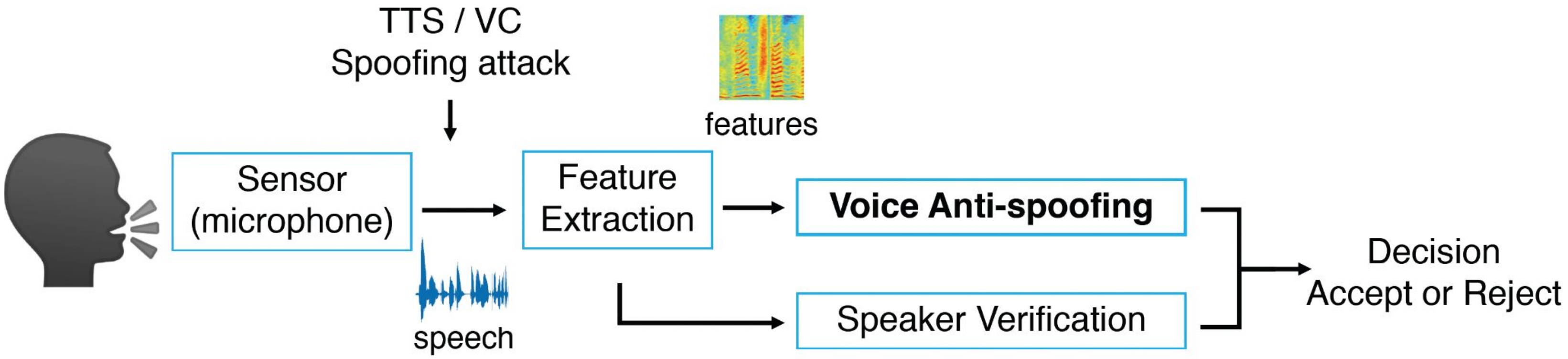
OTM-Titanet: Leveraging Pre-trained Speaker Embeddings with Optimal Transport Memory for Audio Anti-Spoofing

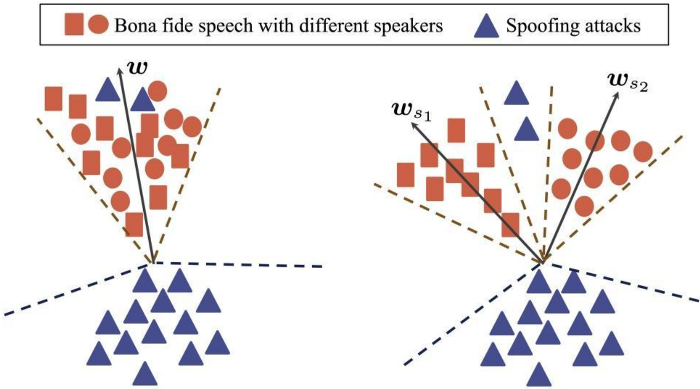
Fei-Yueh Chen
ECE 477 Final Paper

Introduction



Related Works - Training Loss

- OC-Softmax (Zhang et al., 2021): To compact the bona fide speech representation and inject an angular margin to separate the spoofing attacks in the embedding space.
  -> **Assume all data has the same center**
- SAMO (Ding et al., 2023): To cluster bona fide speech around a number of speaker attractors and pushes away spoofing attacks from all the attractors in a high-dimensional embedding space.
  -> **Require Speaker ID as enrollment**



(a) One-class learning (b) SAMO

Rethinking training loss: Could we create multiple pseudo labels for training?

