, 9] and spectrogram-based methods [6] remains an important direction for future work.

## 5 Conclusion

In this study, we redesigned the ReMASC corpus to enable fair and systematic evaluation of spoofing detection using multi-channel information. By unifying the composition ratios of recording conditions and introducing controlled evaluation splits, the proposed framework eliminates condition bias, enabling a precise analysis of device-specific performance characteristics. Experimental results confirm that the effects of individual recording conditions can be independently assessed,

with unknown conditions causing notable performance degradation. The framework facilitates fair model comparison and factor analysis, and provides a solid foundation for spoofing detection and other audio-related tasks.

## References

1. Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., Auli, M.: XLS-R: Self-supervised cross-lingual speech representation learning at scale. arXiv:2111.09296 (2021), <https://arxiv.org/abs/2111.09296>
2. Gong, Y., Yang, J., Huber, J., MacKnight, M., Poellabauer, C.: ReMASC: Realistic replay attack corpus for voice controlled systems. In: Proc. Interspeech. pp. 2355–2359 (2019)
3. Gong, Y., Yang, J., Poellabauer, C.: Detecting replay attacks using multi-channel audio: A neural network-based method. *IEEE Signal Processing Letters* **27**, 920–924 (2020). <https://doi.org/10.1109/lsp.2020.2996908>, <http://dx.doi.org/10.1109/LSP.2020.2996908>
4. Jung, J.W., Heo, H.S., Tak, H., Shim, H.J., Chung, J.S., Lee, B.J., Yu, H.J., Evans, N.: AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In: Proc. ICASSP. pp. 6367–6371 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9747766>
5. Li, C., Wang, W., Zhang, W., Saijo, K., Scheibler, R., Cornell, S., Ni, Z., Kumar, A., Sach, M., Fu, Y., Fingscheidt, T., Watanabe, S., Qian, Y.: ICASSP 2026 URGENT challenge. <https://urgent-challenge.github.io/urgent2026/>
6. Li, Z., Shi, C., Zhang, T., Xie, Y., Liu, J., Yuan, B., Chen, Y.: Robust detection of machine-induced audio attacks in intelligent audio systems with microphone array. In: Proc. ACM SIGSAC. pp. 1884–1899 (2021). <https://doi.org/10.1145/3460120.3484755>, <https://doi.org/10.1145/3460120.3484755>
7. Nandwana, M.K., van Hout, J., Richey, C., McLaren, M., Barrios, M.A., Lawson, A.: The VOICES from a distance challenge 2019. In: Proc. Interspeech. pp. 2438–2442 (2019)
8. Neri, M., Virtanen, T.: Impact of microphone array mismatches to learning-based replay speech detection. In: Proc. EUSIPCO. pp. 1243–1247 (2025)
9. Neri, M., Virtanen, T.: Multi-channel replay speech detection using an adaptive learnable beamformer. In: *IEEE Open Journal of Signal Processing*. vol. 6, pp. 530–535 (2025)
10. Reddy, C.K., Gopal, V., Cutler, R.: DNSMOS: A non-intrusive perceptual objective speech quality metric to evaluate noise suppressors. In: Proc. ICASSP (2021)
11. Talukder, M., Xie, J.: Exploiting playback device’s effect on multi-channel audio to secure voice assistants. In: Proc. GLOBECOM. pp. 6085–6090 (2022)
12. Yaguchi, R., Shiota, S., Ono, N., Kiya, H.: Replay attack detection based on spatial and spectral features of stereo signal. *Journal of Information Processing* **29**, 275–282 (2021)
13. Yamaguchi, T., Shiota, S., Tawara, N.: Investigating self-supervised learning-based front-end for multi-channel replay attack detection. In: Proc. APSIPA ASC. pp. 2098–2103 (2025)
14. Yang, Q., Cui, K., Zheng, Y.: Room-scale voice liveness detection for smart devices. *Trans. on TDSC* **21**(5), 4982–4996 (2024). <https://doi.org/10.1109/TDSC.2024.3367269>