icassp 2022

Singapore, China, Online

AUDIO INFORMATION RESEARCH

One-class Learning Towards Synthetic Voice Spoofing Detection

You Zhang, *Fei Jiang, Zhiyao Duan*

University of Rochester, USA

Y. Zhang, F. Jiang and Z. Duan, "One-Class Learning Towards Synthetic Voice Spoofing Detection," in *IEEE Signal Processing Letters*, vol. 28, pp. 937-941, 2021, doi: 10.1109/LSP.2021.3076358.

Outline

Background

Method

Experiments

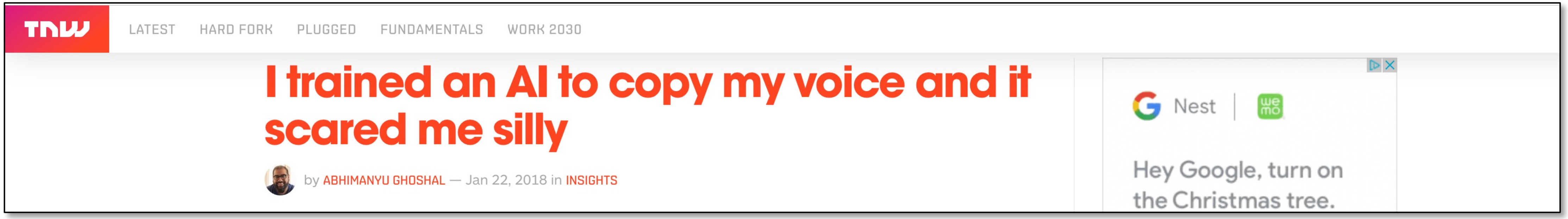
Conclusion


Synced

AI TECHNOLOGY

Clone a Voice in Five Seconds With This AI Toolbox

A new Github project introduces a remarkable Real-Time Voice Cloning Toolbox that enables anyone to clone a voice from as little as five seconds of sample audio.


TNW LATEST HARD FORK PLUGGED FUNDAMENTALS WORK 2030

I trained an AI to copy my voice and it scared me silly

by ABHIMANYU GHOSHAL — Jan 22, 2018 in INSIGHTS


THE WALL STREET JOURNAL.

Subscribe | Sign In

Special Offer

English Edition | Print Edition | Video | Podcasts | Latest Headlines

Home World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate WSJ. Magazine Search

SHARE

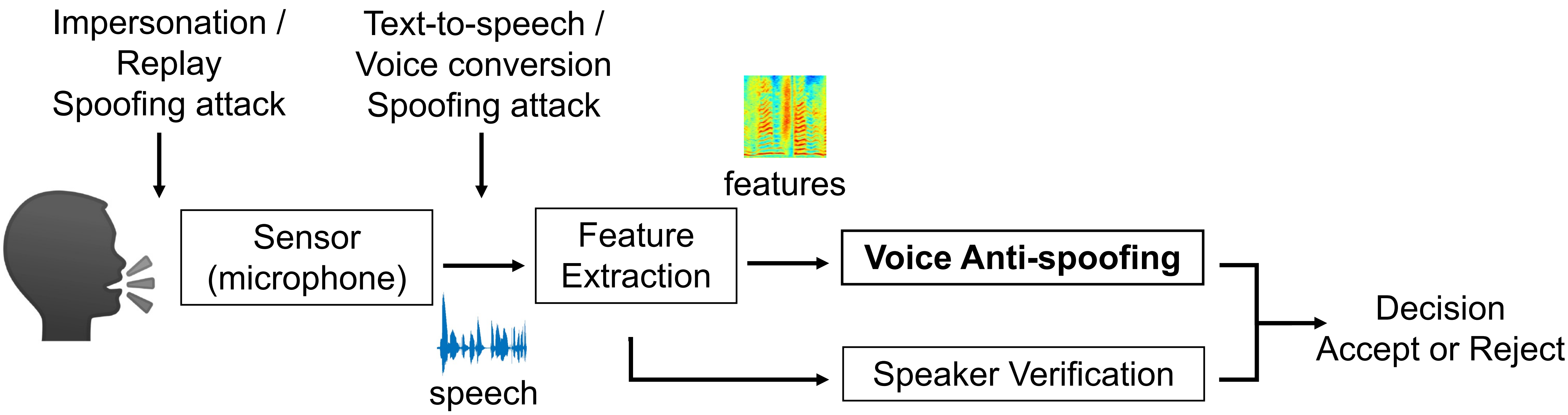
PRO CYBER NEWS

Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case

Scams using artificial intelligence are a new challenge for companies

Anti-spoofing

- Spoofing Countermeasure: Detect spoofing attacks



Research question

AUDIO INFORMATION RESEARCH

Motivation:

- The **fast development** of speech **synthesis** are posing increasingly more threat.
- The **distribution mismatch** between the training set and test set for the **spoofing** attacks class.