Related Works - Model Architecture

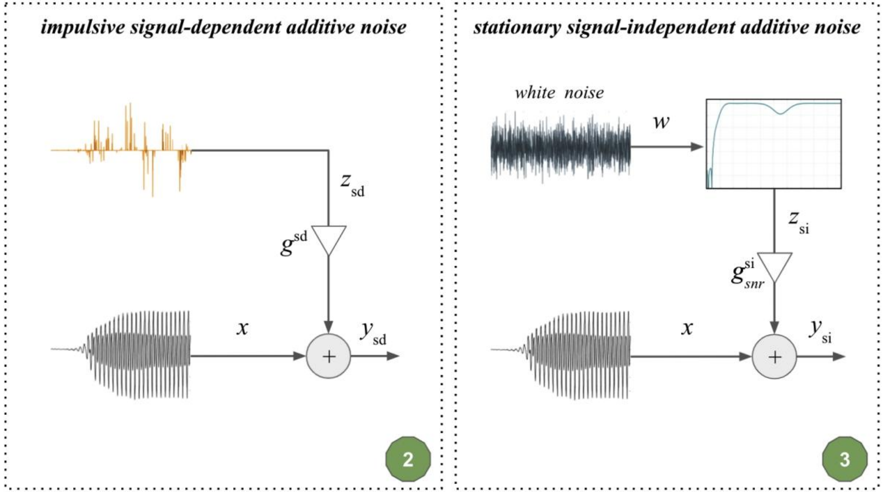
- Wav2Vec-Conformer (Rosello et al., 2023): Use XLS-R with comformer blocks.
- Wav2Vec-TCM (Truong et al., 2024): Use XLS-R + comformer blocks with Temporal-channel modeling.
- Wav2Vec-SCL (Doan et al., 2024): Use XLS-R with three linear layers.
- (The above names are from Kwok et al. (2025))

They successfully demonstrate that a good pretrained audio encoder / feature extractor is sufficient for anti-spoofing.

-> Could we use a pretrained speaker embedding model as the backbone, instead of a general audio encoder (From 300M params to 10M)?

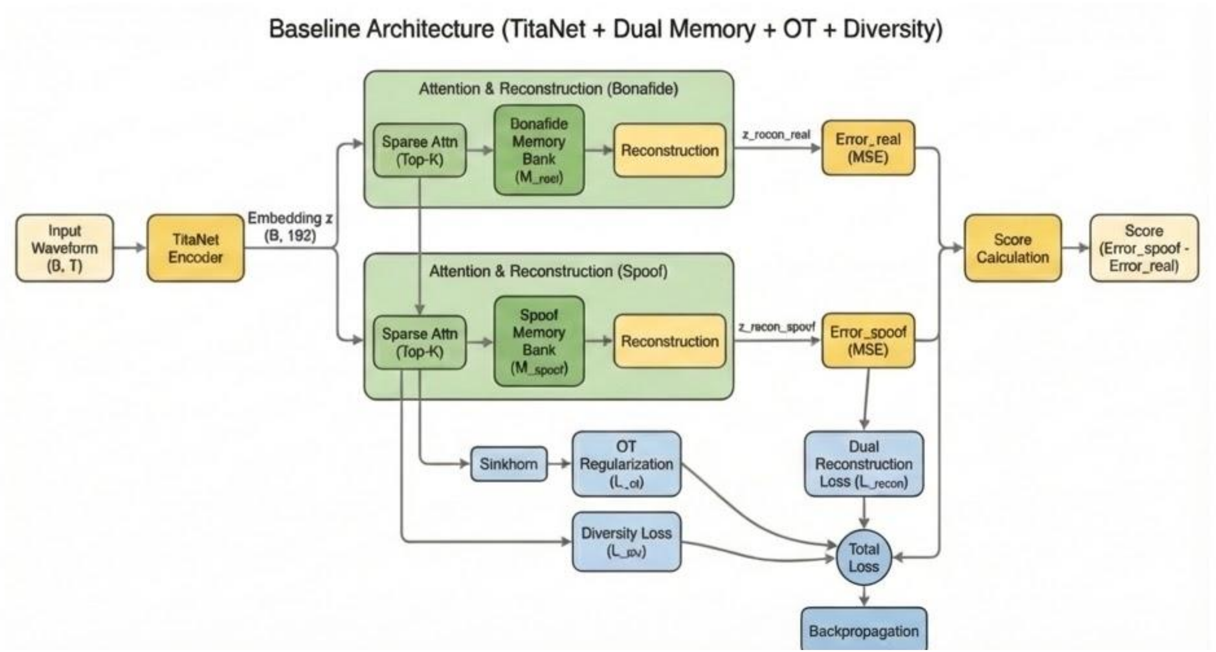
Related Works - Data Augmentation

- We apply two methods from RawBoost (Tak et al., 2021) to add noise to audio.



