



ATMM-SAGA: Alternating Training for Multi-Module with Score-Aware Gated Attention SASV system

Amro Asali¹, Yehuda Ben-Shimol¹, Itshak Lapidot^{2,3}

¹Electrical and computer engineering school, Ben Gurion University of the Negev, Israel

²Electrical engineering school, Afeka the Academic College of Engineering in Tel Aviv, Israel

³Avignon University, LIA, France

asali@bgu.ac.il, benshimo@bgu.ac.il, itshakl@afeka.ac.il

Abstract

The objective of *automatic speaker verification* (ASV) systems is to determine whether a given test speech utterance corresponds to a claimed enrolled speaker. These systems have a wide range of applications, and ensuring their reliability is crucial. In this paper, we propose a *spoofing-robust automatic speaker verification* (SASV) system employing a *score-aware gated attention* (SAGA) fusion scheme, integrating scores from a pre-trained countermeasure (CM) with speaker embeddings from a pre-trained ASV. Specifically, we employ the AASIST and ECAPA-TDNN models. SAGA acts as an adaptive gating mechanism, where the CM score determines how strongly ASV embeddings influence the final SASV decision. Experiments on the ASVspoof2019 *logical access* dataset demonstrate that the proposed SASV system achieves an SASV *equal error rate* (SASV-EER) and *agnostic detection cost function* (a-DCF) of 2.31%, 0.0603 for the development set and 2.18%, 0.0480 for the evaluation set.

Index Terms: spoofing-robust automatic speaker verification, countermeasure, score-aware gated attention, alternating training for multi-module (ATMM)

1. Introduction

Recent studies have demonstrated that ASV systems are undergoing a gradual evolution, acquiring the capacity to reject spoofed inputs in a zero-shot manner. However, rapid advancements in speech synthesis techniques, such as *text-to-speech* (TTS) or *voice conversion* (VC), highlight the ongoing necessity to further enhance spoofing-robust ASV systems [1]. The findings indicate that contemporary SASV systems exhibit superior performance in terms of the SASV-EER, as measured by trials encompassing all three classes: target, bona fide non-target, and spoofed non-target [2].

Reliable speaker verification is frequently comprised of two distinct subsystems: ASV [3,4] and spoofing CM [5–7] classifiers. This prompts the following research question: how should the two subsystems be integrated to achieve robust speaker verification? The results presented in [8] indicate that while joint optimization enhances the reliability of ASV at the SASV level, superior performance is achieved by integrating fixed pre-trained subsystems. The CM and ASV can be combined in a cascade or in a parallel fashion [9–11], with integration typically occurring at the score or embedding levels, as evidenced in [12–14].

2. Background

This section presents a concise review of the pertinent literature on the proposed solution, along with a concise overview of the

ASV and CM systems employed for embedding extraction.

2.1. Automatic speaker verification system

In this study, the *emphasized channel attention, propagation, and aggregation time delay neural network* (ECAPA-TDNN) speaker verification system is employed [4]. The system utilizes 80-dimensional *Mel-frequency cepstral coefficients* (MFCCs) as features and incorporates a modified Res2Net as its backbone processing block, augmented with dimensional *squeeze-excitation* (SE) blocks, to model global channel interdependencies. The model incorporates attentive statistics pooling, enabling the model to select relevant frames, and multi-layer feature aggregation, which captures both shallow and complex speaker identity features at the frame level and aggregates them into utterance-level embeddings.

2.2. CM system

In this study, we adopt the *audio anti-spoofing using integrated spectro-temporal graph attention networks* (AASIST) approach, which utilizes the RawNet2 frontend and graph attention framework [5]. The model employs a Sinc convolution encoder to extract time-frequency representations, which are then processed by a residual network to learn high-level features. Subsequently, two graph modules are employed to model the spectral and temporal domains, respectively.

2.3. Structural transformation on ReLU

For the affine layer $W_i x + b_i$, with learnable weights and biases W_i and b_i , respectively, we define a structural transformation $t\text{ReLU}$ as its activation function. $t\text{ReLU}$ is defined as follows:

$$t\text{ReLU}_{W_a}(W_i x + b_i) = \max(W_a(W_i x + b_i), \mathbf{0}) \quad (1)$$

where W_a is the learnable structural transformation, initialized as the identity matrix. Here, the $\max(\cdot, \cdot)$ operation is performed element-wise. Same definition with a diagonal constraint of W_a was implemented in [14–16].

