Advances in Audio Anti-spoofing and Deepfake Detection using Graph Neural Networks and Self-supervised Learning




Jee-weon Jung, Hye-jin Shim, Hemlata Tak, Xin Wang

August 20, 2023

Speakers

Jee-weon Jung

Postdoctoral research associate, CMU, USA

Speech anti-spoofing, speaker recognition, audio/speech tasks

Hye-jin Shim

Postdoctoral research associate, UEF, Finland

Speech anti-spoofing, speaker recognition, audio/speech tasks

Hemlata Tak

PhD graduate EURECOM, France

Speech anti-spoofing and deepfake detection

https://takhemlata.github.io/

Xin Wang

Project assistant professor, NII, Japan

Speech synthesis, speech anti-spoofing, speech privacy

https://researchmap.jp/wangxin?lang=en

Outline

- Introduction to speech anti-spoofing
- Introduction to graph attention networks
- Graph attention networks for speech anti-spoofing
- Self-supervised learning for speech anti-spoofing

Session 1

Session 2





Hands-on code will be available at:

https://github.com/Jungjee/INTERSPEECH2023_T6

Session 1

- **Introduction to speech anti-spoofing**
  - Background
  - Task definition
  - Evaluation metrics
  - Common approach
  - Toy example

→ Colab Jupyter notebook 

→ Colab Jupyter notebook 

- Introduction to graph attention networks
- Graph attention networks for speech anti-spoofing

Background

Human user

smart device, digital human

Speech synthesis

voice authentication

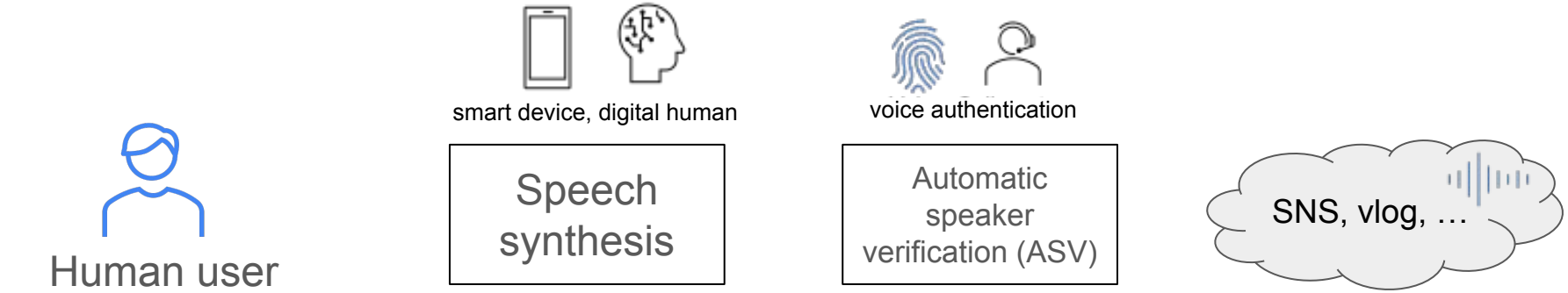
Automatic speaker verification (ASV)

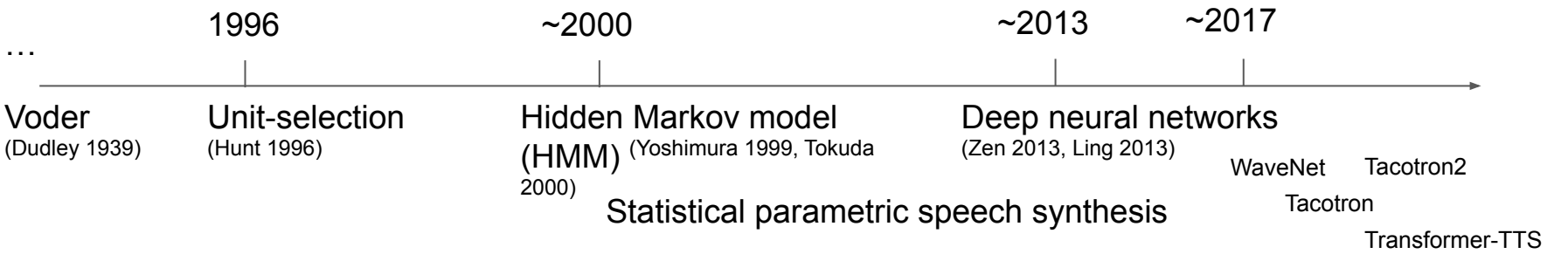
SNS, vlog, …

Live conversation in game by Nvidia

ASV solution by Nuance

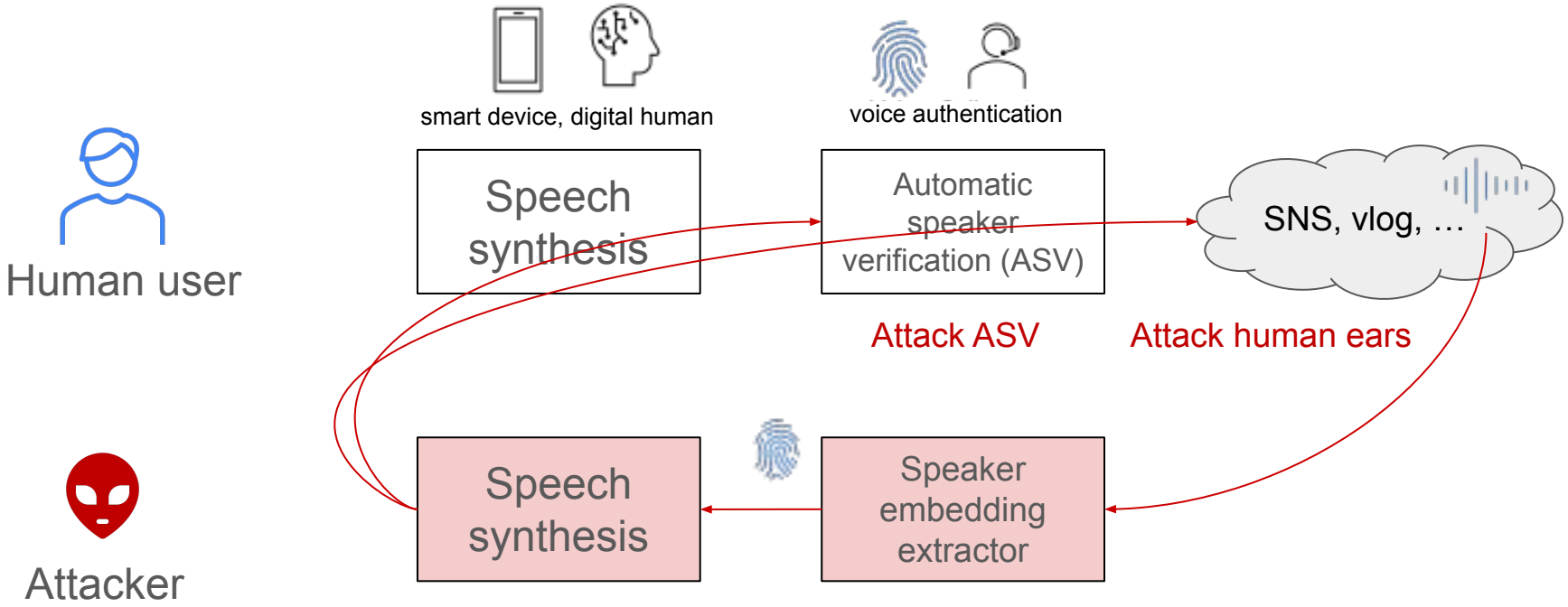
Background





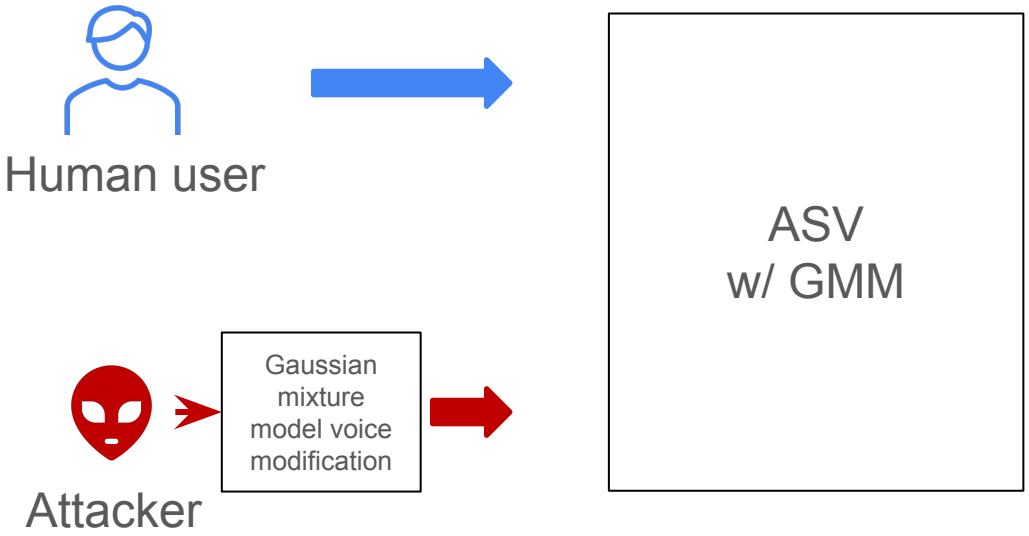
Not all relevant models are listed. See more in Reference section

Background


smart device, digital human
voice authentication
Human user
Speech synthesis
Automatic speaker verification (ASV)
SNS, vlog, …
Attack ASV
Attack human ears
Attacker
Speech synthesis
Speaker embedding extractor

Background

Attack ASV system (Pellom 1999)



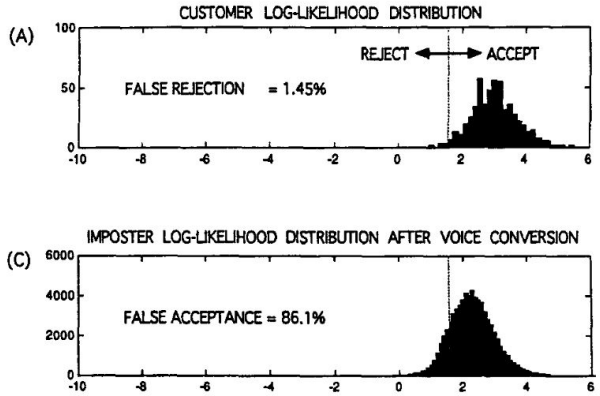
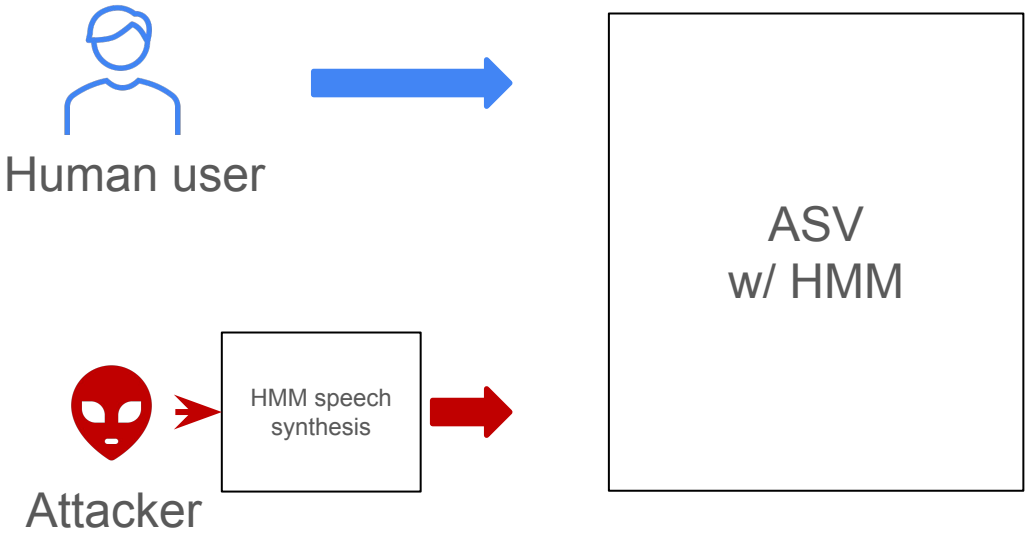


Figure 2: Histogram plot of log–likelihood ratio scores, $\Lambda(\mathbf{O})$, for (A) hypothesis $\mathcal{H}1$: customer access, (B) hypothesis $\mathcal{H}0$: casual imposter attempts, and for (C) hypothesis $\mathcal{H}0$: voice–altered imposter attempts.