(Tak et al., 2021)

Framework


Baseline Architecture (TitaNet + Dual Memory + OT + Diversity)
Input Waveform (B, T)
TitaNet Encoder
Embedding z (B, 192)
Attention & Reconstruction (Bonafide)
Sparse Attn (Top-K)
Bonafide Memory Bank (M_real)
Reconstruction
z_recon_real
Error_real (MSE)
Attention & Reconstruction (Spoof)
Sparse Attn (Top-K)
Spoof Memory Bank (M_spoof)
Reconstruction
z_recon_spoof
Error_spoof (MSE)
Score Calculation
Score (Error_spoof - Error_real)
Sinkhorn
OT Regularization (L_ot)
Diversity Loss (L_div)
Dual Reconstruction Loss (L_recon)
Total Loss
Backpropagation

Methodology - Speaker Embedding Model

- We use Titanet (Koluguri et al., 2021) based on NVIDIA Nemo framework to extract speaker embedding.
- It focuses on global context, which means that it doesn't contain temporal information in the audio.

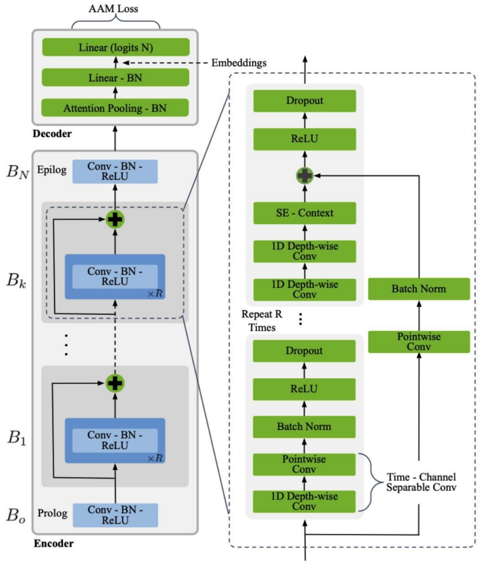


Fig. 1. TitaNet Encoder and Decoder Architecture

Methodology - Dual Memory Bank with Sparse Attention

- The idea is using top k memory vectors to do the reconstruction for bonafide and spoof audios separately (We et al. 2018).

$$M_{real} \in \mathbb{R}^{K \times D}, \quad M_{spoof} \in \mathbb{R}^{K \times D}$$

Algorithm 2 Memory Bank Initialization

Input: Number of slots K, Embedding dimension D
Output: Bonafide bank $\mathbf{M}_{real}$, Spoof bank $\mathbf{M}_{spoof}$

1: $\mathbf{M}_{real} \sim \mathcal{N}(0,1)^{K \times D}$
2: $\mathbf{M}_{spoof} \sim \mathcal{N}(0,1)^{K \times D}$
3: $\mathbf{M}_{real} \leftarrow \text{RowL2Normalize}(\mathbf{M}_{real})$
4: $\mathbf{M}_{spoof} \leftarrow \text{RowL2Normalize}(\mathbf{M}_{spoof})$
5: **return** $\mathbf{M}_{real}, \mathbf{M}_{spoof}$

Algorithm 3 Top-K Sparse Reconstruction

Input: Embedding $\mathbf{z} \in \mathbb{R}^{B \times D}$, Memory Bank $\mathbf{M} \in \mathbb{R}^{K \times D}$, Top-$k$ parameter k
Output: Reconstructed $\hat{\mathbf{z}}$, Reconstruction Error E, Similarity Matrix $\mathbf{S}$