➢ How can the anti-spoofing system defend **against unseen** spoofing attacks?

(Generalization ability)

Outline

Introduction

Method

Experiments

Conclusion

Definition of one-class

- "In **one-class classification**, one of the classes (referred to as the positive class or target class) is well characterized by instances in the training data. For the other class (nontarget), it has either no instances at all, very few of them, or they do not form a statistically-representative sample of the negative concept."

Khan, Shehroz S., and Michael G. Madden. "A survey of recent trends in one class classification." *Irish Conference on Artificial Intelligence and Cognitive Science*. Springer, Berlin, Heidelberg, 2009.

One-class learning (OC-Softmax)

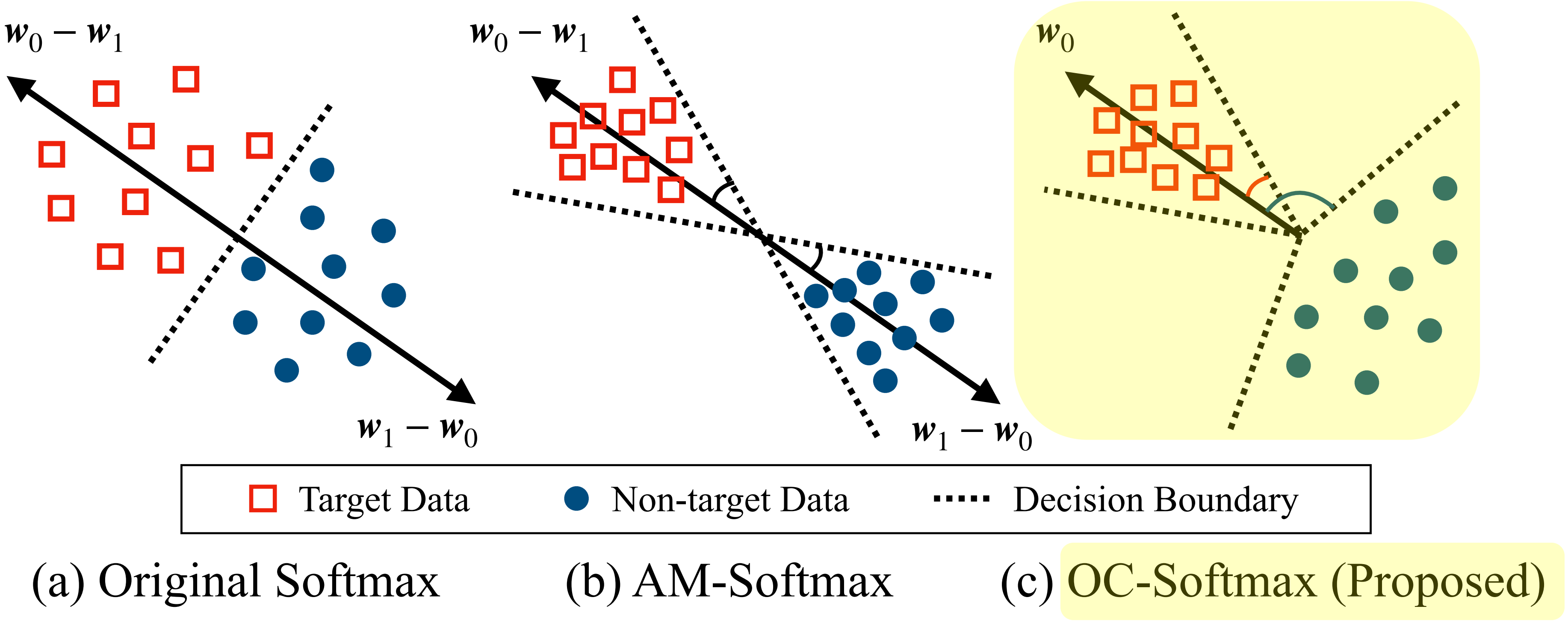


Fig. 1. Illustration of the original Softmax and AM-Softmax for binary classification, and our proposed OC-Softmax for one-class learning. (The embeddings and the weight vectors shown are non-normalized).