86% of synthetic voice data will be accepted as the target speaker's voice

Background

Attack ASV system (Masuko 2000, Chen 2010, De Leon 2012)



In this paper, we have evaluated the vulnerability of SV to imposture using synthetic speech …

we have shown that with state-of-the-art speech synthesis, ***over 81% of matched claims, i.e. a synthetic speech signal matched to a targeted speaker*** *(De Leon 2012)*

Background

Attack ASV system (Wang 2020)

Latest ASV systems may be fooled by old and new speech synthesis systems

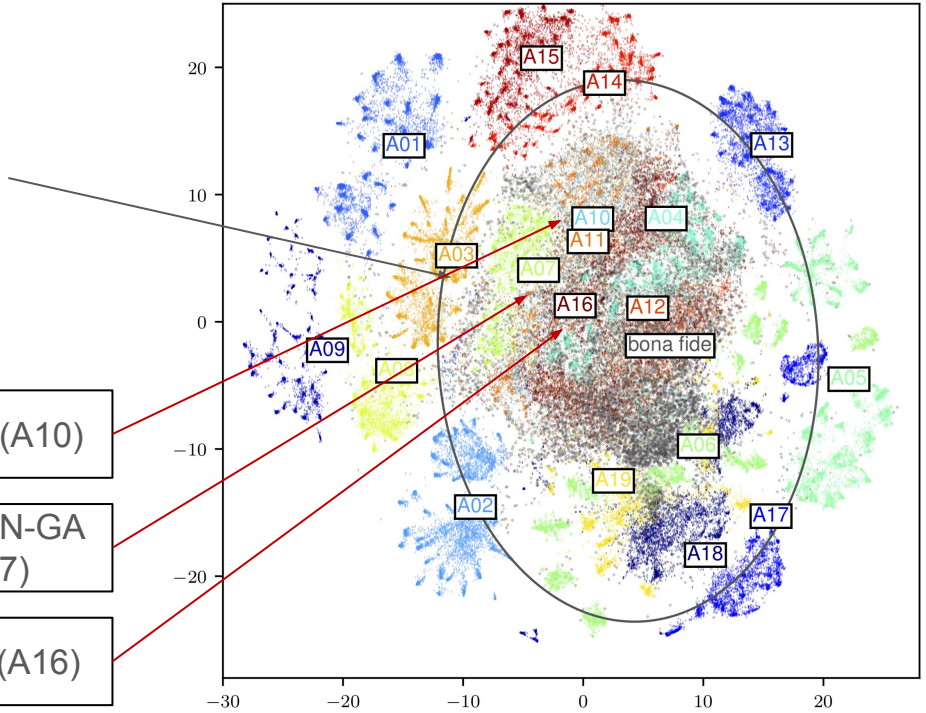
Human user

Attacker

Tacotron (A10)

HMM-RNN-GAN (A07)

Unit-sel. (A16)



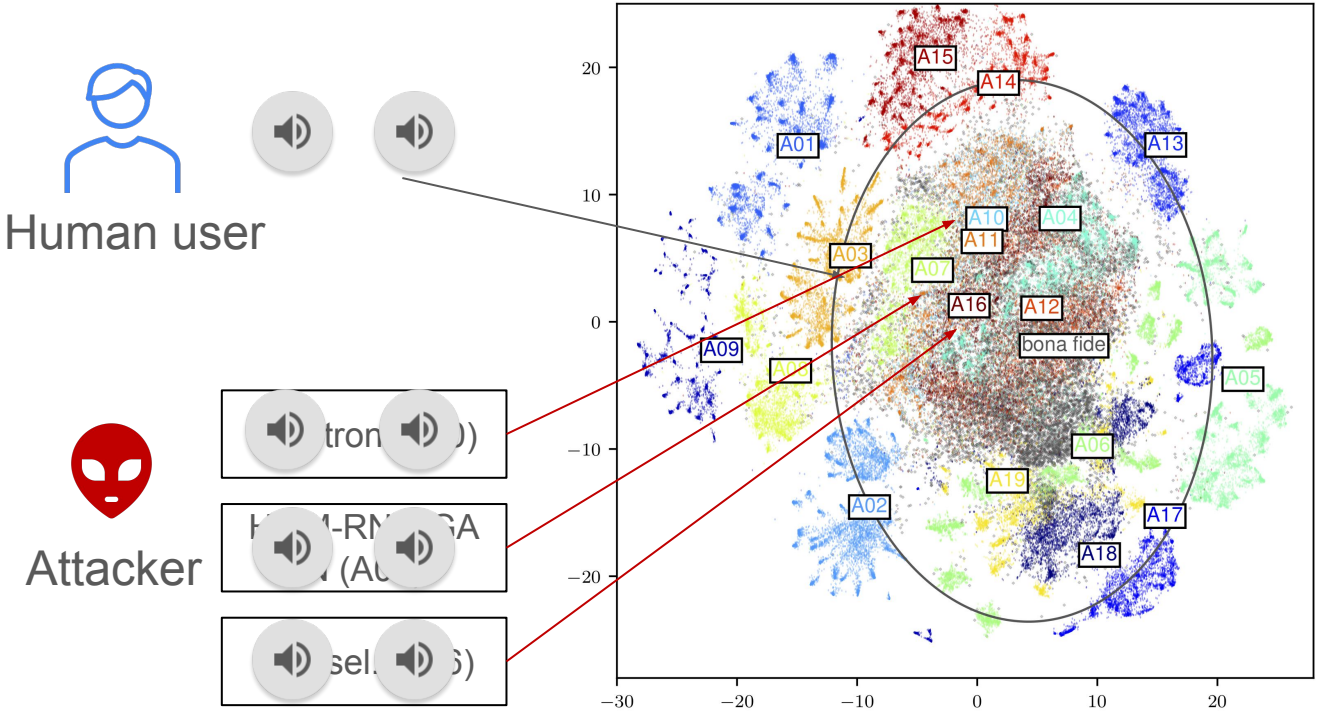
X-vector in 2D space
(Wang 2020)

Data from ASVspoof 2019 LA database

Background

Attack ASV system (Wang 2020)

Latest ASV systems may be fooled by old and new speech synthesis systems



X-vector in 2D space
(Wang 2020)

Data from ASVspoof 2019 LA database

Background

Fool human ears: Human ears may be fooled by latest speech synthesis systems

Do the synthetic and natural samples have the same quality?

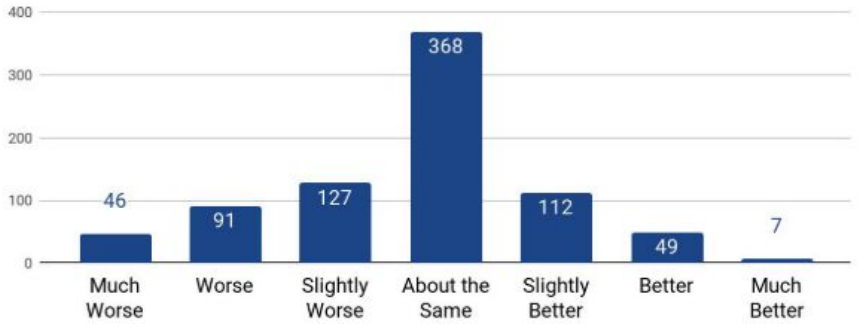


Fig. 2. Synthesized vs. ground truth: 800 ratings on 100 items.

Results from Tacotron 2 (Shen 2018)

Background

Unit-selection

Statistical parametric speech synthesis

MaryTTS

HMM-post-filtering

LSTM-GAN

Tacotron-WaveRNN

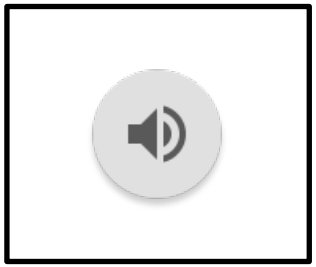
Target

Samples from https://nii-yamagishilab.github.io/samples-xin/main-asvspoof2019

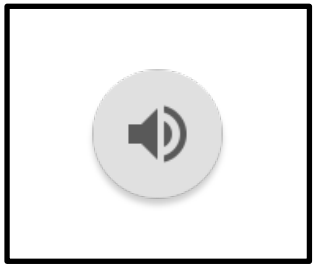
Background

There are **three** synthetic and **one** human voices. Which is the human voice?

A



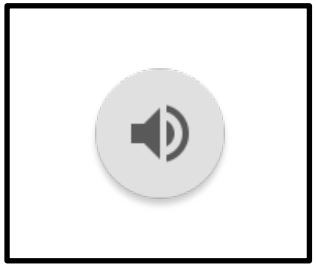
B



Test 1

C

D

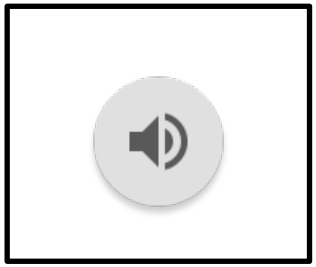


Background

There are **three** synthetic and **one** human voices. Which is the human voice?

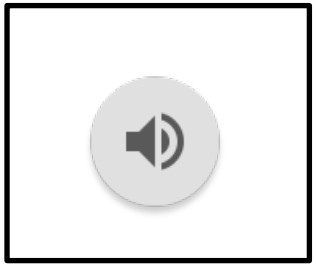
A

B

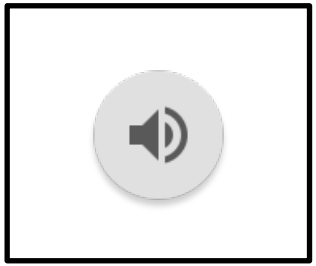


Test 2

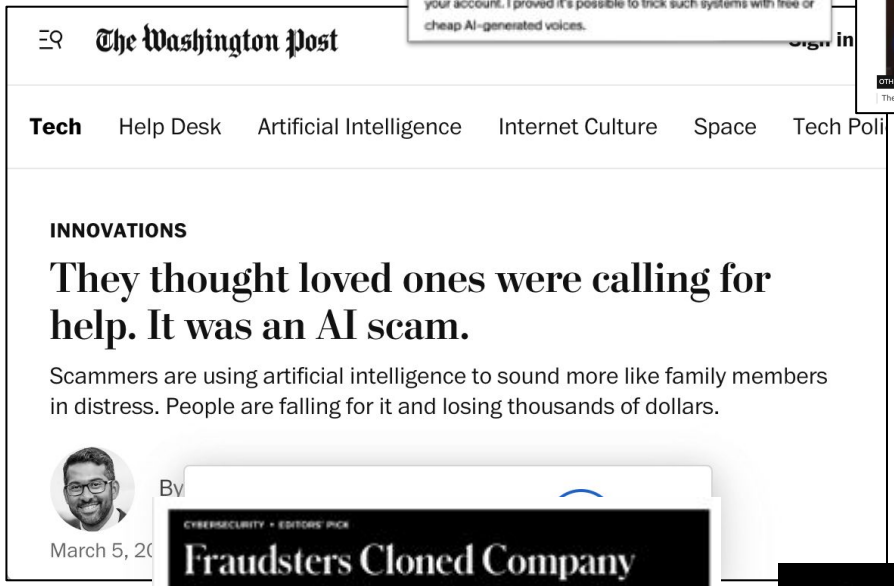
C

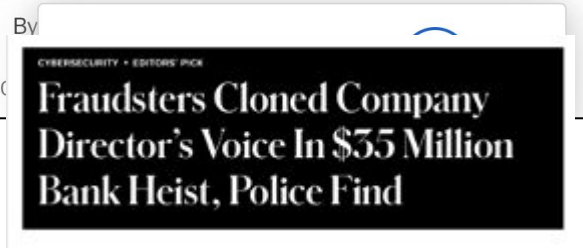


D



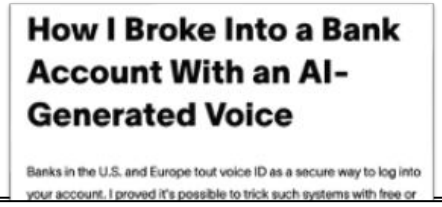
Background

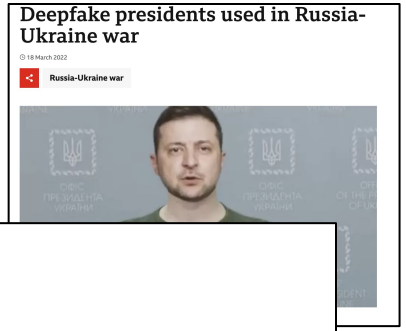


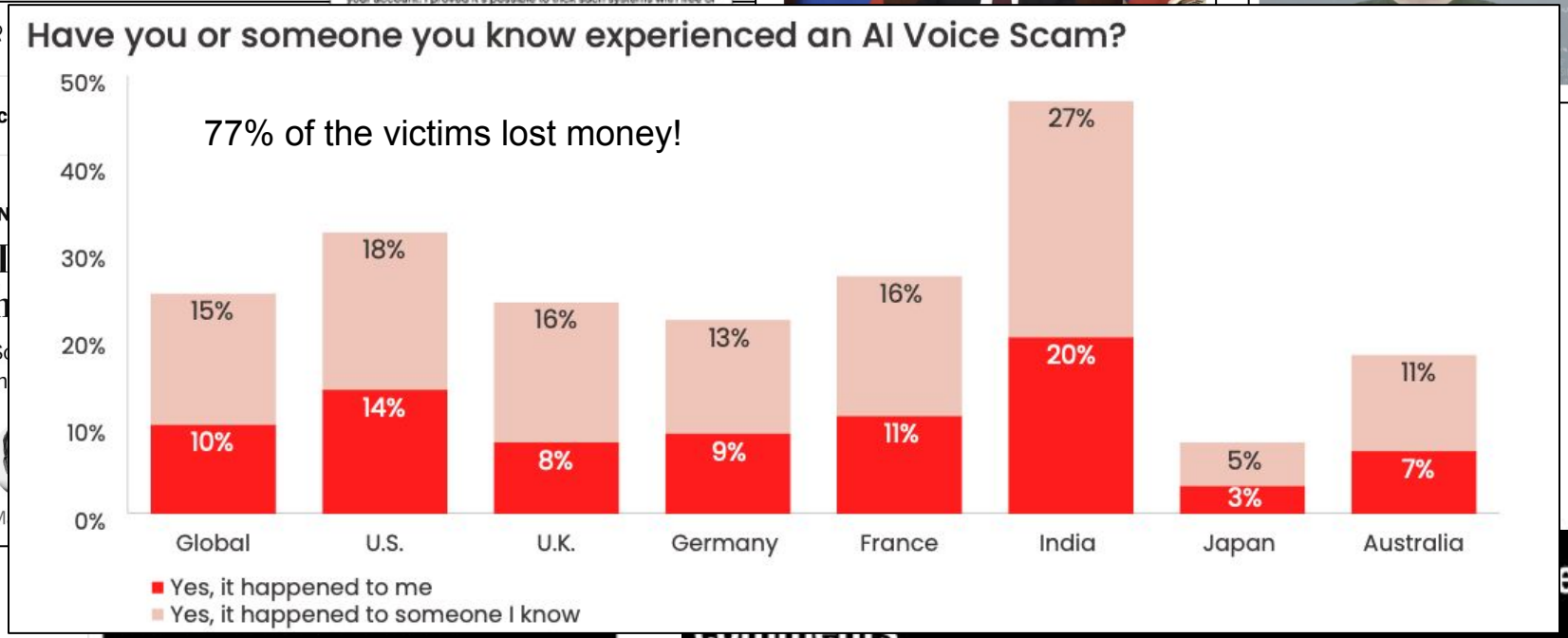


https://www.washingtonpost.com/technology/2023/03/05/ai-voice-scam/
https://www.forbes.com/sites/jessedamiani/2019/09/03/a-voice-deepfake-was-used-to-scam-a-ceo-out-of-243000/?sh=1081d7802241
https://apnews.com/article/fact-check-biden-audio-banking-fake-746021122607