1: $\hat{\mathbf{M}} \leftarrow \text{RowL2Normalize}(\mathbf{M})$
2: $\mathbf{S} \leftarrow \mathbf{z} \cdot \hat{\mathbf{M}}^\top$ ▷ Cosine Similarity
3: $\mathbf{V}_{top}, \mathbf{I}_{top} \leftarrow \text{TopK}(\mathbf{S}, k)$ ▷ Select top-k slots
4: $\mathbf{W} \leftarrow \text{Softmax}(\mathbf{V}_{top})$ ▷ Compute attention weights
5: $\mathbf{M}_{selected} \leftarrow \text{Gather}(\hat{\mathbf{M}}, \mathbf{I}_{top})$
6: $\hat{\mathbf{z}} \leftarrow \sum_{j=1}^{k} \mathbf{W}_{:,j} \cdot \mathbf{M}_{selected,:,j}$ ▷ Weighted Sum
7: $E \leftarrow \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2$ ▷ MSE Calculation
8: **return** $\hat{\mathbf{z}}, E, \mathbf{S}$

Methodology - Optimal Transport

- The idea is to fix the unbalanced distribution of the memory bank to avoid model collapse (Karon et al., 2020).

Algorithm 4 Sinkhorn-Knopp Algorithm (OT Regularization)

Input: Logits matrix $\mathbf{L} \in \mathbb{R}^{B \times K}$, Smoothing ϵ, Iterations T
Output: Optimal Assignment Matrix $\mathbf{Q}$

$\mathbf{Q} \leftarrow \exp(\mathbf{L}/\epsilon)$
for $t = 1$ to T **do**
  $\mathbf{Q} \leftarrow \mathbf{Q} \oslash (\mathbf{Q} \cdot \mathbf{1}_K \cdot \mathbf{1}_K^\top)$ ▷ Row Normalization
  $\mathbf{Q} \leftarrow \mathbf{Q} \oslash (\mathbf{1}_B \cdot \mathbf{1}_B^\top \cdot \mathbf{Q})$ ▷ Column Normalization
end for
$\mathbf{Q} \leftarrow \mathbf{Q} \oslash (\mathbf{Q} \cdot \mathbf{1}_K \cdot \mathbf{1}_K^\top)$ ▷ Final Row Norm
return $\mathbf{Q}$

Algorithm 6 OT Loss Computation

Input: Logits $\mathbf{L}$, Target Assignment $\mathbf{Q}$ (from Sinkhorn)
Output: Loss scalar $\mathcal{L}_{ot}$

$\mathbf{P} \leftarrow \text{LogSoftmax}(\mathbf{L})$
$\mathbf{Q}_{target} \leftarrow \text{Detach}(\mathbf{Q})$ ▷ Stop gradient for target
$\mathcal{L}_{ot} \leftarrow -\frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{K} \mathbf{Q}_{target,i,j} \cdot \mathbf{P}_{i,j}$
return $\mathcal{L}_{ot}$

Examples for SK-Algorithm

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 2 \end{bmatrix}$$

Examples for SK-Algorithm

$$A = \begin{bmatrix} 1 & 4 \\ 2 & 2 \end{bmatrix}$$

1st Row Normalization:

$$1 + 4 = 5 \rightarrow [1/5, 4/5] = [0.2, 0.8]$$

$$2 + 2 = 4 \rightarrow [2/4, 2/4] = [0.5, 0.5]$$

$$A_1 = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \end{bmatrix}$$

Examples for SK-Algorithm

$$A_1 = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \end{bmatrix}$$

1st Column Normalization:

$$0.2 + 0.5 = 0.7$$

$$0.8 + 0.5 = 1.3$$

$$A_2 = \begin{bmatrix} 0.2/0.7 & 0.8/1.3 \\ 0.5/0.7 & 0.5/1.3 \end{bmatrix} \approx \begin{bmatrix} 0.286 & 0.615 \\ 0.714 & 0.385 \end{bmatrix}$$