One-Class Softmax (Proposed)

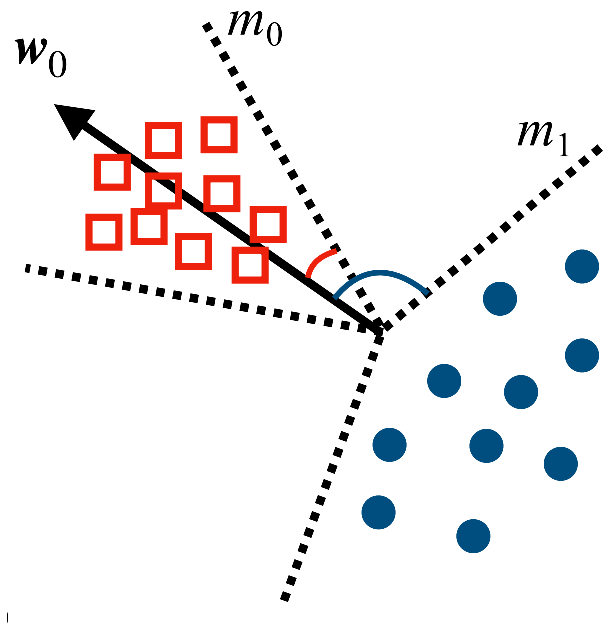
- Training (Loss):

scale factor center vector label

$$\mathcal{L}_{OCS} = \frac{1}{N}\sum_{i=1}^{N}\log\left(1+e^{\alpha(m_{y_i}-\hat{\boldsymbol{w}}_0\hat{\boldsymbol{x}}_i)(-1)^{y_i}}\right).$$

margin embedding

samples



- Inference (Score):

$$\mathcal{S}_{OCS} = \hat{\boldsymbol{w}}_0\hat{\boldsymbol{x}}_i.$$

Comparing loss

- Softmax

$$\mathcal{L}_S = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\boldsymbol{w}_{y_i}^T\boldsymbol{x}_i}}{e^{\boldsymbol{w}_{y_i}^T\boldsymbol{x}_i}+e^{\boldsymbol{w}_{1-y_i}^T\boldsymbol{x}_i}}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\log\left(1+e^{(\boldsymbol{w}_{1-y_i}-\boldsymbol{w}_{y_i})^T\boldsymbol{x}_i}\right),$$

- AM-Softmax

$$\mathcal{L}_{AMS} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\alpha(\hat{\boldsymbol{w}}_{y_i}^T\hat{\boldsymbol{x}}_i-m)}}{e^{\alpha(\hat{\boldsymbol{w}}_{y_i}^T\hat{\boldsymbol{x}}_i-m)}+e^{\alpha\hat{\boldsymbol{w}}_{1-y_i}^T\hat{\boldsymbol{x}}_i}}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\log\left(1+e^{\alpha\left(m-(\hat{\boldsymbol{w}}_{y_i}-\hat{\boldsymbol{w}}_{1-y_i})^T\hat{\boldsymbol{x}}_i\right)}\right),$$

- **OC-Softmax**

$$\mathcal{L}_{OCS} = \frac{1}{N}\sum_{i=1}^{N}\log\left(1+e^{\alpha(m_{y_i}-\hat{\boldsymbol{w}}_0\hat{\boldsymbol{x}}_i)(-1)^{y_i}}\right).$$