3. Proposed system

In this section, we will present the operation of the proposed system, which has been designed to address the SASV problem. Given a pair of utterances, an enrollment utterance U_{enl} of the target speaker and a test utterance U_{tst} , the system will evaluate whether U_{tst} was spoken by the target speaker (output $y = 1$) or by a non-target speaker ($y = 0$). Non-target attacks can be either zero-effort impostor attacks or spoofing attacks.

<https://github.com/amro-asali-2/ATMM-SAGA>

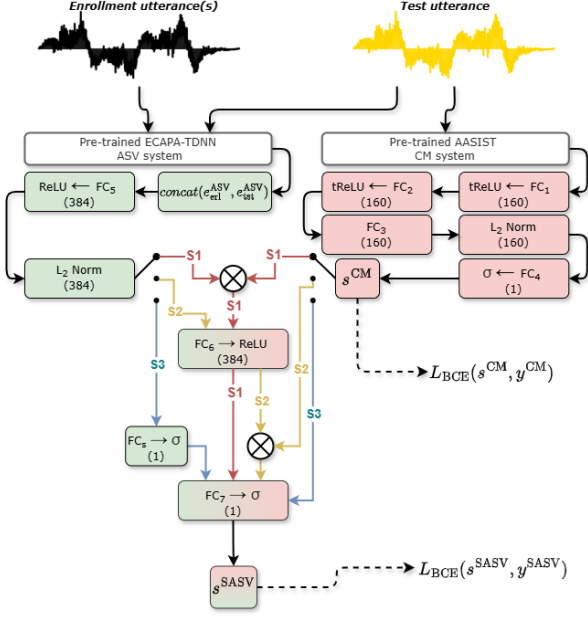


Figure 1: Diagram of the proposed SASV system, illustrating various CM score integration strategies, represented by distinct color-coded paths. Dashed arrows denote operations exclusive to the training stage.

3.1. Model architecture

This study investigates the integration of speaker embeddings, derived from a pre-trained ECAPA-TDNN model [4], with a CM score, based on a pre-trained AASIST embeddings [5]. This integration is achieved by multiplying the obtained CM score with the normalized activations of the speaker embeddings, thereby generating spoofing-aware speaker embeddings. This approach is referred to as *score-aware gated attention* (SAGA). These embeddings are subsequently employed to calculate SASV scores. The proposed system is illustrated in Figure 1. In this investigation two integration levels are examined while maintaining the integrity of the overall system architecture. Additionally, a score fusion strategy is explored to facilitate a more comprehensive comparative analysis.

3.1.1. Strategy S1: Early Integration

As depicted in Figure 1, S1 follows the red path, where the AASIST’s CM embeddings are processed through two fully connected layers on the top right processing path (colored soft red in the figure). These embeddings undergo parameter-shared *tReLU* activations, as defined in Equation 1. Subsequently, the embeddings pass through an additional fully connected layer and are normalized using L2-normalization, and are then processed through a final fully connected layer with a Sigmoid activation, ultimately yielding a CM score, $s^{CM} \in [0, 1]$.

Concurrently, on the left processing path (colored green in the figure), the ECAPA-TDNN speaker embeddings are concatenated and processed through a fully connected layer, followed by the classic ReLU activation function. These embeddings are then normalized using L2-normalization to obtain the normalized speaker embeddings, e^{ASV} . To incorporate SAGA, a multiplicative gating mechanism is applied, in which the CM score functions as an adaptive attention weight over the ASV

embeddings:

$$e^{SASV} = g(s^{CM}, e^{ASV}) = s^{CM} e^{ASV} \quad (2)$$

where $g(s^{CM}, e^{ASV})$ represents the SAGA operation, ensuring that spoofed samples ($s^{CM} \approx 0$) are suppressed while bona fide samples ($s^{CM} \approx 1$) are preserved. e^{SASV} represents the spoofing-aware speaker verification embeddings. These spoofing-aware embeddings undergo further processing through an additional fully connected layer with ReLU activation before being processed through a final fully connected layer with a Sigmoid activation, ultimately yielding an SASV score.