$\longrightarrow$ **After a few iterations (~3), rows and columns would be normalized**

Methodology - Reconstruction / Diversity Loss

Algorithm 5 Dual Reconstruction Loss

Input: Errors E_{real}, E_{spoof}, Ground Truth $y \in \{0, 1\}$, Margin m

$\mathcal{B} \leftarrow \{i \mid y_i = 0\}$ ▷ Indices of Bonafide

$\mathcal{S} \leftarrow \{i \mid y_i = 1\}$ ▷ Indices of Spoof

$\mathcal{L} \leftarrow 0$

if $|\mathcal{B}| > 0$ **then**

  $\mathcal{L} \leftarrow \mathcal{L} + \mathrm{Mean}(E_{real}[\mathcal{B}])$ ▷ Attract Real

  $\mathcal{L} \leftarrow \mathcal{L} + \mathrm{Mean}(\mathrm{ReLU}(m - E_{spoof}[\mathcal{B}]))$ ▷ Repel Spoof

end if

if $|\mathcal{S}| > 0$ **then**

  $\mathcal{L} \leftarrow \mathcal{L} + \mathrm{Mean}(E_{spoof}[\mathcal{S}])$ ▷ Attract Spoof

  $\mathcal{L} \leftarrow \mathcal{L} + \mathrm{Mean}(\mathrm{ReLU}(m - E_{real}[\mathcal{S}]))$ ▷ Repel Real

end if

return $\mathcal{L}$

Algorithm 10 Diversity Loss (Entropy Maximization)

Input: Attention Weights $\mathbf{W} \in \mathbb{R}^{B \times K}$

Output: Loss scalar $\mathcal{L}_{div}$

$\bar{\mathbf{w}} \leftarrow \frac{1}{B} \sum_{i=1}^{B} \mathbf{W}_{i,:}$ Compute batch-wise mean attention

$H \leftarrow -\sum_{j=1}^{K} \bar{\mathbf{w}}_j \cdot \log(\bar{\mathbf{w}}_j + \epsilon)$ Shannon Entropy of the mean distribution

$\mathcal{L}_{div} \leftarrow -H$ Minimize negative entropy $\Rightarrow$ Maximize diversity

return $\mathcal{L}_{div}$

Methodology (Additional): Contrastive Memory Loss

Algorithm 8 Contrastive Memory Loss

Input: Embedding $\mathbf{z}$, Memory $\mathbf{M}$, Labels y, Temp τ, Margin m

```
S ← (z · M^⊤)/τ
L_pull ← 0, L_push ← 0
if Bonafide samples exist then
    L_pull ← −Mean(LogSumExp(S[Bonafide]))
end if
if Spoof samples exist then
    L_push ← Mean(ReLU(max(S[Spoof]) + m))
end if
return L_pull + L_push
```

Methodology (Additional): Multi-Center OC-Softmax

Algorithm 7 Multi-Center OC-Softmax Loss

Input: Embeddings $\mathbf{z}$, Centers $\mathbf{C}$, Labels y, Margins m_{real}, m_{fake}, Scale α

$\mathbf{S} \leftarrow \text{L2Normalize}(\mathbf{z}) \cdot \text{L2Normalize}(\mathbf{C})^\top$

$s_{max} \leftarrow \max_j(\mathbf{S}_{:,j})$ ▷ Max similarity across centers

$\mathcal{L} \leftarrow 0$

for each sample i in batch **do**

  if $y_i = 0$ **then** ▷ Bonafide

    $\mathcal{L} \leftarrow \mathcal{L} + \text{Softplus}(\alpha(m_{real} - s_{max,i}))$

  else ▷ Spoof

    $\mathcal{L} \leftarrow \mathcal{L} + \text{Softplus}(\alpha(s_{max,i} - m_{fake}))$

  end if

end for

return Mean($\mathcal{L}$)

Methodology (Additional): Addaptive Margin

- The idea is to gradually change the adaptive margin for OC-Softmax.