Background



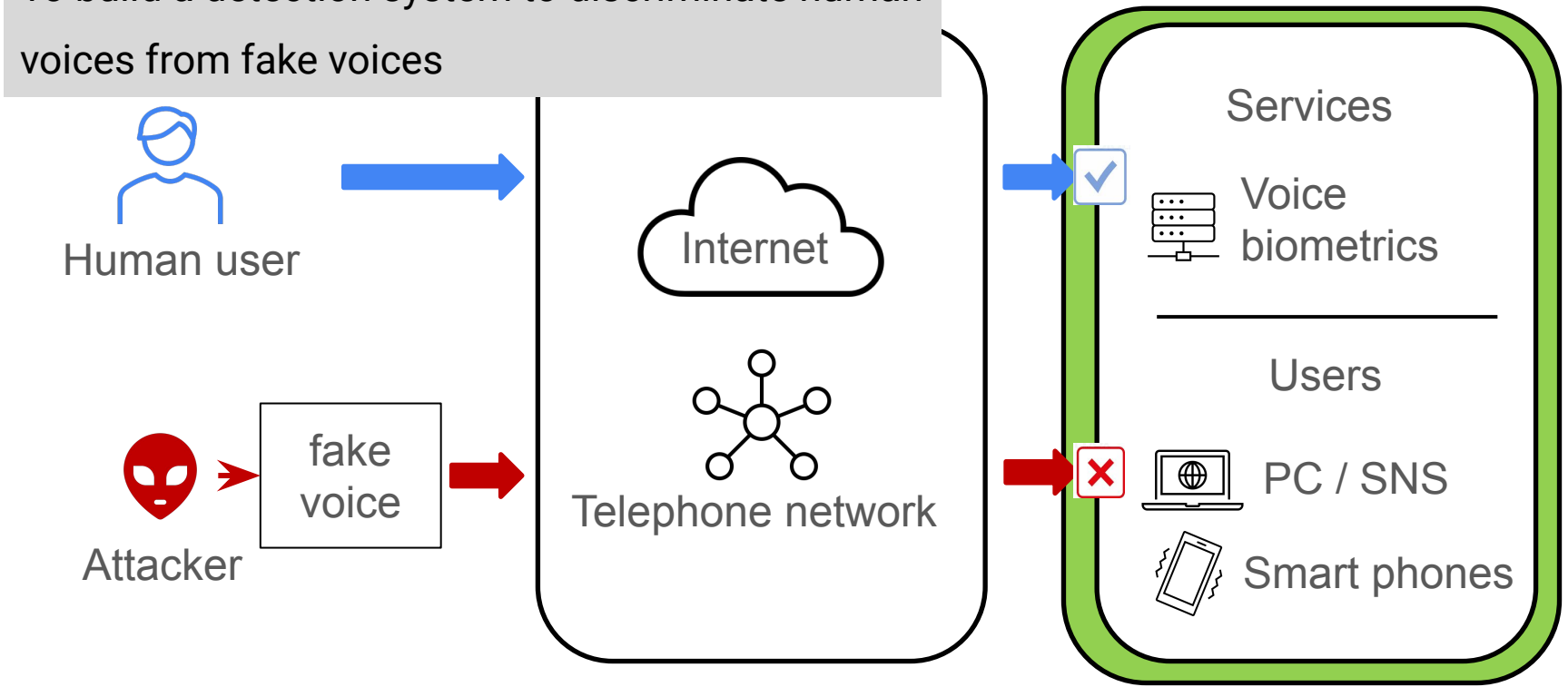




https://www.mcafee.com/blogs/privacy-identity-protection/artificial-imposters-cybercriminals-turn-to-ai-voice-cloning-for-a-new-breed-of-scam/

Task definition

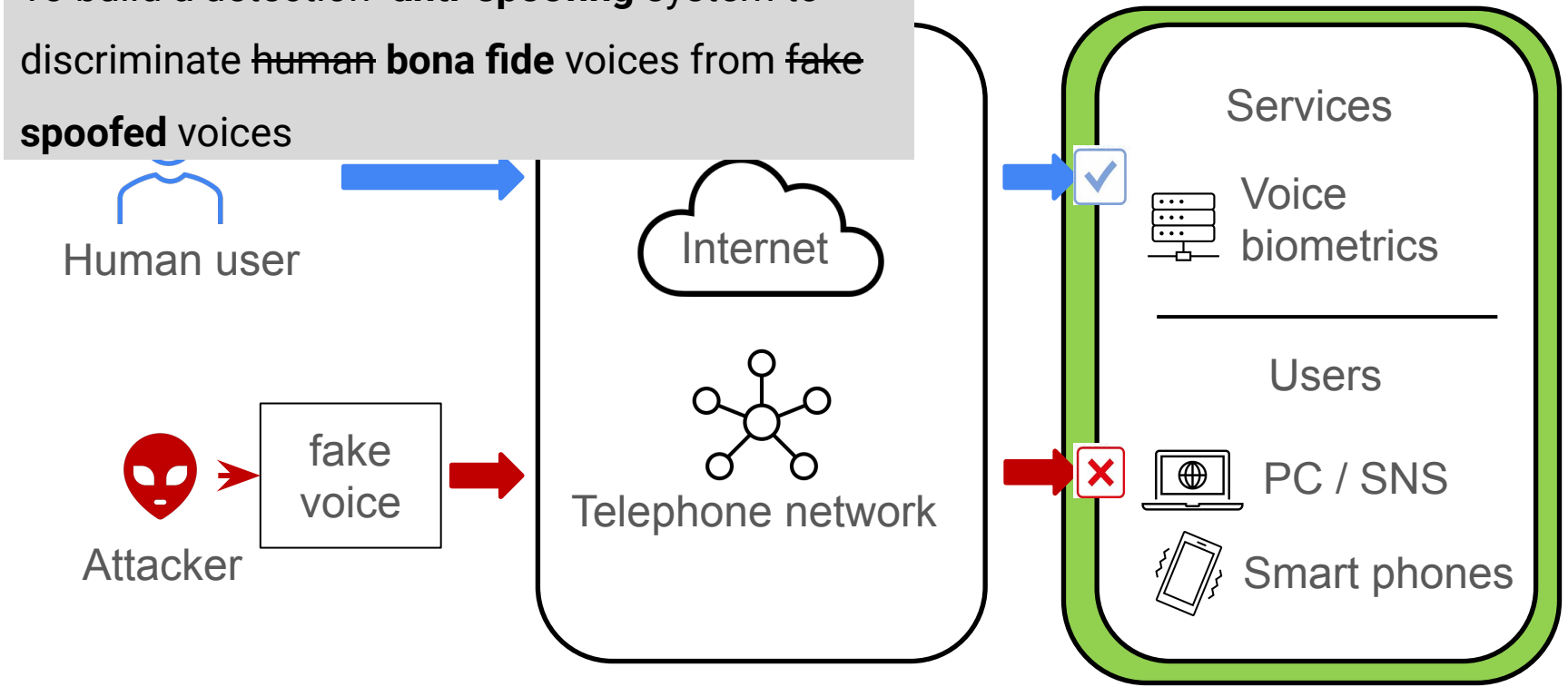
To build a detection system to discriminate human voices from fake voices



Task definition

Other name: presentation attack detection

To build a ~~detection~~ **anti-spoofing** system to discriminate ~~human~~ **bona fide** voices from ~~fake~~ **spoofed** voices



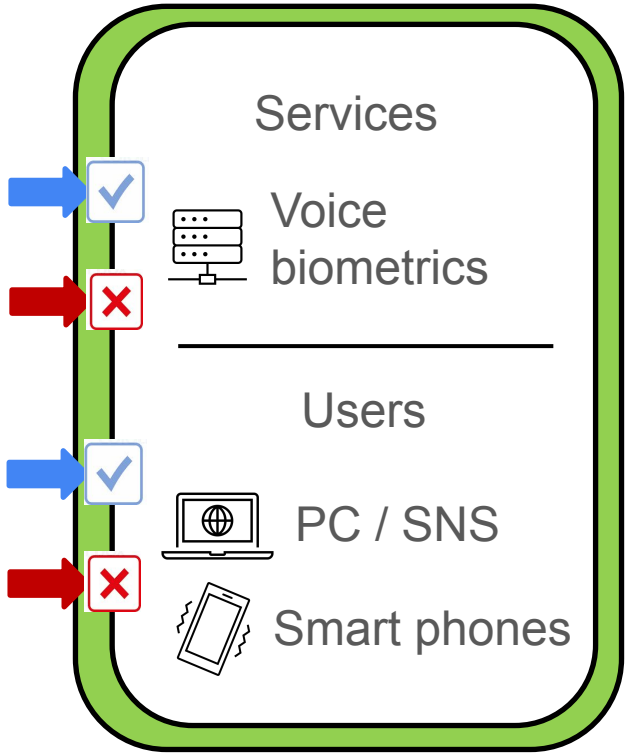
Task definition

Targeting at voice biometrics

- Fake voice needs to fool ASV
- It may not fool human ears

Targeting at human ears

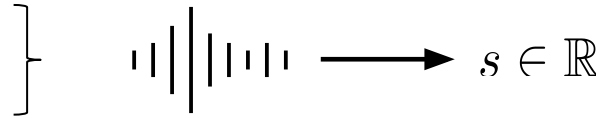
- Fake voice needs to fool human ears
- No voice biometrics involved