3.1.2. Strategy S2: Late Integration

While S1 and S2 strategies share the same overall architecture, the approach of S2 involves delaying the application of the SAGA mechanism, integrating it later in the ASV processing path (yellow path in Figure 1).

3.1.3. Strategy S3: Score Fusion

In this strategy, score fusion between the ASV and CM systems is achieved through a fully connected layer to generate the SASV score. This approach (blue path in Figure 1) is used exclusively for comparative analysis.

3.2. Training

In order to train the model for both speaker verification and countermeasure tasks, a multi-task learning paradigm is adopted as outlined in [17]. During the training phase of our proposed system, the total multi-task classification loss is defined as follows:

$$L^{total} = \lambda \cdot L_{BCE}^{SASV}(s^{SASV}, y^{SASV}) + (1-\lambda) \cdot L_{BCE}^{CM}(s^{CM}, y^{CM}) \quad (3)$$

where $\lambda \in [0, 1]$ is fixed at 0.5 to assign equal importance to both tasks. In this context, $L_{BCE}^{SASV}(\cdot, \cdot)$ denotes the *binary cross-entropy* (BCE) loss calculated between the system’s SASV output score $s^{SASV} \in [0, 1]$ and the SASV label $y^{SASV} \in \{0, 1\}$. In contrast, $L_{BCE}^{CM}(\cdot, \cdot)$ represents the BCE loss calculated between the system’s CM output score $s^{CM} \in [0, 1]$ and the CM label $y^{CM} \in \{0, 1\}$, where $y^{CM} = 1$ indicates a bona fide trial and $y^{CM} = 0$ represents a spoof trial.

3.2.1. Alternating Training for Multi-Module (ATMM)

During the training phase, two separate datasets are employed, obtained by partitioning the unified training set used for training the SAGA SASV system without employing ATMM, as described in Table 1. The first dataset is only used for the spoofing countermeasure training, while the second dataset is only used for the speaker verification training. For the CM training dataset, the ASVspoof2019 *logical access* (LA) train set [18] is employed to generate pairs of enrollment and test utterances. Specifically, each bona fide utterance is paired with a random subset of the same speaker’s bona fide utterances to form target trials and with other speakers’ bona fide utterances to form zero-effort non-target trials. Additionally, we have paired bona fide utterances with the same speaker’s spoofed utterances to create spoof non-target trials.

The speaker verification training dataset is comprised of the VoxCeleb1 E and H partitions, and the ASVspoof2019 LA training set bona fide pairs. The ATMM algorithm optimizes joint training by alternating update focus between the two

Table 1: Overview of the training datasets used in our experiments, detailing the number of target, non-target, and spoofed trials for both the spoofing CM and speaker verification datasets.

Dataset	Target	Non-target	Spoof
Spoofing CM	262228	249094	463910
Speaker verification	806025	779601	0

Algorithm 1 ATMM: One Round of Training

- 1: **for** 100 iterations **do**
 - 2: Choose a random $p \in \{0, 1\}$.
 - 3: **if** $p = 0$ **then**
 - 4: Set $\lambda \leftarrow 0.1$.
 - 5: Sample 1% of the spoofing CM dataset.
 - 6: Freeze the speaker verification weights (colored in green), as shown in Figure 1.
 - 7: **else**
 - 8: Set $\lambda \leftarrow 0.9$.
 - 9: Sample 1% of the speaker verification dataset.
 - 10: Freeze the CM weights (colored in soft red), as shown in Figure 1.
 - 11: **end if**
 - 12: Compute the total loss according to Equation 3.
 - 13: Perform backpropagation and update the weights of the unfrozen components.
 - 14: **end for**
-

modalities. At each training step, a binary decision determines whether to prioritize the ASV or CM module. This prevents overfitting to either task, while preserving previously learned knowledge with selective backpropagation achieved through strategic weight freezing. It is important to note that a pair of utterances (enrollment and test) is required for both CM and ASV training, since the joint representation layers (depicted with a mix of soft red and green in Figure 1) remain unfrozen, ensuring continuous learning. Maintaining $\lambda \in (0, 1)$, empirically chosen to alternate between 0.1 and 0.9, allows for continuous adaptive gradient flow from both tasks, preserving joint feature learning while preventing overfitting. A detailed description of the training algorithm is provided in Alg.1.