Algorithm 9 Adaptive Margin Scheduler

Input: Current Step t, Warmup T_{warm}, Total Steps T_{total}
Output: Current margins m_{real}, m_{fake}

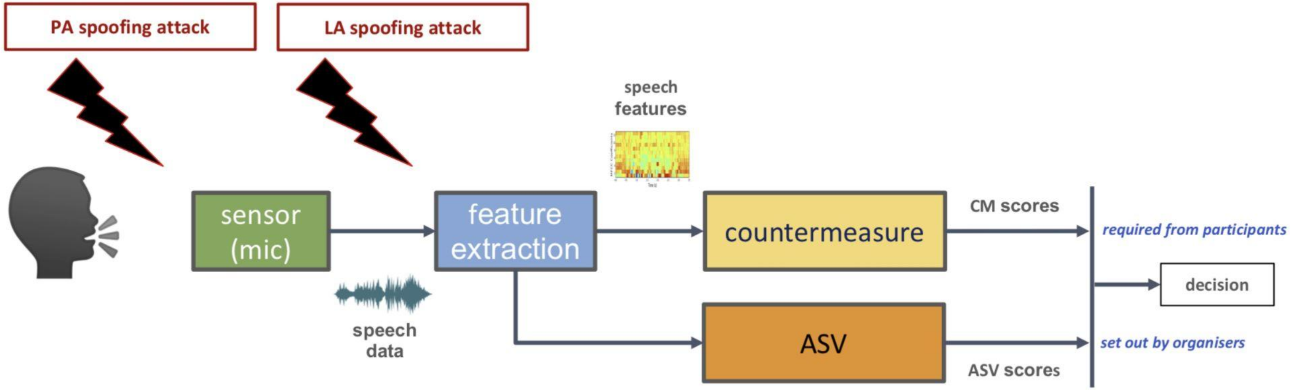
Hyperparams: $m_{real}^{start} = 0.7, m_{real}^{end} = 0.95$
Hyperparams: $m_{fake}^{start} = 0.3, m_{fake}^{end} = 0.1$

if $t < T_{warm}$ **then**
    $p \leftarrow 0$ Warmup phase
else
    $p \leftarrow \frac{t - T_{warm}}{T_{total} - T_{warm}}$
    $p \leftarrow \min(p, 1.0)$ Linear progress $[0, 1]$
end if

$m_{real} \leftarrow m_{real}^{start} + p \cdot (m_{real}^{end} - m_{real}^{start})$ Stricter constraints for Bonafide
$m_{fake} \leftarrow m_{fake}^{start} - p \cdot (m_{fake}^{start} - m_{fake}^{end})$ Lower tolerance for Spoof
return m_{real}, m_{fake}

Dataset

- Training and validation: ASVspoof 2019 (Wang et al., 2019)
- Evaluation: ASVspoof 2019 LA (Wang et al., 2019), ASVspoof 2021 LA (Delgado et al., 2021)



(Wang et al., 2019)

Experiements

- We choose best validation EER score to run the evaluation.
- We report the EER score and mini t-DCF score.
- Learning Rate: 1e-4, Weight decay: 2e-3, Steps: 5000 (~50 epochs)
- Lambda:
  - recon: 1.0
  - ot: 0.2
  - diversity: 0.1
  - oc (if apply): 0.5
- Sinkhorn iterations: 3
- Memory slots: 64
- Top K: 10

$$L_{total} = \lambda_{recon} L_{recon} + \lambda_{ot} L_{ot} + \lambda_{oc} L_{oc} + \lambda_{div} L_{div} + \lambda_{con} L_{con}$$

Results

Table 1. Experimental Results on ASVspoof 2019 and 2021

Method	EER (%)	min t-DCF	2021 EER (%)
Baseline (Recon + OT)	**1.37**	**0.0412**	11.03
+ OC-Softmax	1.52	0.0438	18.39
+ Multi-Center OC	3.47	0.0714	11.92
+ Contrastive Loss	2.87	0.0510	12.76
+ Large Model	5.26	0.1150	**9.90**
+ Adaptive Margin	3.22	0.0635	10.19
+ Score Fusion	2.47	0.0626	11.02
Larger Memory (128M, 20K)	1.90	0.0584	10.25
TitaNet + OC	1.70	0.0548	10.50