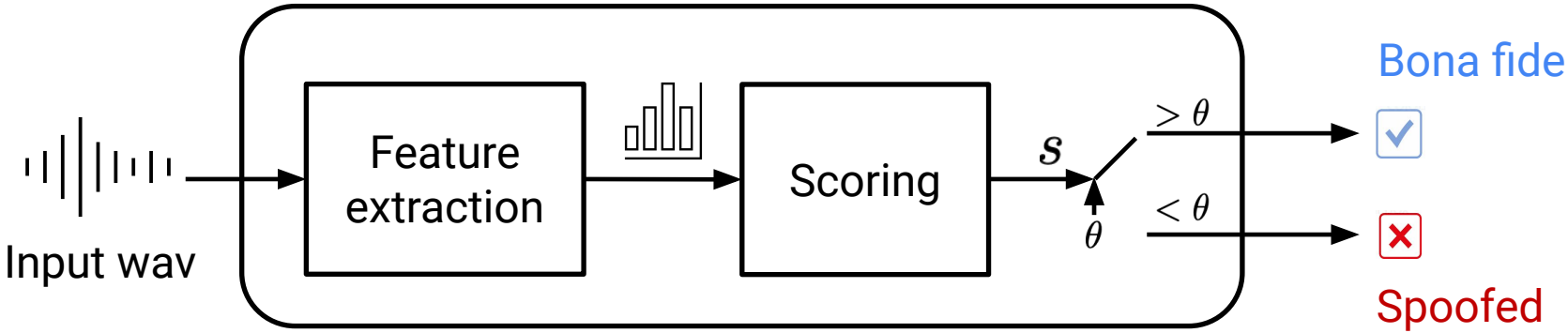
Task definition

A binary classification task

- Feature extraction: front end
- Scoring: back end
- Decision

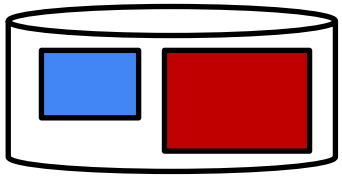


How likely the input waveform is bona fide

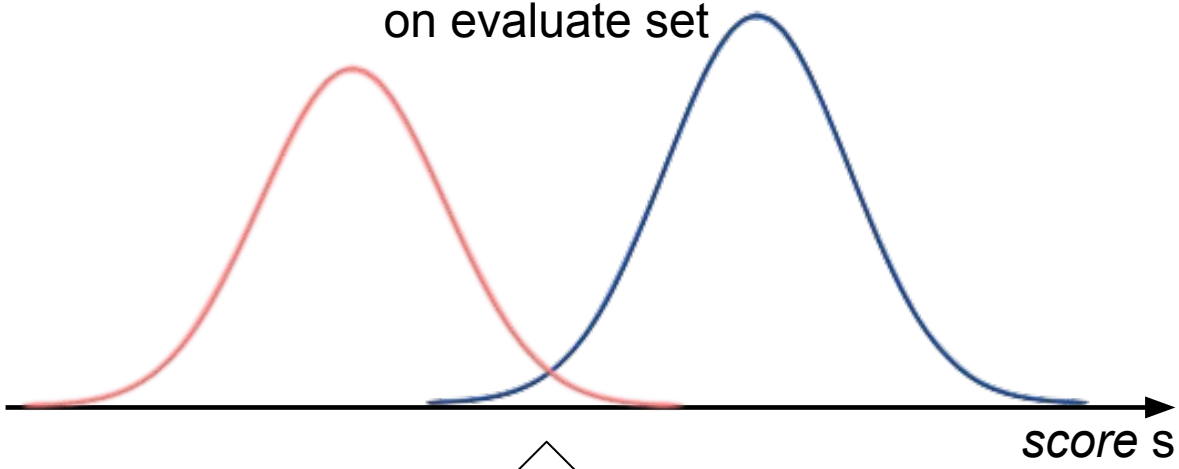


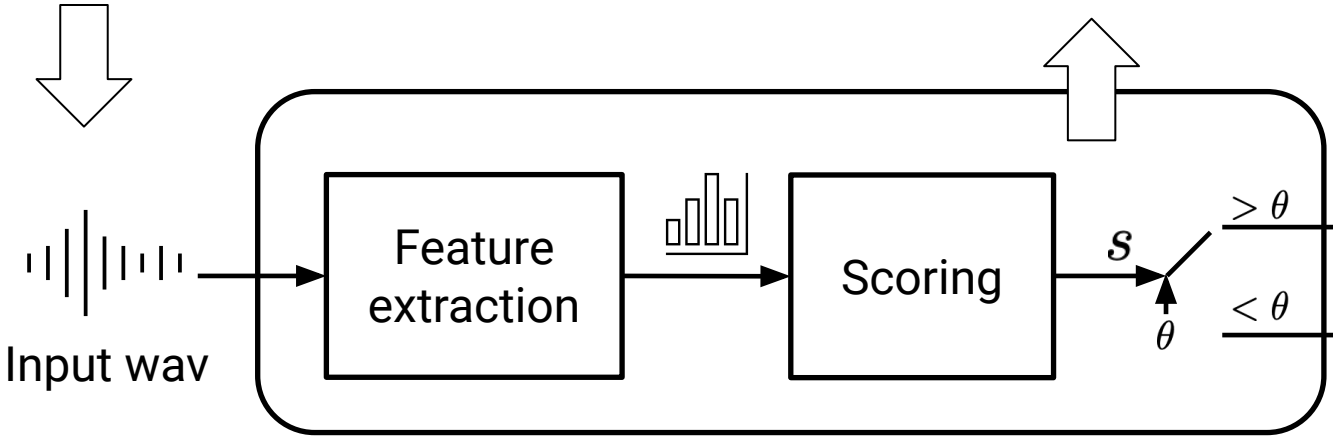
Evaluation metrics

Data set for evaluation



Distribution of scores
on evaluate set



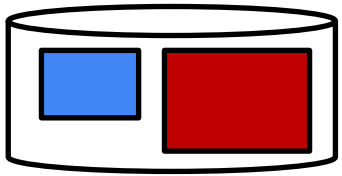


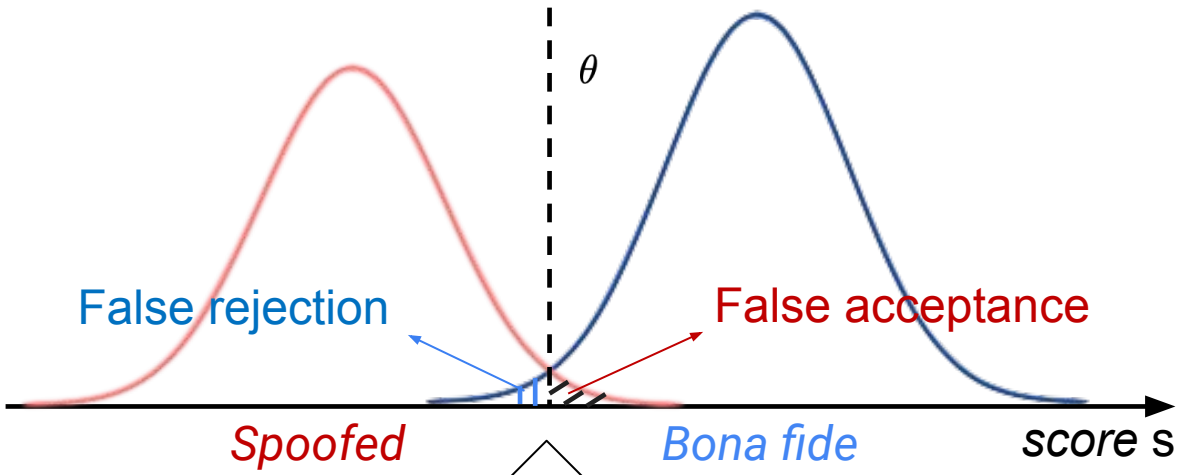
Bona fide

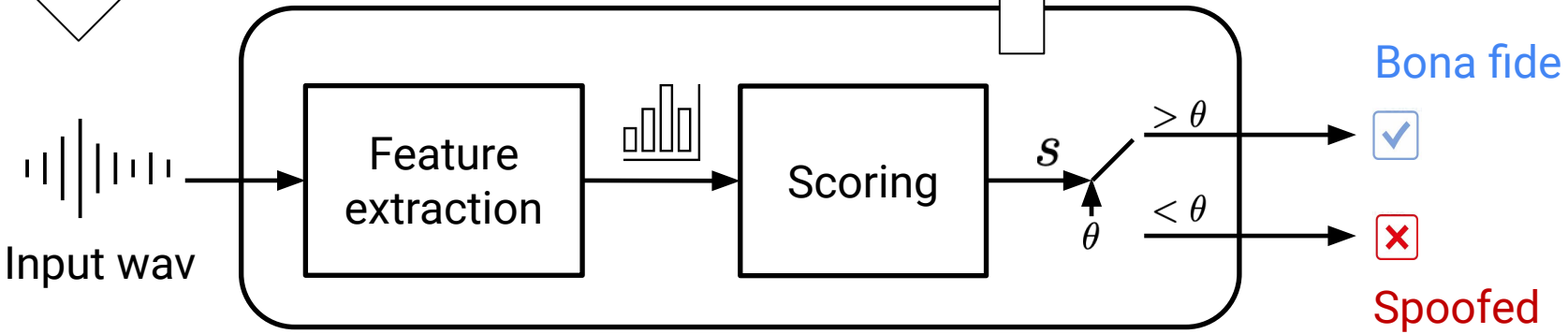
Spoofed

Evaluation metrics

Data set for evaluation

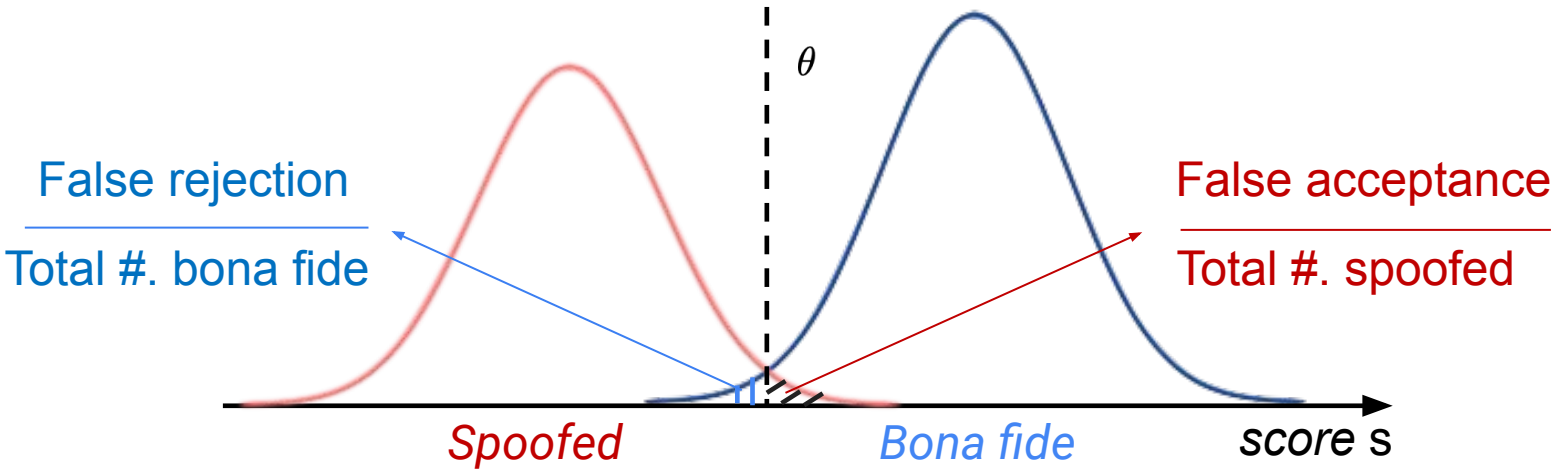






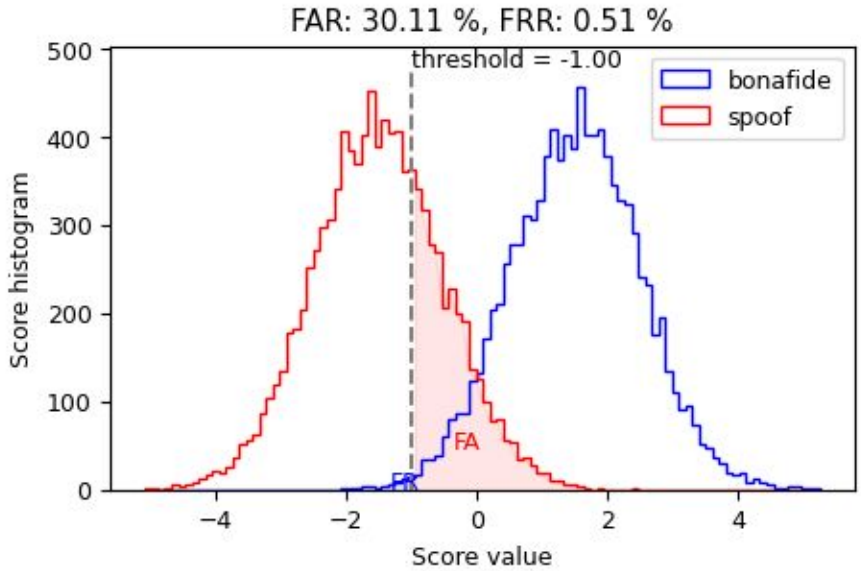
Evaluation metrics

- False rejection rate (FRR)
- False acceptance rate (FAR)



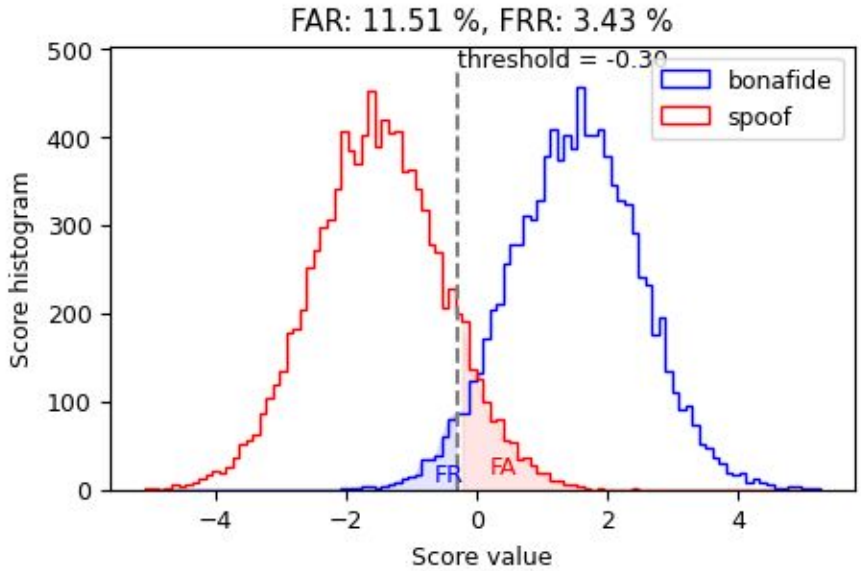
Evaluation metrics

- False rejection rate (FRR)
- False acceptance rate (FAR)



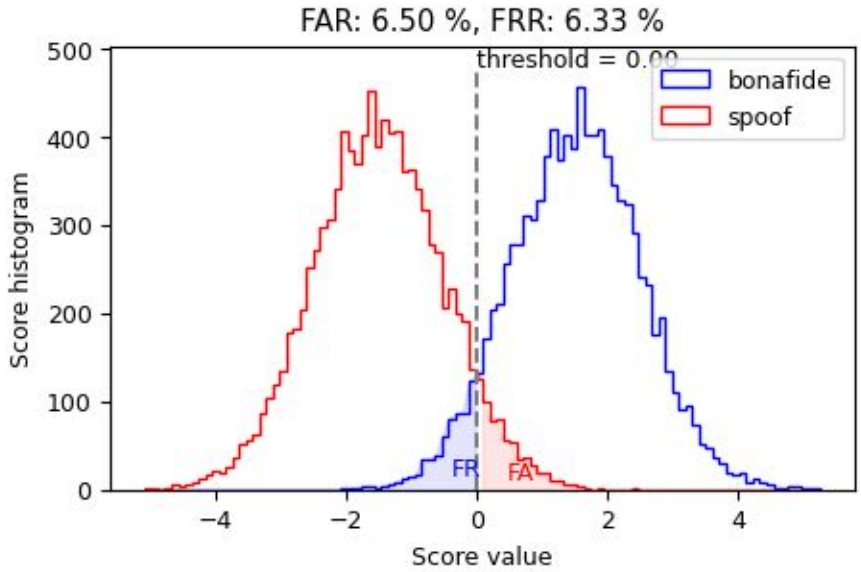
Evaluation metrics

- False rejection rate (FRR)
- False acceptance rate (FAR)



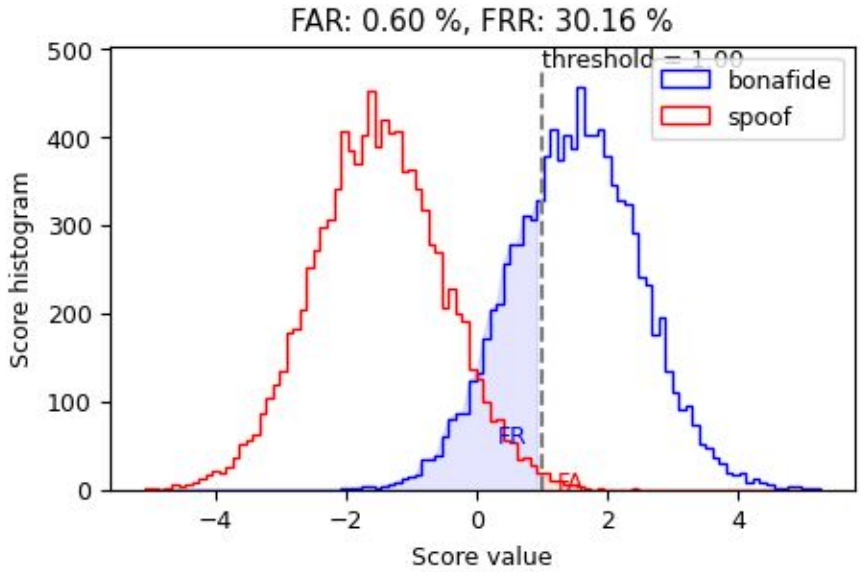
Evaluation metrics

- False rejection rate (FRR)
- False acceptance rate (FAR)