Outline

Introduction

Method

Experiments

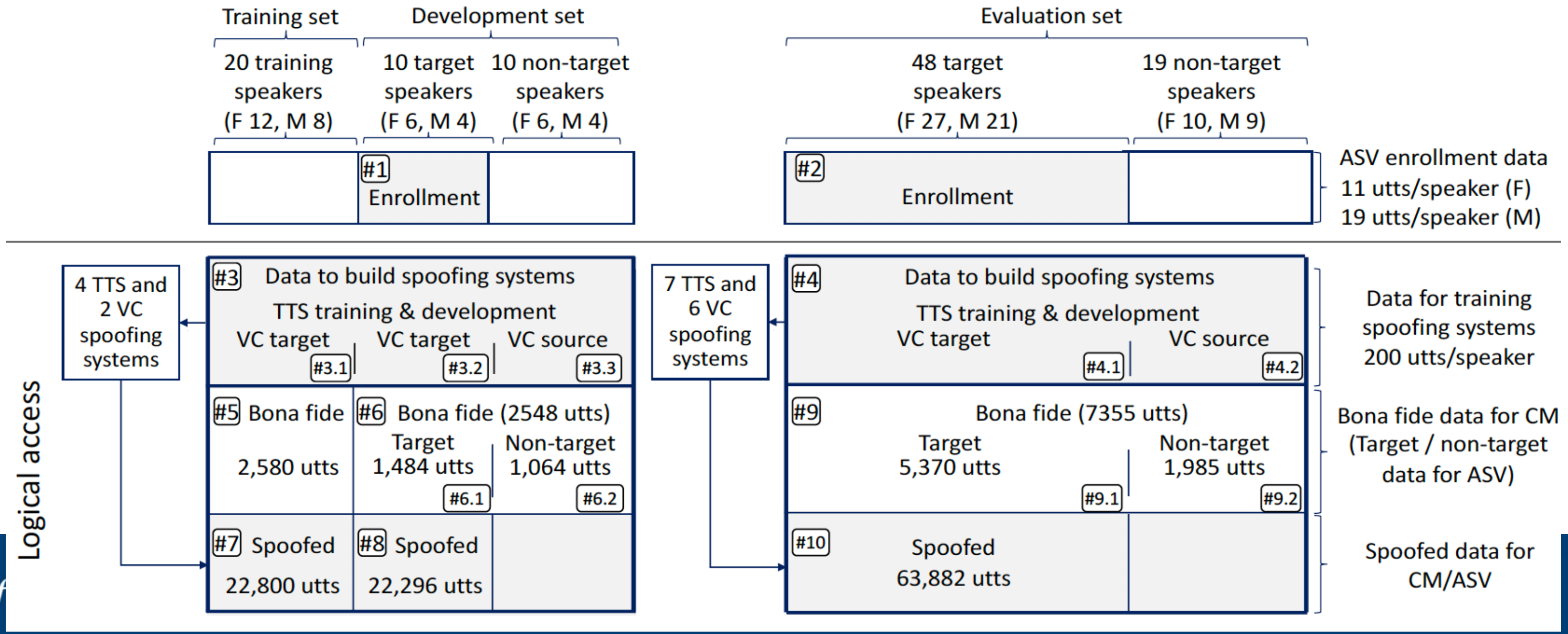
Conclusion

Dataset

ASVspoof 2019 Logical Access (TTS + VC)

- Bona fide speech (VCTK dataset)
- 6 known attacks (appear in training set)
- 13 unknown attacks (only appear in eval set)

	Bona fide	Spoofed	
	# utterance	# utterance	attacks
Training	2,580	22,800	A01 - A06
Development	2,548	22,296	A01 - A06
Evaluation	7,533	63,882	A07 - A19