Results

Table 2. Comparison with State-of-the-Art on ASVspoof 2019 LA

Method	**Backbone**	**EER (%)**	**min t-DCF**
OC-Softmax	AASIST	1.25	0.0415
SAMO	AASIST	**1.08**	**0.0363**
RawNet2	RawNet2	2.48	-
Ours	TitaNet Small	1.37	0.0412

Results

Table 3. Experimental Results on ASVspoof 2021 LA dataset

Method	**2021 EER (%)**
RawNet2	9.50
AASIST	5.59
XLSR-Conformer	1.38
XLSR-Conformer + TCM	1.03
Ours (Titanet Large)	9.90

Conclusion

1. We have demonstrated the potential for speaker embedding models, yet it is not SOTA.
2. All enhanced modifications failed, maybe these methods are too complicate for the dataset.
3. General applications are bad, which implies overfitting in the training data.
4. Adding a decoder while freezing the encoder would be our future works.

Reference

- Koluguri, N. R., Park, T., Ginsburg, B. (2021) TitaNet: Neural Model for speaker representation with 1D Depth-wise separable convolutions and global context. *arXiv preprint arXiv:2110.04410*.
- Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A. (2020) Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 9912-9924.
- Wu, Z., Xiong, Y., Yu, S. X., Lin, D. (2018) Unsupervised Feature Learning via Non-Parametric Instance Discrimination. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3733-3742.
- Zhang, Y., Jiang, F., Duan, Z. (2021) One-Class Learning Towards Synthetic Voice Spoofing Detection. *IEEE Signal Processing Letters*, vol. 28, 937-941, doi: 10.1109/LSP.2021.3076358.
- Rosello, E., Gomez-Alanis, A., Gomez, A.M., Peinado, A. (2023) A conformer-based classifier for variable-length utterance processing in anti-spoofing. Proc. Interspeech 2023, 5281-5285, doi: 10.21437/Interspeech.2023-1820
- Yuen, K. C., Yip, J. Q., Qiu, Z., Chi, C. H., Lam, K. Y. (2025) Bona fide Cross Testing Reveals Weak Spot in Audio Deepfake Detection Systems. *CoRR*, vol. abs/2509.09204, doi: 10.48550/arXiv.2509.09204.
- Delgado, H., Evans, N., Kinnunen, T., Lee, K. A., Liu, X., Nautsch, A., Patino, J., Sahidullah, M., Todisco, M., Wang, X., and others. (2021) ASVspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan. *arXiv preprint arXiv:2109.00535*.

Reference

- Tak, H., Kamble, M., Patino, J., Todisco, M., Evans, N. (2022) RawBoost: A Raw Data Boosting and Augmentation Method applied to Automatic Speaker Verification Anti-Spoofing. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
- Ding, S., Zhang, Y., Duan, Z. (2023) SAMO: Speaker Attractor Multi-Center One-Class Learning for Voice Anti-Spoofing. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*
- Tak, H., Patino, J., Todisco, M., Nautsch, A., Evans, N., Larcher, A. (2021) End-to-end anti-spoofing with RawNet2. *arXiv preprint arXiv:2011.01108*.
- Zhang, Y., Jiang, F., Duan, Z. (2021) One-Class Learning Towards Synthetic Voice Spoofing Detection. *IEEE Signal Processing Letters*, vol. 28, 937-941, doi: 10.1109/LSP.2021.3076358.
- Jung, J. W., Heo, H. S., Tak, H., Shim, H. J., Chung, J. S., Lee, B. J., Yu, H. J., Evans, N. (2021) AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks. *arXiv preprint arXiv:2110.01200*.