Evaluation metrics

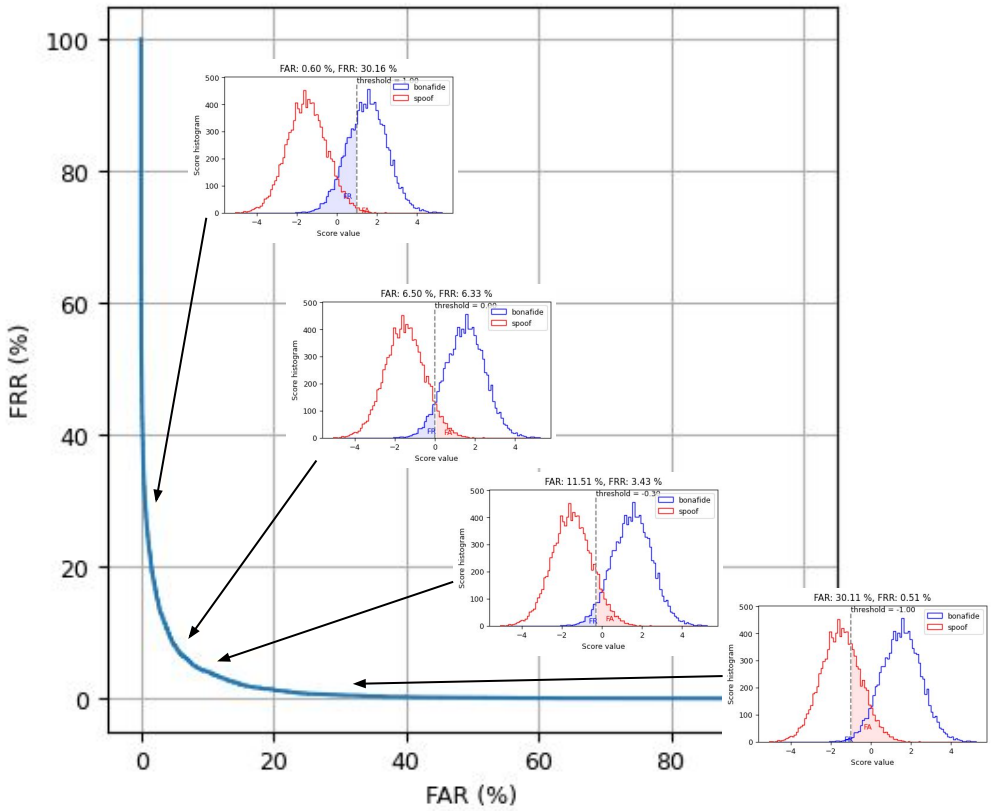
- False rejection rate (FRR)
- False acceptance rate (FAR)



Evaluation metrics

Trade-off between FRR and FAR

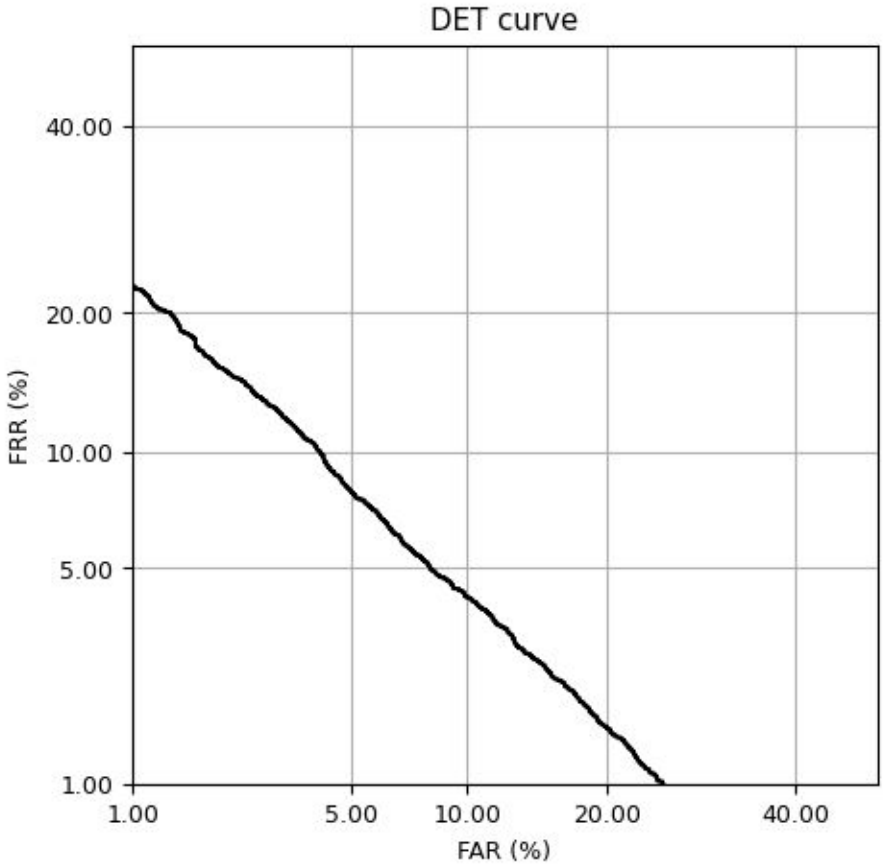
- one increases, the other decreases
- similar to ROC curve, but on errors



Evaluation metrics

Detection error trade-off (DET) curve (Martin 1997)

- Wrap the axes to make it easier to read
- See Tutorial notebook for details

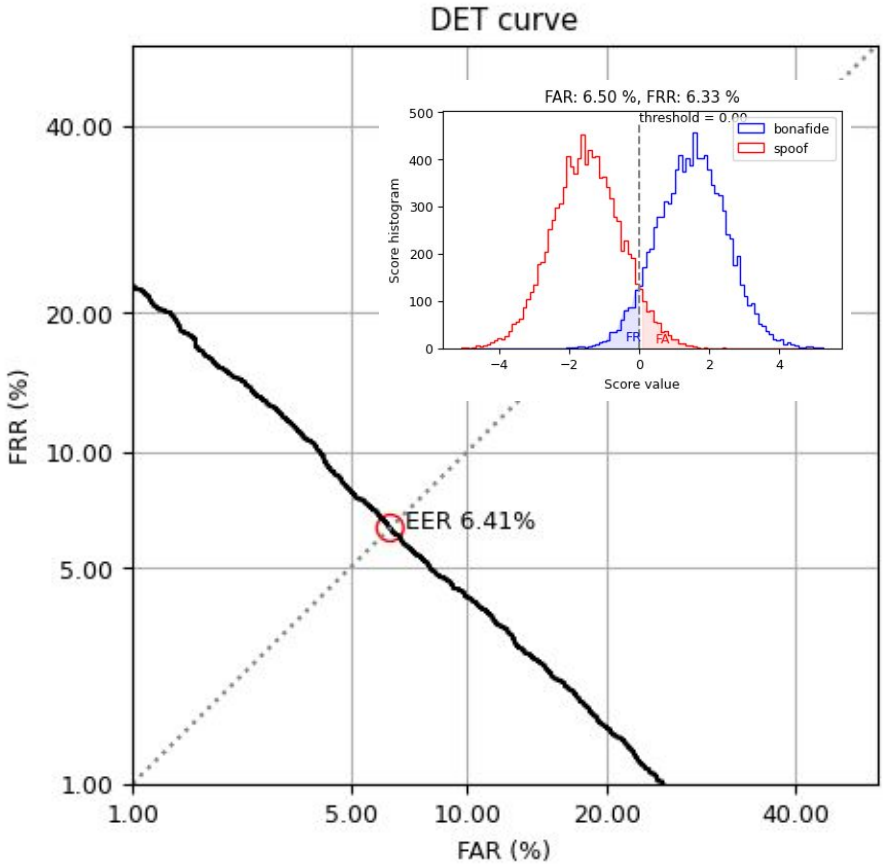


Evaluation metrics

Detection error trade-off (DET) curve (Martin 1997)

Equal error rate (EER)

- Choose the operating point (threshold) so that FAR is **roughly** equal to FRR
- We don’t need to specify the threshold manually

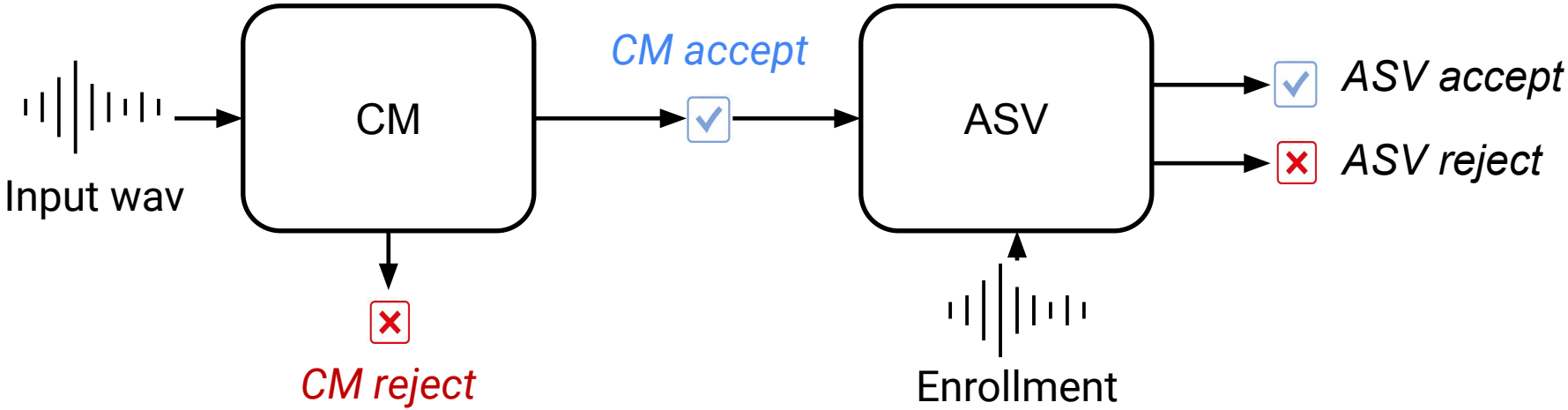


EER measures discrimination, but not calibration. See more in (Van Leeuwen 2007)

Evaluation metrics

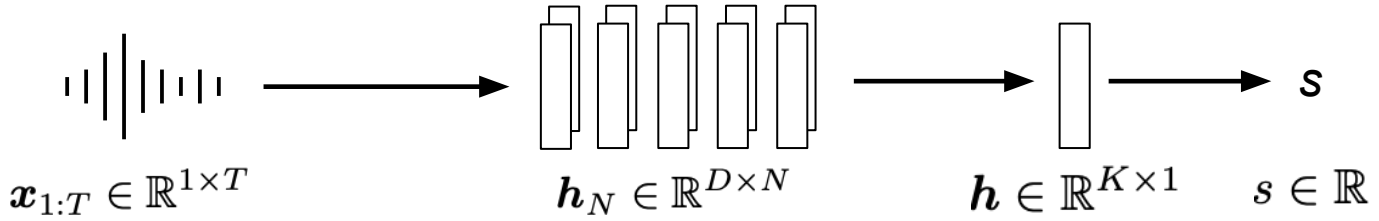
Other metrics

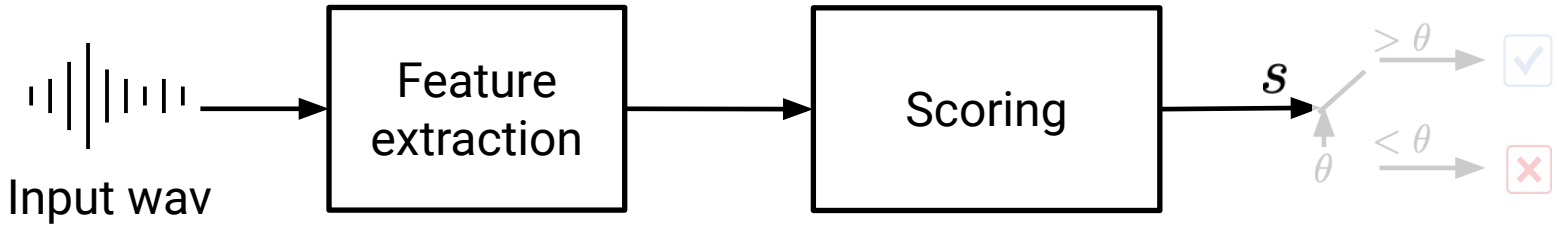
- tandem Detection Cost Function (t-DCF) (Kinnunen 2020)