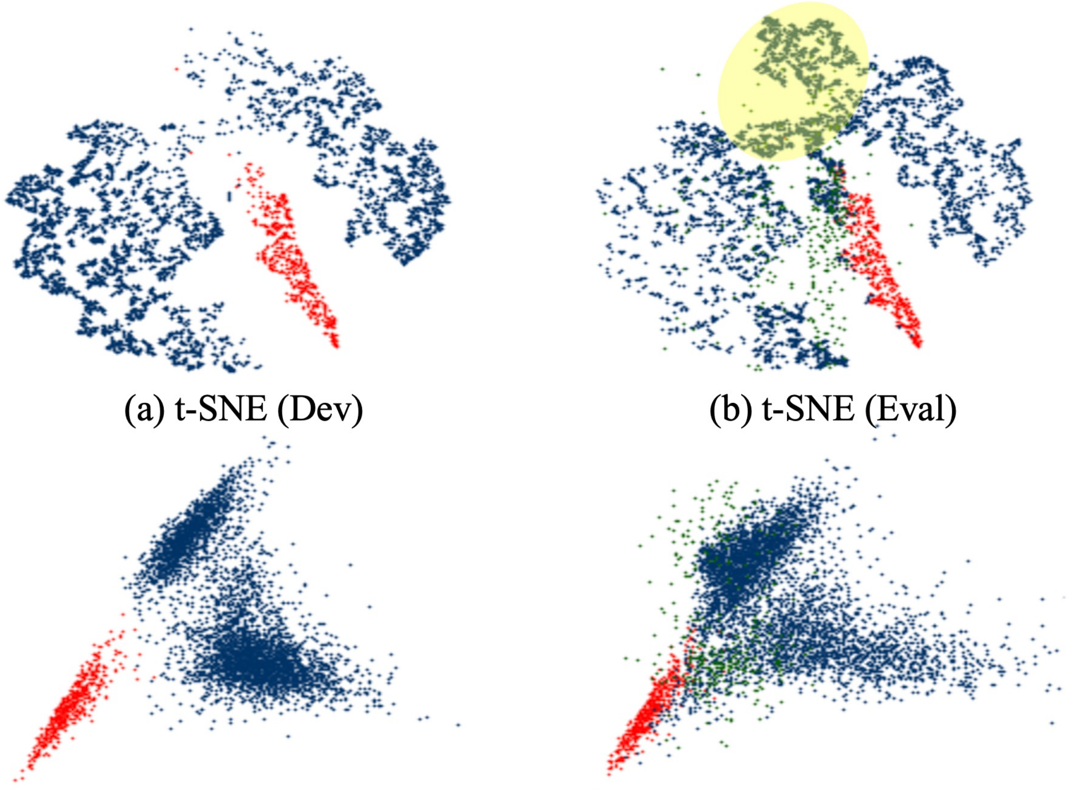


Evaluation of OC-Softmax

- Results on the development and evaluation sets of the ASVspoof 2019 LA scenario using different losses

Loss	Dev Set		Eval Set	
	EER (%)	t-DCF	EER (%)	t-DCF
Softmax	0.35	0.010	4.69	0.125
AM-Softmax	0.43	0.013	3.26	0.082
Proposed	0.20	0.006	**2.19**	**0.059**

- OC-Softmax performs the best on unseen attacks.



(a) t-SNE (Dev) (b) t-SNE (Eval) (c) PCA (Dev) (d) PCA (Eval)

Feature Embedding Visualization

Comparison with single systems

System	EER (%)	min t-DCF
CQCC + GMM [3]	9.57	0.237
LFCC + GMM [3]	8.09	0.212
Chettri et al. [22]	7.66	0.179
Monterio et al. [14]	6.38	0.142
Gomez-Alanis et al. [16]	6.28	-
Aravind et al. [18]	5.32	0.151
Lavrentyeva et al. [21]	4.53	0.103
ResNet + OC-SVM	4.44	0.115
Wu et al. [17]	4.07	0.102
Tak et al. [19]	3.50	0.090
Chen et al. [15]	3.49	0.092
Proposed	**2.19**	**0.059**

Results in the leader board

Ours	0.059	2.19

ASVspoof 2019 LA scenario							
#	ID	t-DCF	EER	#	ID	t-DCF	EER
1	**T05**	0.0069	0.22	26	T57	0.2059	10.65
2	**T45**	0.0510	1.86	27	**T42**	0.2080	8.01
3	**T60**	0.0755	2.64	28	*B02*	0.2116	8.09
4	**T24**	0.0953	3.45	29	**T17**	0.2129	7.63
5	**T50**	0.1118	3.56	30	**T23**	0.2180	8.27
6	**T41**	0.1131	4.50	31	**T53**	0.2252	8.20
7	**T39**	0.1203	7.42	32	**T59**	0.2298	7.95
8	**T32**	0.1239	4.92	33	*B01*	0.2366	9.57
9	**T58**	0.1333	6.14	34	T52	0.2366	9.25
10	T04	0.1404	5.74	35	**T40**	0.2417	8.82
11	**T01**	0.1409	6.01	36	T55	0.2681	10.88
12	**T22**	0.1545	6.20	37	**T43**	0.2720	13.35
13	T02	0.1552	6.34	38	T31	0.2788	15.11
14	**T44**	0.1554	6.70	39	**T25**	0.3025	23.21
15	**T16**	0.1569	6.02	40	**T26**	0.3036	15.09
16	T08	0.1583	6.38	41	T47	0.3049	18.34
17	**T62**	0.1628	6.74	42	T46	0.3214	12.59
18	**T27**	0.1648	6.84	43	T21	0.3393	19.01
19	**T29**	0.1677	6.76	44	T61	0.3437	15.66
20	**T13**	0.1778	6.57	45	**T11**	0.3742	18.15
21	**T48**	0.1791	9.08	46	**T56**	0.3856	15.32
22	**T10**	0.1829	6.81	47	T12	0.4088	18.27
23	T54	0.1852	7.71	48	T14	0.4143	20.60
24	T38	0.1940	7.51	49	T20	1.0000	92.36
25	T33	0.1960	8.93	50	T30	1.0000	49.60