4. Experimental setup

In the following, we describe the experimental setup for our proposed SASV system, detailing the datasets used for evaluation and the metrics employed to assess performance.

4.1. Dataset

The ASVspoof 2019 LA dataset is a widely used benchmark [18]. This dataset comprises genuine speech utterances and those that have been spoofed using a variety of text-to-speech (TTS) and voice conversion (VC) techniques. The dataset is split into three sets: training (Train), development (Dev), and evaluation (Eval). It includes official development and evaluation protocols, and for each trial, multiple corresponding enrollment utterances are provided to register the target speaker. Notably, the training and development sets incorporate the same six spoofing attacks (A01–A06), which include four TTS and two VC attacks. In contrast, the evaluation set

comprises 11 previously unseen attacks (A07–A15, A17, A18) and two additional attacks (A16, A19) that, despite employing similar underlying algorithms as some training attacks, are trained with different data. The evaluation set includes 5,370 target trials, 33,327 non-target trials, and 63,882 spoofed trials.

4.2. Evaluation Metrics

4.2.1. SASV-EER

The SASV-EER evaluates system performance across all three trial types: target, zero-effort non-target, and spoofed non-target. It is defined as the error rate at the threshold where the false rejection rate of target trials equals the false acceptance rate of both zero-effort and spoofed non-target trials, providing a unified measure of robustness against both speaker mismatch and spoofing attacks.

4.2.2. Minimum Normalized Agnostic Detection Cost Function (min a-DCF)

The min a-DCF evaluates system performance by optimizing the Bayes risk while incorporating class priors and detection costs [19]:

$$\min_t \frac{C_{\text{miss}}\pi_{\text{tar}}P_{\text{miss}}(t) + C_{\text{fa,non}}\pi_{\text{non}}P_{\text{fa,non}}(t) + C_{\text{fa,spf}}\pi_{\text{spf}}P_{\text{fa,spf}}(t)}{\min\{C_{\text{miss}}\pi_{\text{tar}}, C_{\text{fa,non}}\pi_{\text{non}} + C_{\text{fa,spf}}\pi_{\text{spf}}\}} \quad (4)$$

where P_{miss} , $P_{\text{fa,non}}$, $P_{\text{fa,spf}}$ are the miss rate (false rejection), non-target false acceptance rate, and spoof false acceptance rate at threshold t , respectively. The terms π_{tar} , π_{non} , π_{spf} denote the priors for target, non-target, and spoof trials, while C_{miss} , $C_{\text{fa,non}}$, $C_{\text{fa,spf}}$ are their respective detection costs.

Unlike the EER metric, which operates independently of class priors and decision costs, min a-DCF provides a more comprehensive evaluation by considering the trade-off between different types of errors under real-world conditions.

5. Results

In this section, we present a summary of our experimental findings. The present study commences with an examination of the influence of distinct training methodologies on model performance and its generalizability to unseen attacks. In the subsequent step, an evaluation of the various strategies for integrating the CM score is conducted, with a focus on the effectiveness of the proposed SASV system in comparison with several baseline methods. The assessment of performance is conducted through the utilization of the SASV-EER and min a-DCF methodologies. Confidence intervals are computed via bootstrapping, utilizing 1,000 iterations and a 95% confidence level [20].

5.1. Comparison of Training Approaches

In order to enhance the robustness of the SASV system, a series of experiments were conducted, in which various training approaches were implemented in conjunction with the S1 integration strategy. These experiments involved modifying the training algorithm, as detailed in Alg. 1. In addition, we incorporated regularization techniques including *dropout* and *batch normalization* (BN) layers to mitigate the risk of overfitting and stabilize the training process. By systematically evaluating different training configurations, we aimed to identify the most effective combination for enhancing model generalization.