Common methods

Transformation from waveform to a scalar value *s*

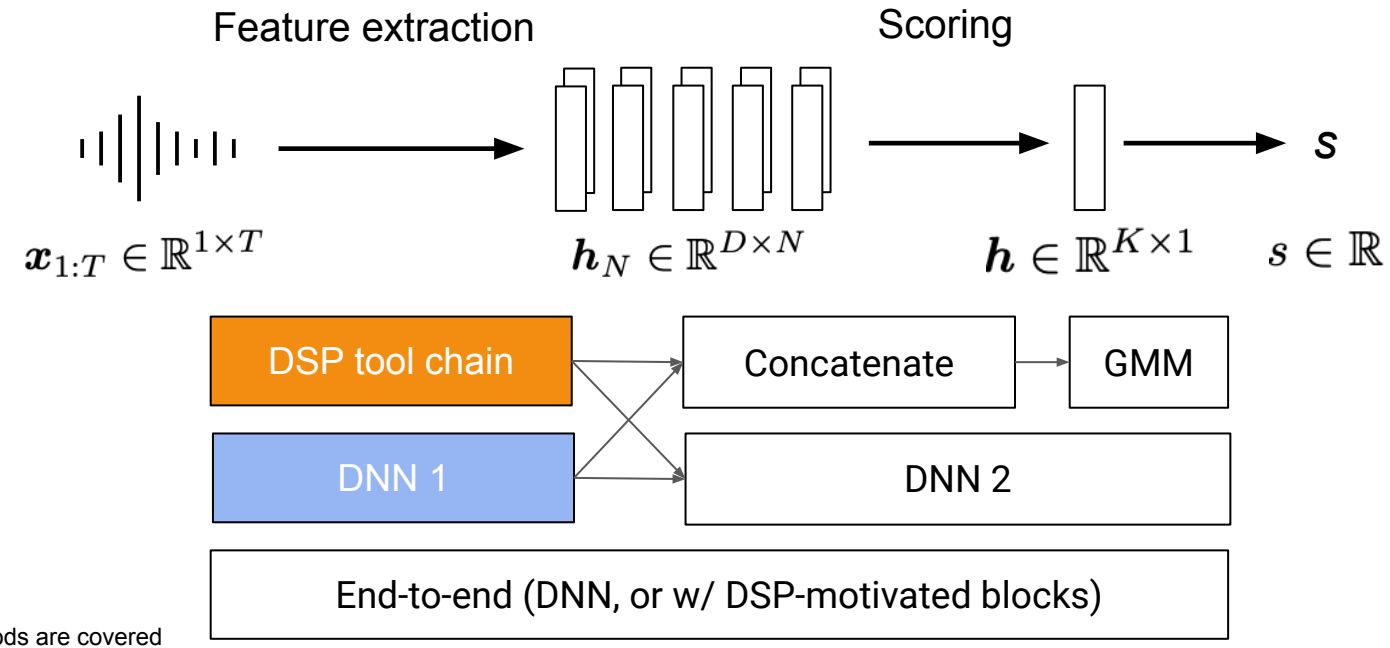




Not all methods are covered

Common methods

Flavors of feature extraction and scoring



Not all methods are covered

Common methods

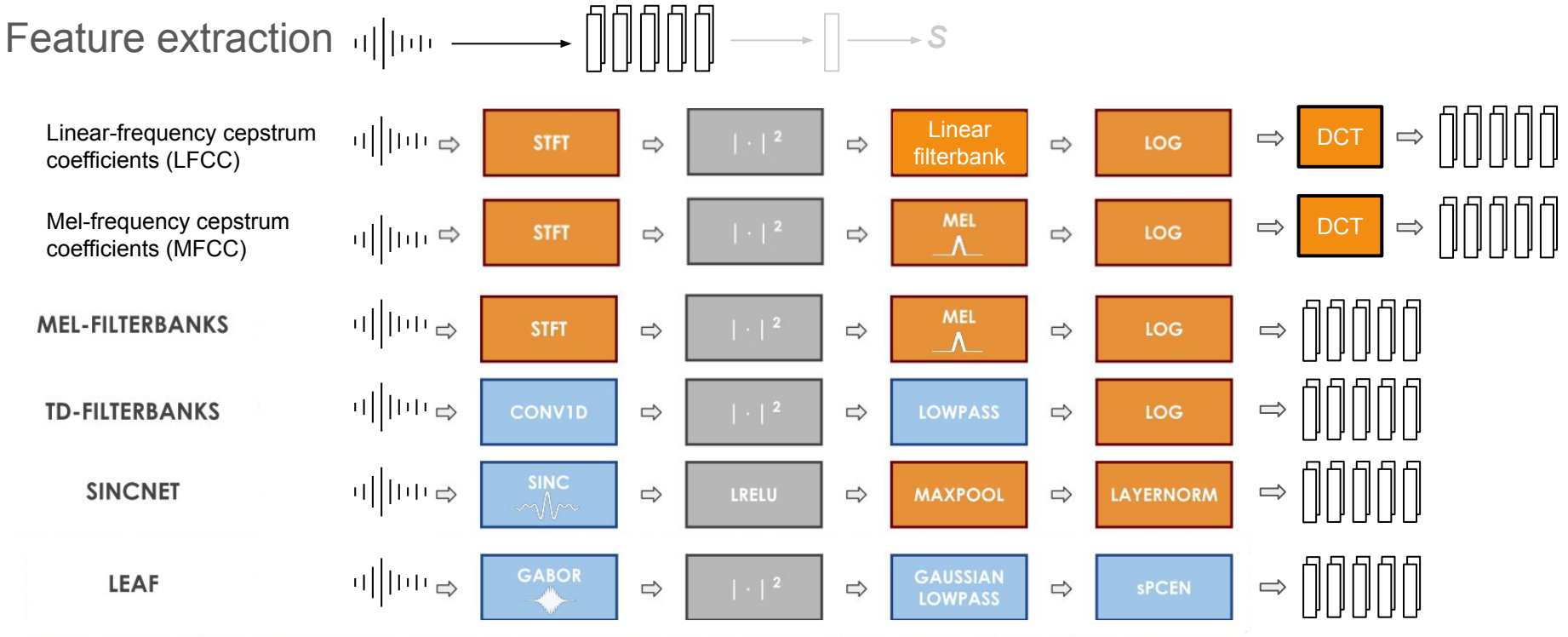
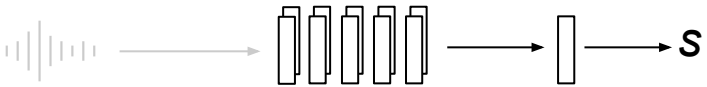
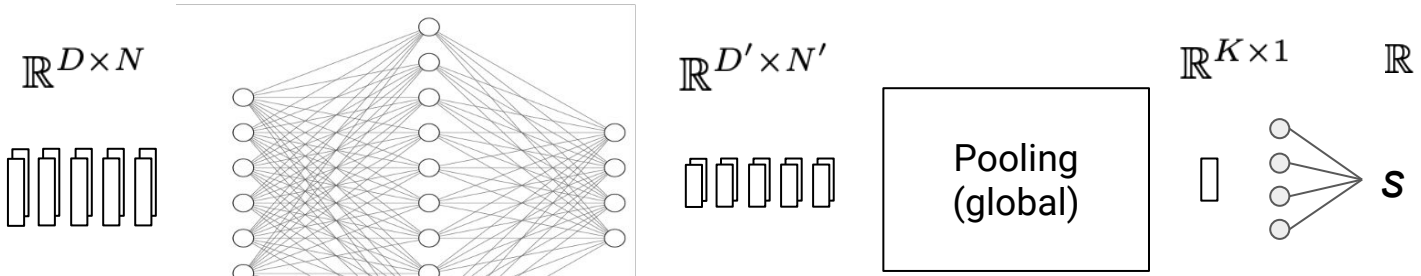


Figure adapted from (Zeghidour 2021)

Common methods

Back-end scoring





Multi-layer perceptron

Recurrent neural network

Convolution neural network

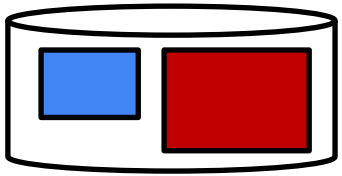
Light CNN, ResNet, LSTM, …

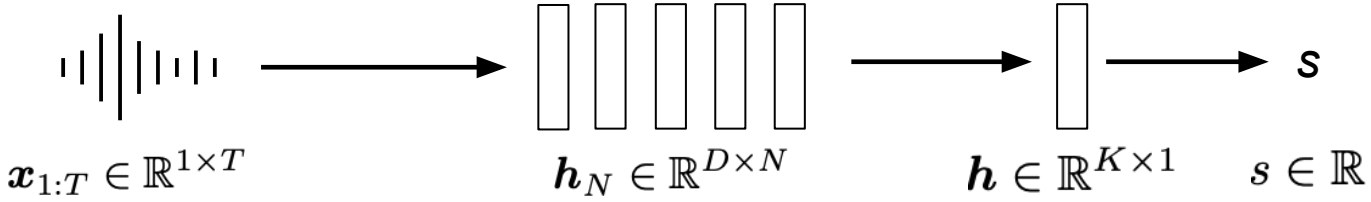
Averaging
Max pooling
Attention
…

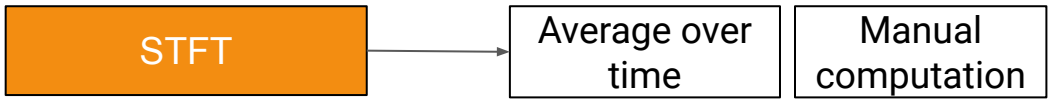
See overview papers (Kamble 2020, Wang 2022)

Toy example

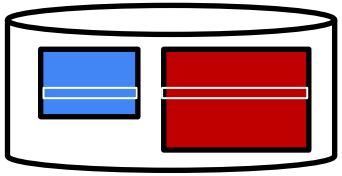
Let’s create a simple CM by hands on a toy training set (from ASVspoof 2019 LA database)







Toy example

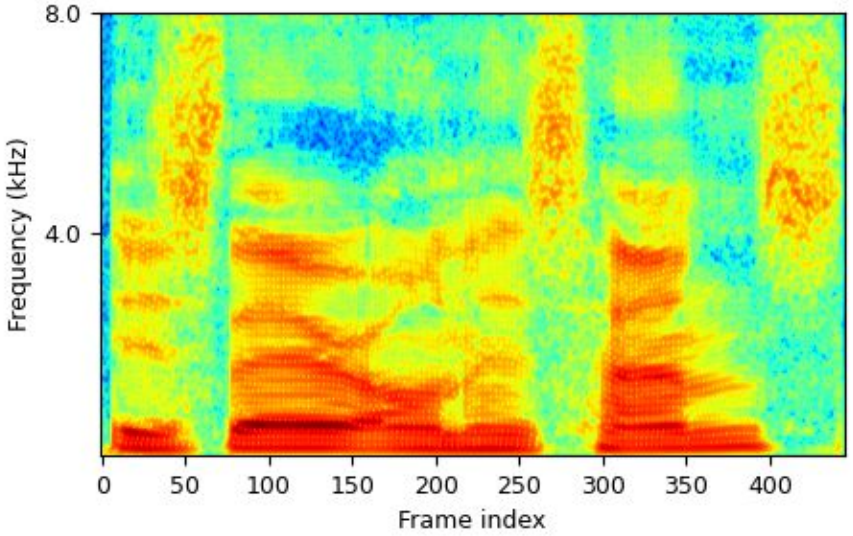


One bona fide utterance from speaker LA_0082

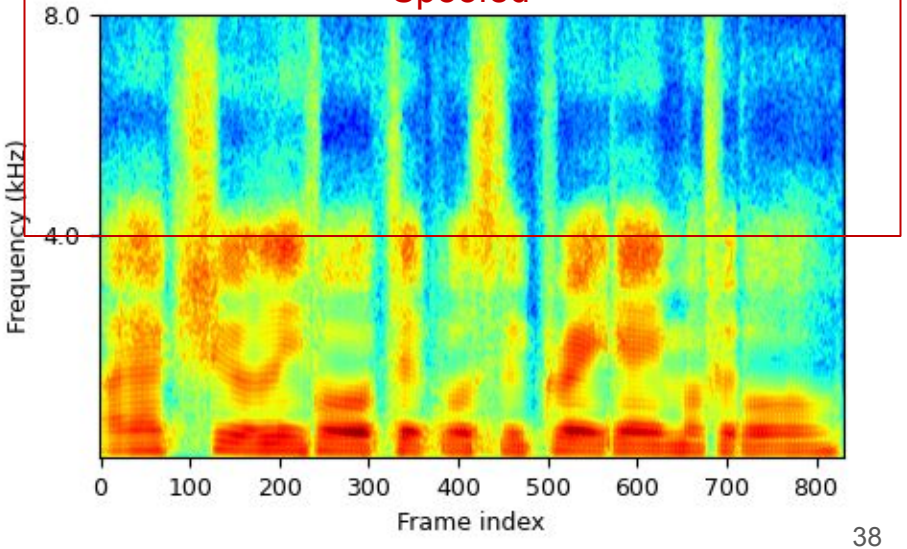
One spoofed utterance from attack 1

Darker value in high-frequency region

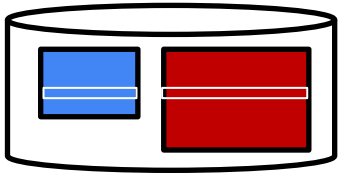
Bona fide



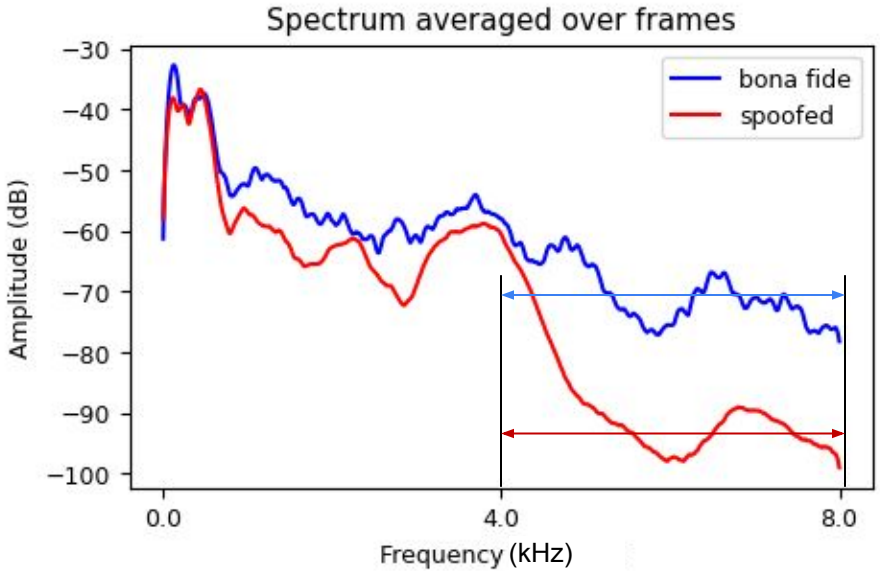
Spoofed



Toy example

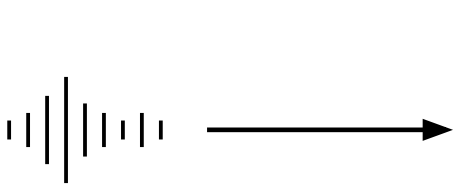


One bona fide utterance from speaker LA_0082
One spoofed utterance from attack 1

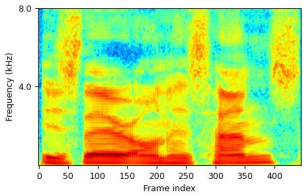


- Large difference above 4.0 kHz
- Let's compute the average value within 4.0 – 8.0 kHz