- Could rank between the 2nd and 3rd
- Top systems all use model fusion, but we do not

Outline

Background

Method

Experiments

Conclusion

Takeaways

- One-class learning aims to **compact the target** class representation in the embedding space, set a tight classification boundary around it and **push away non-target**.

- The proposed OC-Softmax could improve the **generalization ability** of anti-spoofing system against **unseen spoofing attacks**.

Follow-up works

- Channel Robustness
  - **You Zhang**, Ge Zhu, Fei Jiang, and Zhiyao Duan, "An Empirical Study on Channel Effects for Synthetic Voice Spoofing Countermeasure Systems", in *Proc. Interspeech*, pp. 4309-4313, 2021. [link][code][video]
  - Xinhui Chen*, **You Zhang***, Ge Zhu*, and Zhiyao Duan, "UR Channel-Robust Synthetic Speech Detection System for ASVspoof 2021", in *Proc. ASVspoof 2021 Workshop*, pp. 75-82, 2021. (* equal contribution) [link][code][video]
- Joint Optimization with ASV
  - **You Zhang**, Ge Zhu, and Zhiyao Duan, "A Probabilistic Fusion Framework for Spoofing Aware Speaker Verification", in *Proc. Odyssey*, 2022. [link][code]

Future directions

- Defend against diversified spoofing attacks
  - TTS+VC, replay
  - Partially spoofed
  - Adversarial attack
- Explainable anti-spoofing
  - Understanding the difference between synthetic vs. natural speech
- Visually-informed anti-spoofing
  - Deepfake detection, multimedia forensics

Thank you !

Q & A

Resources

Full Paper

Code

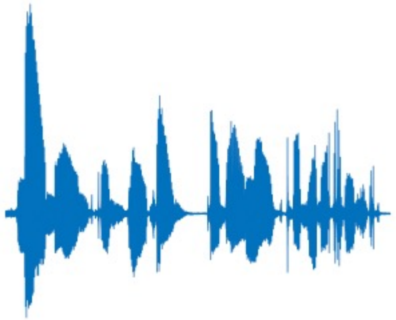
Poster

Takeaways

- One-class learning aims to **compact the target** class representation in the embedding space, set a tight classification boundary around it and **push away non-target**.
- One-class learning could improve the **generalization ability** of anti-spoofing system against **unknown spoofing attacks**.

Voice Biometrics

- Speaker Verification: Verify the identity of a speaker



Bona fide speech → Automatic Speaker Verification (ASV) → Decision Accept or Reject