The results in Table 2 indicate that while conventional regularization techniques, such as BN and dropout, have been effec-

Table 2: Performance comparison of different training technique combinations with S1 integration strategy, evaluated in terms of min a-DCF and SASV-EER, with confidence intervals included, on the ASVspoof2019 LA dataset’s development and evaluation sets.

min a-DCF		EER (%)		BN	Drop	ATMM
Dev	Eval	Dev	Eval			
0.0500	0.1464	1.46	5.74	✗	✗	✗
[0.0409, 0.0567]	[0.1413, 0.1517]	[1.27, 1.77]	[5.61, 5.90]			
0.1189	0.1386	4.45	5.22	✓	✗	✗
[0.1174, 0.1430]	[0.1321, 0.1439]	[3.39, 5.17]	[5.01, 5.47]			
0.0653	0.1422	2.25	5.58	✗	✓	✗
[0.0552, 0.0730]	[0.1364, 0.1466]	[1.83, 2.55]	[5.38, 5.69]			
0.1079	0.1315	4.48	4.98	✓	✓	✗
[0.0933, 0.1185]	[0.1243, 0.1363]	[3.41, 5.16]	[4.77, 5.15]			
0.0603	0.0480	2.31	2.18	✗	✗	✓
[0.0522, 0.0609]	[0.0435, 0.0527]	[1.63, 3.15]	[1.81, 2.56]			
0.0975	0.0702	6.60	4.21	✓	✗	✓
[0.0863, 0.1094]	[0.0644, 0.0754]	[5.48, 7.59]	[3.77, 4.57]			
0.0620	0.0516	2.42	2.27	✗	✓	✓
[0.0519, 0.0699]	[0.0474, 0.0554]	[1.66, 3.31]	[1.98, 2.60]			
0.0937	0.0707	6.84	4.17	✓	✓	✓
[0.0818, 0.1051]	[0.0648, 0.0752]	[5.74, 7.87]	[3.79, 4.49]			

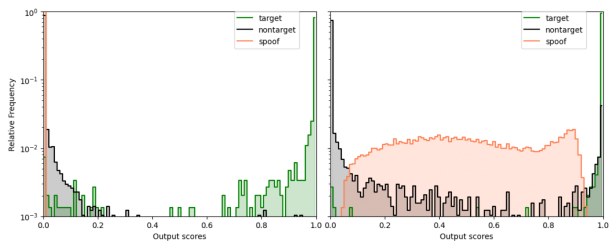


Figure 2: Log-scale normalized histograms of output scores for target, zero-effort non-target, and spoof trials on the ASVspoof2019 LA development dataset. The left plot represents results from a system trained using conventional training (without ATMM), while the right plot corresponds to the system trained with ATMM. Both systems utilize the S1 strategy.

tive in some cases, modifying the training procedure, as detailed in Alg.1, leads to significant improvements. Notably, applying ATMM without BN and without dropout (colored in blue in Table 2) resulted in the lowest min a-DCF and SASV-EER scores on the evaluation set. However, it performed worse on the development set compared to the model without ATMM (colored in orange in Table 2). This discrepancy can be attributed to the development set containing the same attacks as the training set. This suggests that a model trained with ATMM is less prone to overfitting to training attacks and generalizes better to unseen attacks. Figure 2 further suggests that ATMM prevents overfitting by balancing score distributions. Without ATMM (left side), spoof scores are tightly clustered near zero, indicating overconfidence. After employing ATMM (right side), spoof scores are more evenly distributed while maintaining a clear separation from target scores. This suggests that careful adjustment of the training algorithm can be more effective in reducing overfitting and enhancing model performance than standard regularization methods.

5.2. Comparison of CM Score Integration Strategies and Baseline Systems

In this subsection, we analyze the optimal strategy for integrating the CM score into the SASV system and compare the proposed solution with fusion-based baseline systems from the

Table 3: Comparison of different CM score integration strategies with SASV baselines, standalone ASV and CM systems, evaluated in terms of min a-DCF and SASV-EER, with confidence intervals included, on the ASVspoof2019 LA dataset’s development and evaluation sets.