Toy example

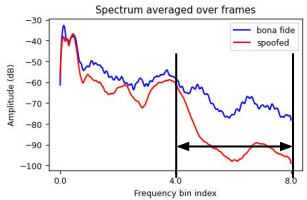


$\boldsymbol{x}_{1:T} \in \mathbb{R}^{1 \times T}$



$\boldsymbol{h}_N \in \mathbb{R}^{D \times N}$

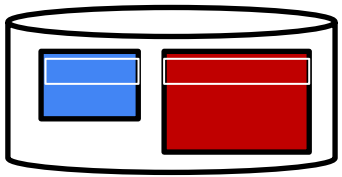
(or higher dimensional tensor)



$\boldsymbol{h} \in \mathbb{R}^{K \times 1}$

s

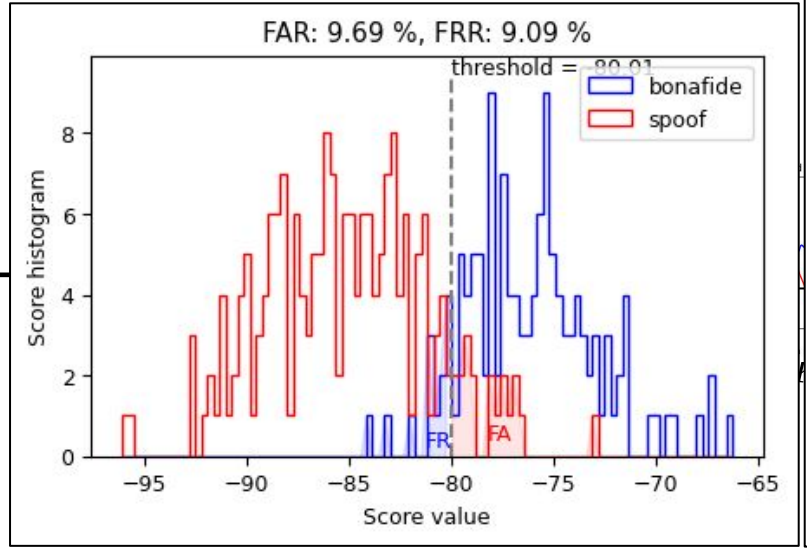
$s \in \mathbb{R}$

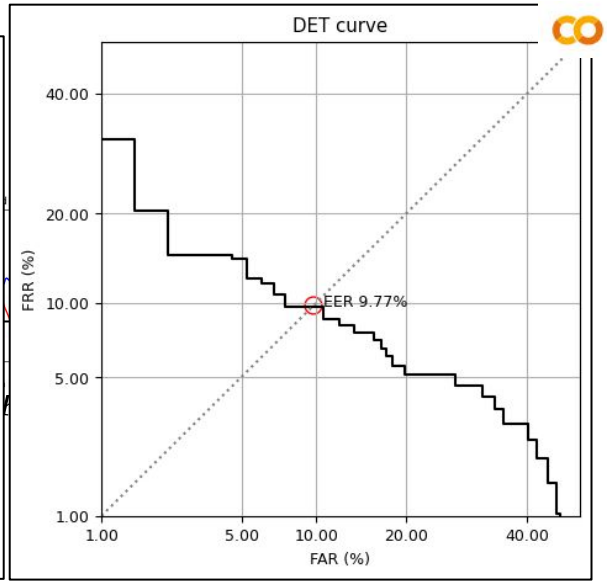


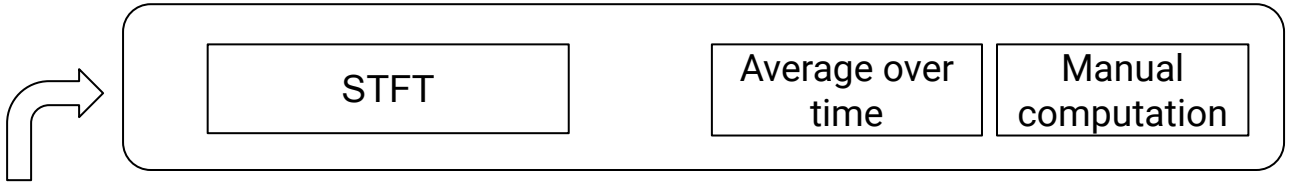
Let’s try the “expert-knowledge” CM on the data of speaker LA_0082

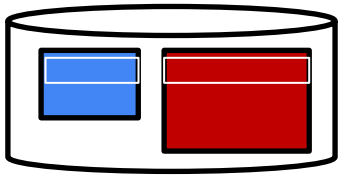
Toy example

$$\boldsymbol{x}_{1:T} \in \mathbb{R}^{1\times T}$$





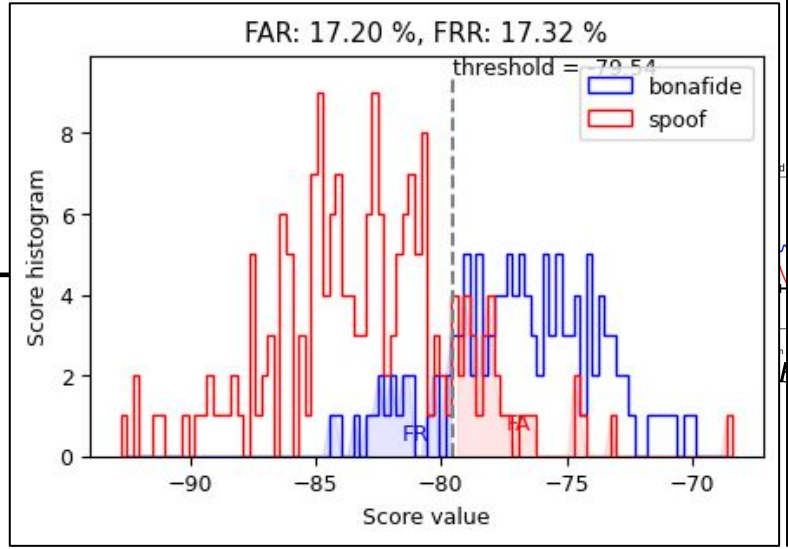


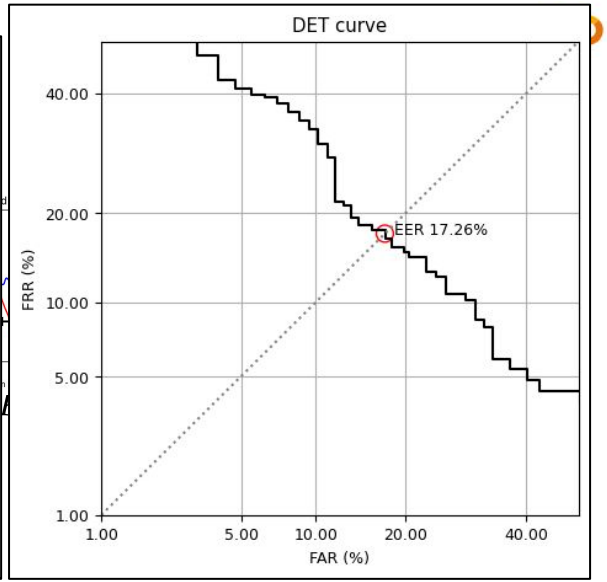


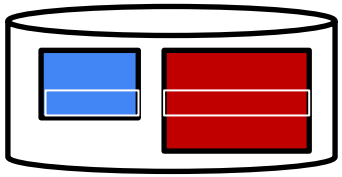
Discriminate bona fide data (**LA_0082**) from **attack 1**

Toy example

$$\boldsymbol{x}_{1:T} \in \mathbb{R}^{1\times T}$$





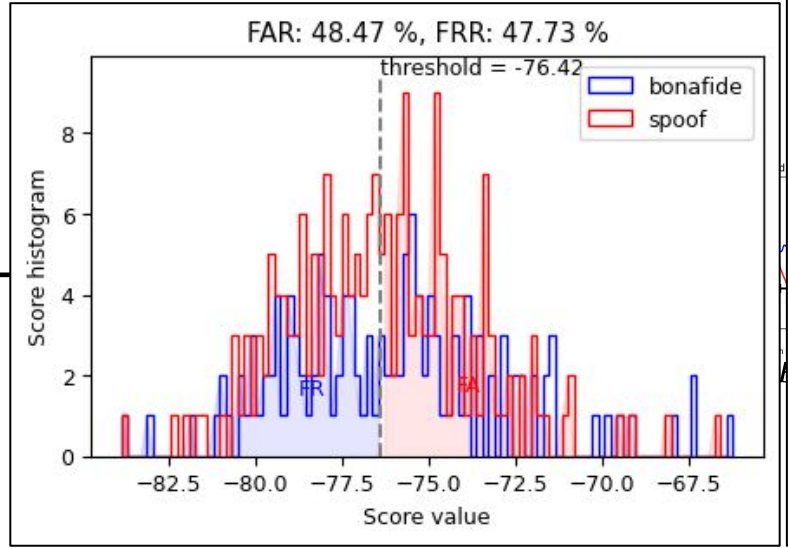


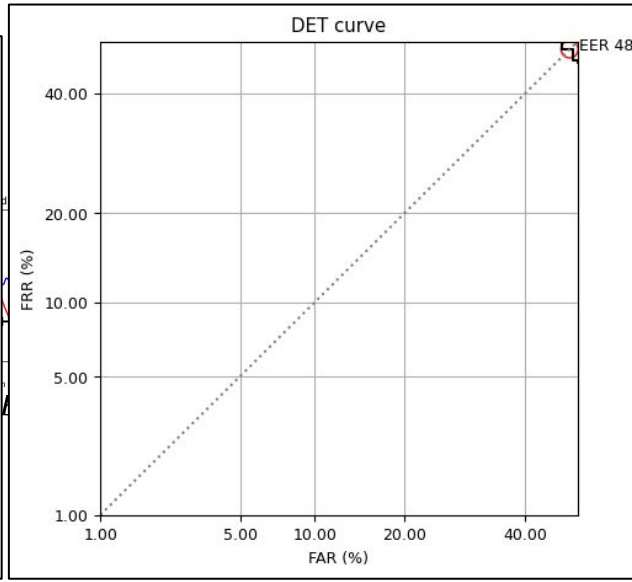
Discriminate bona fide data (**LA_0087**) from **attack 1**

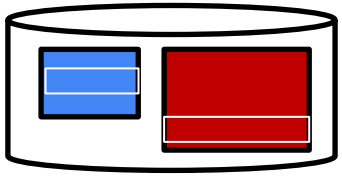
(data from a different speaker)

Toy example

$$\boldsymbol{x}_{1:T} \in \mathbb{R}^{1\times T}$$





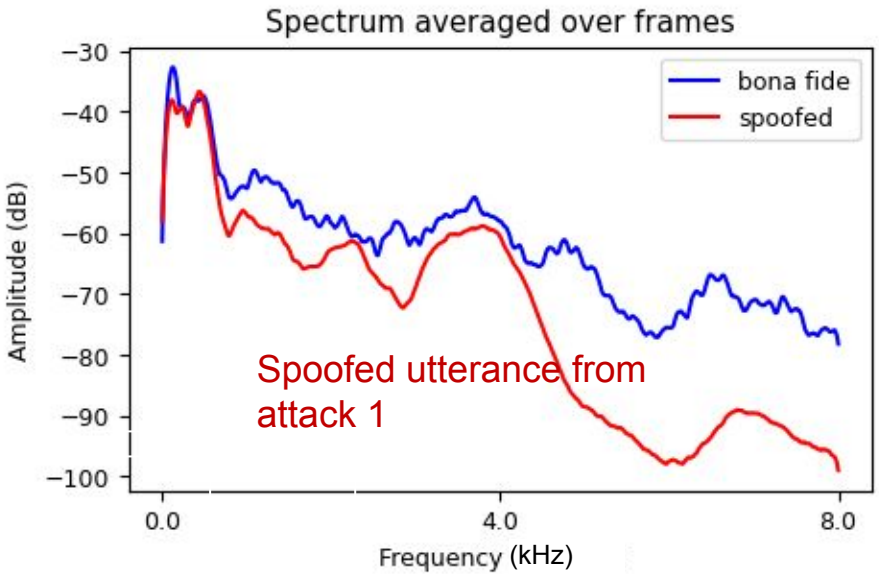


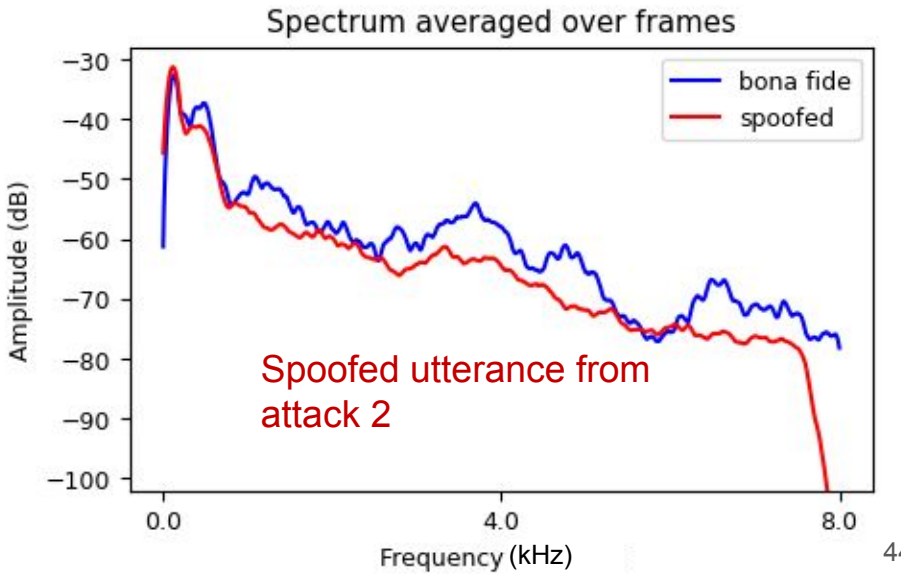
Discriminate bona fide data (**LA_0082**) from attack 2