Spoofing attacks

- **Impersonation**

    -- twins and professional mimics, no database available

- **Replay**

    -- reuse pre-recorded audio, most accessible

- **Text-to-speech (TTS)**

    -- convert written text into spoken words with speech synthesis

- **Voice conversion (VC)**

    -- convert speech from source speaker to target speaker’s voice

ASVspoof Challenge

AUDIO INFORMATION RESEARCH

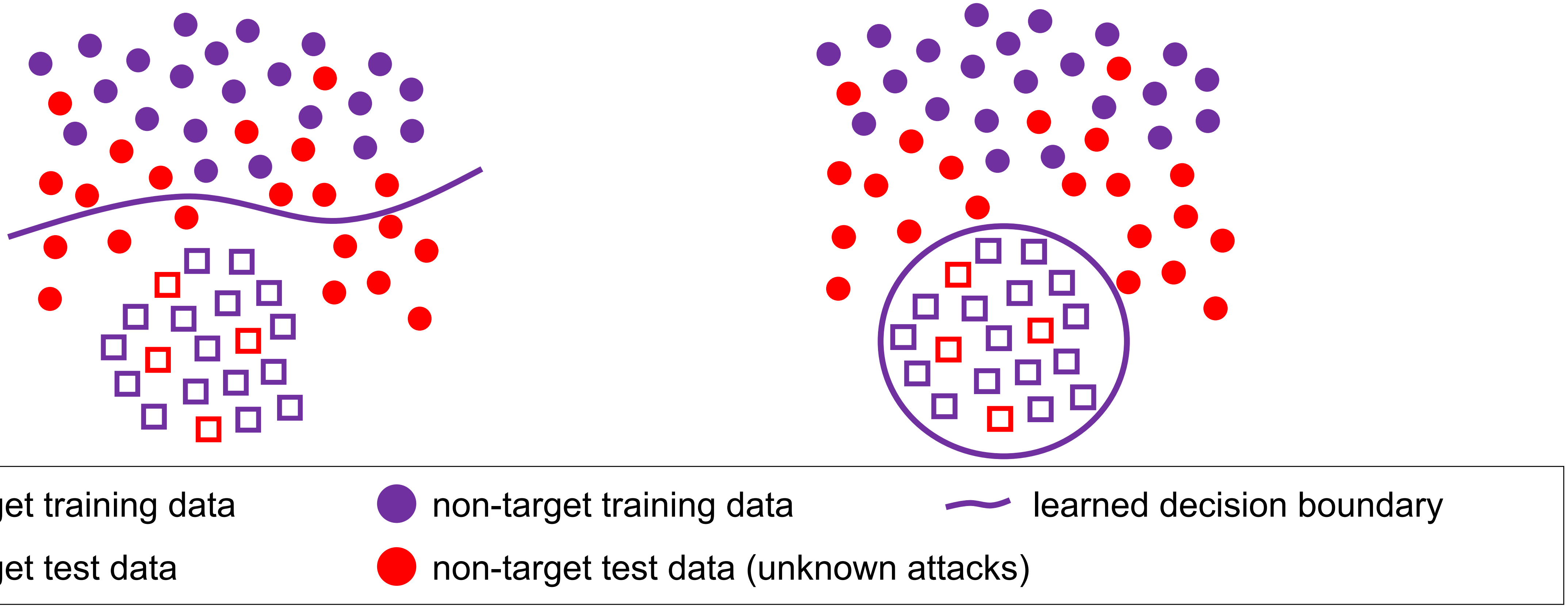
- Logical access (LA) $\left\{ \begin{array}{l} \text{Text-to-speech (TTS)} \\ \text{Voice conversion (VC)} \\ \text{TTS+VC} \end{array} \right.$

  -- algorithm-related artifacts ★ our current focus

- Physical access (PA) -- pre-recorded, replay

  -- device-related artifacts

Binary versus One-Class Classification



(a) Binary classification (b) One-class classification

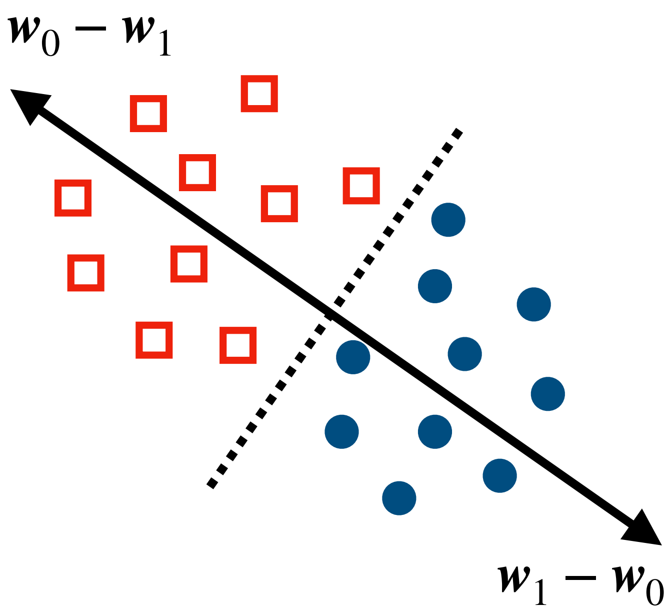
Softmax

- Training (Loss):

$$\mathcal{L}_S = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\boldsymbol{w}_{y_i}^T\boldsymbol{x}_i}}{e^{\boldsymbol{w}_{y_i}^T\boldsymbol{x}_i}+e^{\boldsymbol{w}_{1-y_i}^T\boldsymbol{x}_i}}$$
$$= \frac{1}{N}\sum_{i=1}^{N}\log\left(1+e^{(\boldsymbol{w}_{1-y_i}-\boldsymbol{w}_{y_i})^T\boldsymbol{x}_i}\right),$$