Systems	SASV-EER (%)		min a-DCF	
	Dev	Eval	Dev	Eval
ECAPA-TDNN [4]	17.31	23.84	-	-
AASIST [5]	15.86	24.38	-	-
Baseline1 [2]	13.06	19.31	-	-
G-SASV [14]	-	8.62	-	-
Baseline2 [2]	3.10	6.54	-	-
S3	3.87	5.45	0.1087	0.1245
	[3.40, 4.79]	[4.95, 5.87]	[0.0942, 0.1198]	[0.1127, 0.1354]
S1	2.31	2.18	0.0603	0.0480
	[1.63, 3.15]	[1.81, 2.56]	[0.0522, 0.0609]	[0.0435, 0.0527]
S2	2.28	2.19	0.0571	0.0501
	[1.62, 3.17]	[1.86, 2.48]	[0.0490, 0.0651]	[0.0454, 0.0535]

SASV2022 challenge [2] and the multi-task-based G-SASV system from [14]. As shown in Table 3, SAGA outperforms score fusion, with the S1 strategy achieving slightly better performance than S2, indicating that early integration is a better strategy than late integration. To further demonstrate the efficacy of the proposed approach, we compare it against individual ASV and CM systems, which are utilized for feature extraction. While the ASV and CM subsystems achieve state-of-the-art performance on their respective tasks, they prove ineffective when applied individually to the spoofing-robust automatic speaker verification task, as reported in [12, 14]. As demonstrated in Table 3, the confidence intervals for the employed evaluation metrics on the evaluation dataset for the proposed solution lie entirely below those of the baseline and individual systems. This indicates that integration using the ATMM-SAGA training and integration framework yields statistically significant improvements over both embeddings and score fusion.

6. Conclusions and Future Work

This paper presents a robust SASV system that integrates CM scores with speaker embeddings using the SAGA mechanism and ATMM algorithm. The proposed approach enables seamless fusion of ASV and CM pre-trained models while maintaining a compact and efficient structure. Our results demonstrate that SAGA is the superior method for incorporating CM scores, significantly outperforming conventional score fusion. Furthermore, applying the SAGA mechanism early in the network enhances the system’s ability to discriminate between bona fide and spoofed samples. We also show that ATMM surpasses conventional regularization techniques, such as BN and dropout, in mitigating overfitting and improving generalization. By alternating between ASV and CM training, ATMM effectively balances modality learning and strengthens feature discrimination, leading to superior SASV performance. Despite these advancements, there remains considerable room for further optimization. Future work may focus on refining the system by enhancing the SAGA mechanism, optimizing the ATMM algorithm, and exploring complementary feature spaces.

7. Acknowledgments

This work is supported by the Israel Innovation Authority under project numbers 82457 and 82458.