- Inference (Score):

$$\mathcal{S}_S = \frac{e^{\boldsymbol{w}_0^T\boldsymbol{x}_i}}{e^{\boldsymbol{w}_0^T\boldsymbol{x}_i}+e^{\boldsymbol{w}_1^T\boldsymbol{x}_i}}.$$



Additive Margin Softmax

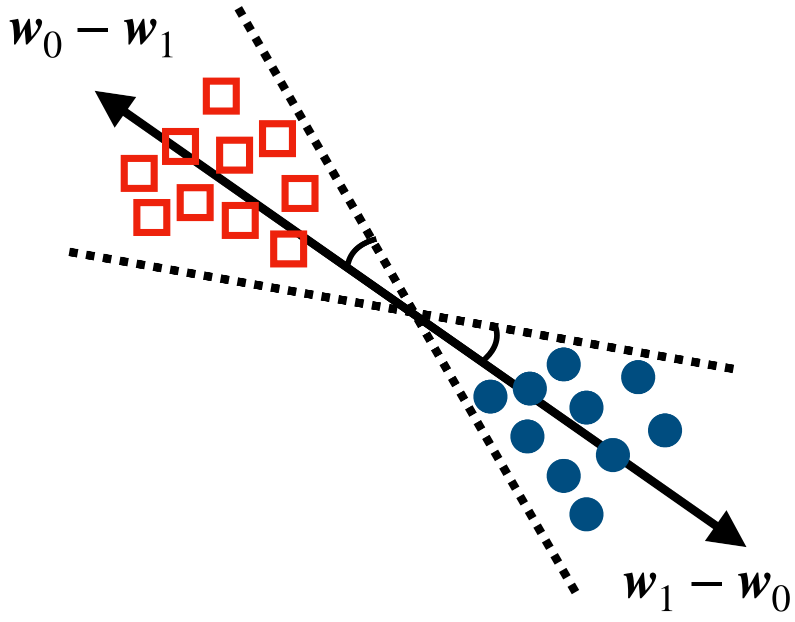
- Training (Loss):

$$\mathcal{L}_{AMS} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\alpha(\hat{\boldsymbol{w}}_{y_i}^T\hat{\boldsymbol{x}}_i - m)}}{e^{\alpha(\hat{\boldsymbol{w}}_{y_i}^T\hat{\boldsymbol{x}}_i - m)} + e^{\alpha\hat{\boldsymbol{w}}_{1-y_i}^T\hat{\boldsymbol{x}}_i}}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\log\left(1 + e^{\alpha\left(m - (\hat{\boldsymbol{w}}_{y_i} - \hat{\boldsymbol{w}}_{1-y_i})^T\hat{\boldsymbol{x}}_i\right)}\right),$$

- Inference (Score):

$$\mathcal{S}_{AMS} = (\hat{\boldsymbol{w}}_0 - \hat{\boldsymbol{w}}_1)^T\hat{\boldsymbol{x}}_i.$$



OC-Softmax output as probability

$$
\begin{aligned}
L_{OCS} &= \frac{1}{N}\sum_{i=1}^{N}\log\left(1+e^{\alpha\left(m_{y_i}-\hat{\mathbf{w}}^T\hat{\mathbf{x}}_i\right)(-1)^{y_i}}\right)\\
&= \frac{1}{N}\left(\sum_{|\Omega|}\log\left(1+e^{\alpha\left(m_0-\hat{\mathbf{w}}^T\hat{\mathbf{x}}_i\right)}\right)+\sum_{|\overline{\Omega}|}\log\left(1+e^{\alpha\left(\hat{\mathbf{w}}^T\hat{\mathbf{x}}_i-m_1\right)}\right)\right)\\
&= -\frac{1}{N}\left(\sum_{|\Omega|}\log\frac{1}{1+e^{\alpha\left(m_0-\hat{\mathbf{w}}^T\hat{\mathbf{x}}_i\right)}}+\sum_{|\overline{\Omega}|}\log\frac{1}{1+e^{\alpha\left(\hat{\mathbf{w}}^T\hat{\mathbf{x}}_i-m_1\right)}}\right)
\end{aligned}
$$