8. References

- [1] J. weon Jung, X. Wang, N. Evans, S. Watanabe, H. jin Shim, H. Tak, S. Arora, J. Yamagishi, and J. S. Chung, "To what extent can ASV systems naturally defend against spoofing attacks?" in *Interspeech 2024*, 2024, pp. 3240–3244.
- [2] J. weon Jung, H. Tak, H. jin Shim, H.-S. Heo, B.-J. Lee, S.-W. Chung, H.-J. Yu, N. Evans, and T. Kinnunen, "SASV 2022: The first spoofing-aware speaker verification challenge," in *Interspeech 2022*, 2022, pp. 2893–2897.
- [3] T. Liu, K. A. Lee, Q. Wang, and H. Li, "Golden gemini is all you need: Finding the sweet spots for speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2324–2337, 2024.
- [4] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech 2020*, ser. interspeech-2020. ISCA, Oct. 2020. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2020-2650>
- [5] J.-w. Jung, H.-S. Heo, H. Tak, H.-j. Shim, J. S. Chung, B.-J. Lee, H.-J. Yu, and N. Evans, "AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6367–6371.
- [6] H. Tak, J. weon Jung, J. Patino, M. Kamble, M. Todisco, and N. Evans, "End-to-End Spectro-Temporal Graph Attention Networks for Speaker Verification Anti-Spoofing and Speech Deepfake Detection," in *Proceedings of Interspeech 2021*, September 2021.
- [7] K. Borodin, V. Kudryavtsev, D. Korzh, A. Efimenko, G. Mkrtchian, M. Gorodnichev, and O. Y. Rogov, "AASIST3: KAN-enhanced AASIST speech deepfake detection using SSL features and additional regularization for the ASVspoof 2024 Challenge," in *The Automatic Speaker Verification Spoofing Countermeasures Workshop (ASVspoof 2024)*, 2024, pp. 48–55.
- [8] W. Ge, H. Tak, M. Todisco, and N. Evans, "On the potential of jointly-optimised solutions to spoofing attack detection and automatic speaker verification," in *IberSPEECH 2022*, 2022, pp. 51–55.
- [9] M. Sahidullah, H. Delgado, M. Todisco, H. Yu, T. Kinnunen, N. Evans, and Z.-H. Tan, "Integrated Spoofing Countermeasures and Automatic Speaker Verification: An Evaluation on ASVspoof 2015," in *Proceedings of the 17th Annual Conference of the International Speech Communication Association (Interspeech)*, 2016, pp. 1700–1704.
- [10] A. Weizman, Y. Ben-Shimol, and I. Lapidot, "Spoofing-Robust Speaker Verification Based on Time-Domain Embedding," in *Cyber Security, Cryptology, and Machine Learning*, S. Dolev, M. Elhadad, M. Kutylowski, and G. Persiano, Eds. Cham: Springer Nature Switzerland, 2025, pp. 64–78.
- [11] —, "Tandem spoofing-robust automatic speaker verification based on time-domain embeddings," 2024. [Online]. Available: <https://arxiv.org/abs/2412.17133>
- [12] Y. Zhang, G. Zhu, and Z. Duan, "A probabilistic fusion framework for spoofing aware speaker verification," in *The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 77–84.
- [13] J.-H. Choi, J.-Y. Yang, Y.-R. Jeoung, and J.-H. Chang, "HYU Submission for the SASV Challenge 2022: Reforming Speaker Embeddings with Spoofing-Aware Conditioning," in *Interspeech 2022*, 2022, pp. 2873–2877.
- [14] X. Liu, M. Sahidullah, K. A. Lee, and T. Kinnunen, "Generalizing speaker verification for spoof awareness in the embedding space," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, p. 1261–1273, 2024. [Online]. Available: <http://dx.doi.org/10.1109/TASLP.2024.3358056>
- [15] C. Zhang and P. C. Woodland, "DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5300–5304.
- [16] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, "Adaptation algorithms for neural network-based speech recognition: An overview," *IEEE Open Journal of Signal Processing*, vol. 2, p. 33–66, 2021. [Online]. Available: <http://dx.doi.org/10.1109/OJSP.2020.3045349>
- [17] J. Li, Z. Wu, J. Dang, and H. Li, "Joint Decision of Anti-Spoofing and Automatic Speaker Verification by Multi-Task Learning With Contrastive Loss," *IEEE Access*, vol. 8, pp. 58 534–58 542, 2020.
- [18] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, L. Juvela, P. Alku, Y.-H. Peng, H.-T. Hwang, Y. Tsao, H.-M. Wang, S. L. Maguer, M. Becker, F. Henderson, R. Clark, Y. Zhang, Q. Wang, Y. Jia, K. Onuma, K. Mushika, T. Kaneda, Y. Jiang, L.-J. Liu, Y.-C. Wu, W.-C. Huang, T. Toda, K. Tanaka, H. Kameoka, I. Steiner, D. Matrouf, J.-F. Bonastre, A. Govender, S. Ronanki, J.-X. Zhang, and Z.-H. Ling, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Computer Speech & Language*, vol. 64, pp. 101–114, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0885230820300474>
- [19] H.-J. Shim, J.-W. Jung, T. Kinnunen, N. Evans, J.-F. Bonastre, and I. Lapidot, "a-DCF: an architecture agnostic metric with application to spoofing-robust speaker verification," in *Odyssey 2024*, 06 2024.
- [20] L. Ferrer and P. Riera, "Confidence intervals for evaluation in machine learning." [Online]. Available: <https://github.com/luferrer/ConfidenceIntervals>