(data from a different spoofing attack)

Toy example

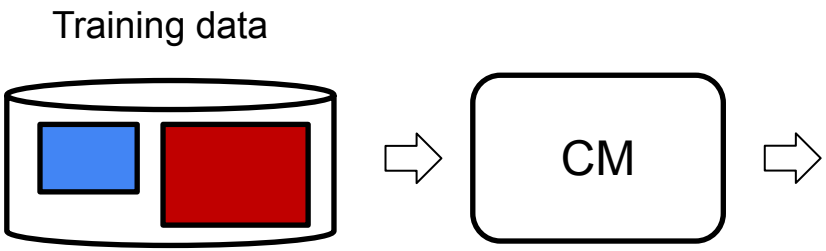
Different attacks have leave different “traits”

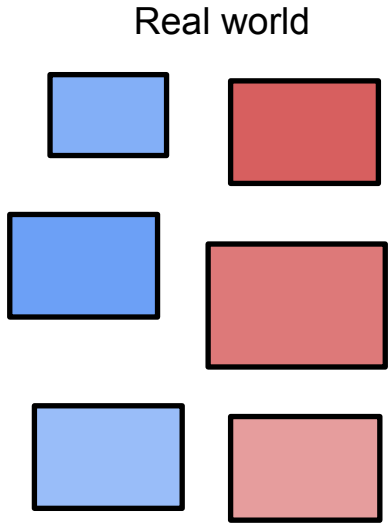




Why is anti-spoofing difficult

Unseen variabilities real world test data





- Speakers
- Unseen attacks
- Channels
- Languages
- …

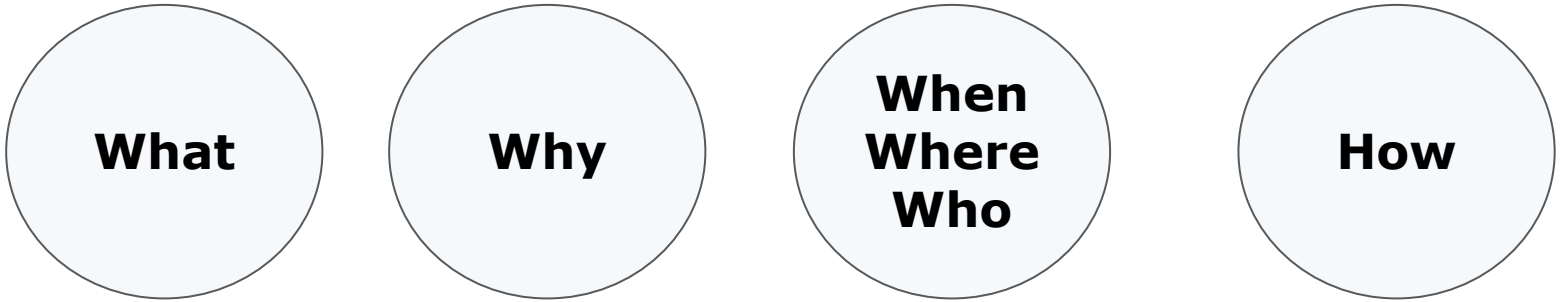
We need more powerful tools to extract features and compute score

Session 1

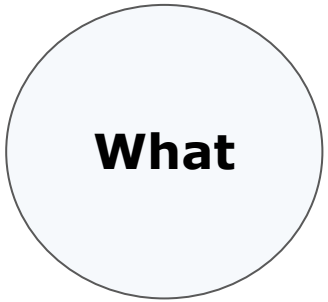
- Introduction to speech anti-spoofing
- **Introduction to graph neural networks**
  - Background
    - Definition
    - Motivation / Advantage
    - Use cases
    - Explanation
  - Brief introduction of graph attention networks
- Graph attention networks for speech anti-spoofing

Background

Graph Neural Networks (GNN)



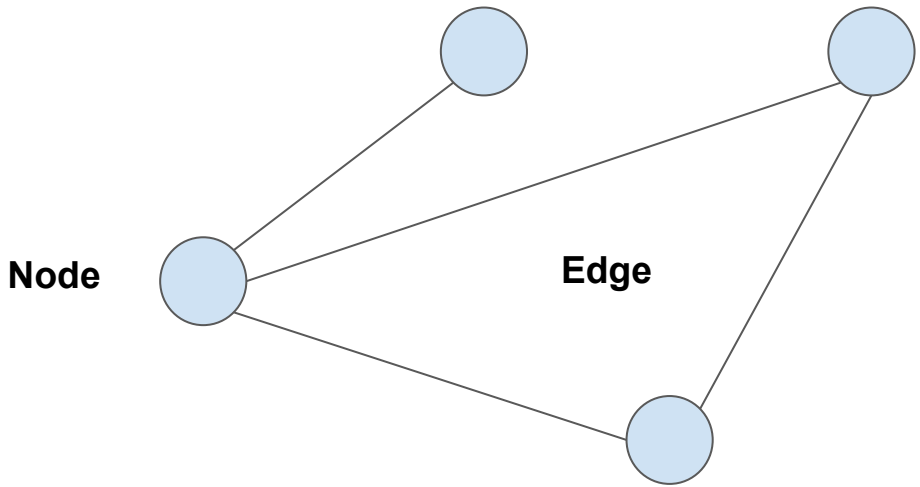
Definition of GNN



is GNN?

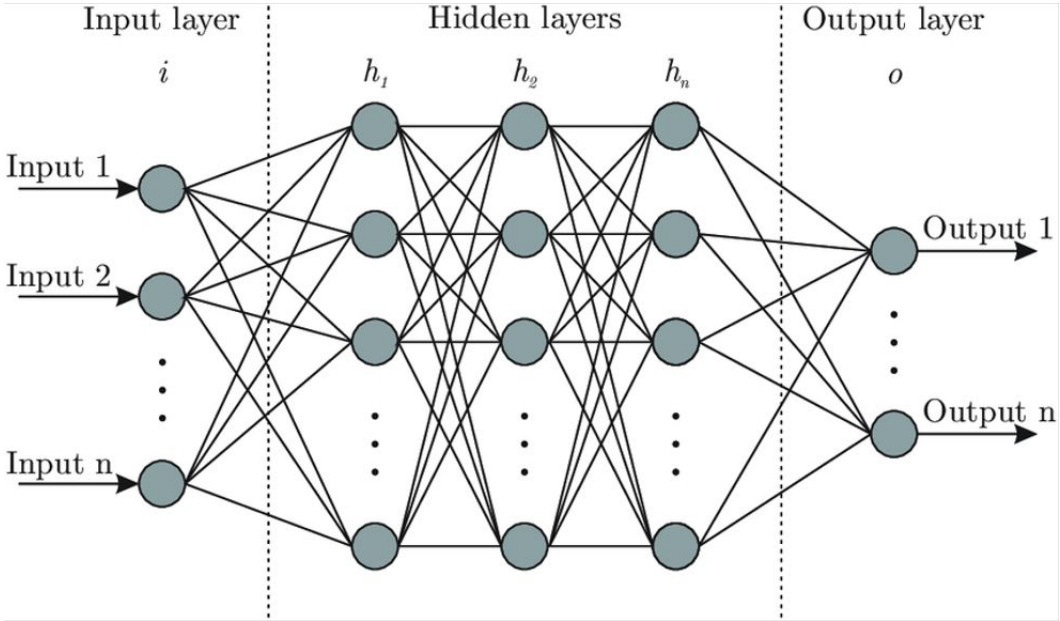
Definition of GNN

Graph Neural Network



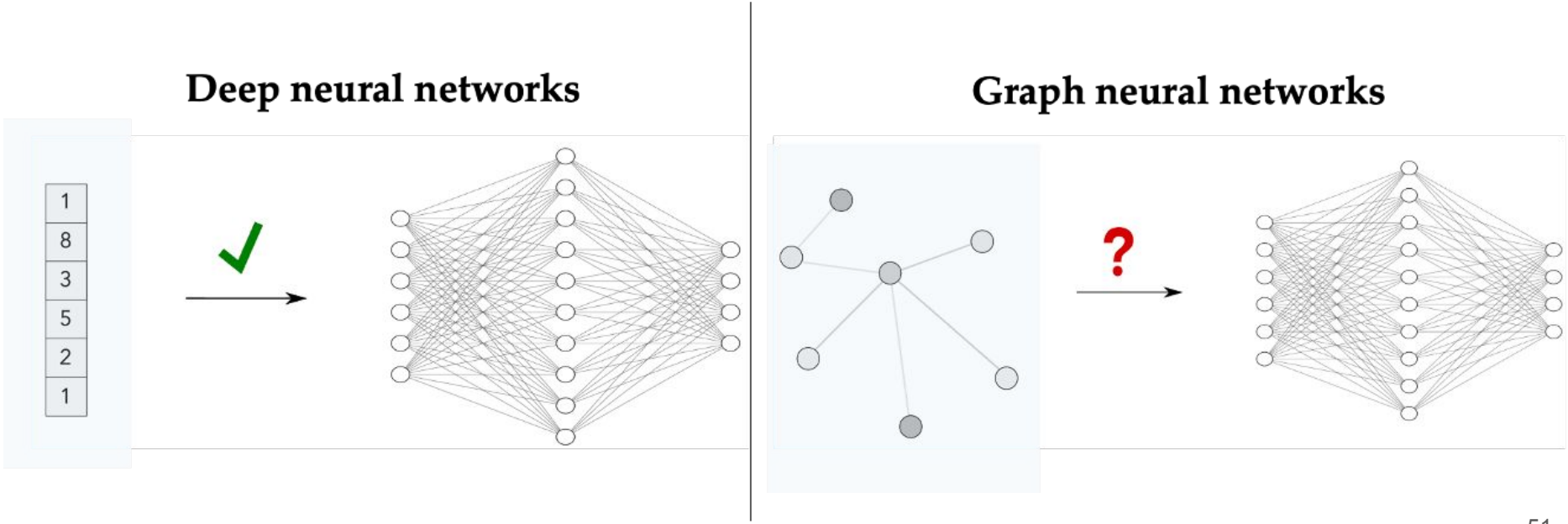
Definition of GNN

Graph **Neural Network**

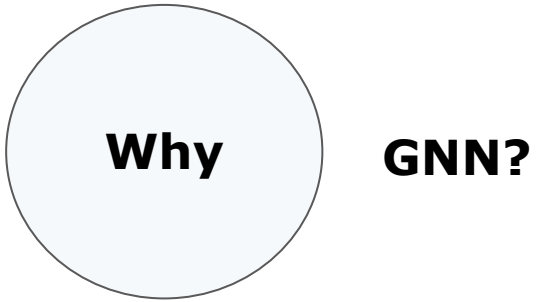


Definition of GNN

Graph Neural Network

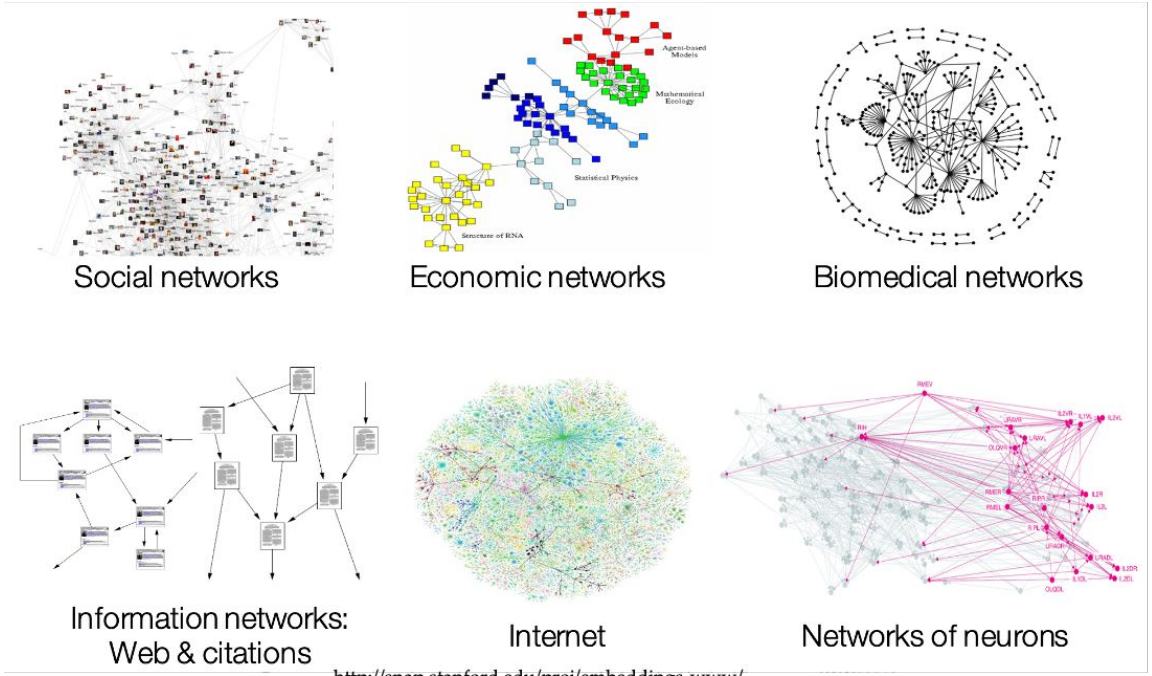


Motivation / Advantage of GNN



Motivation / Advantage of GNN

Example of graph data



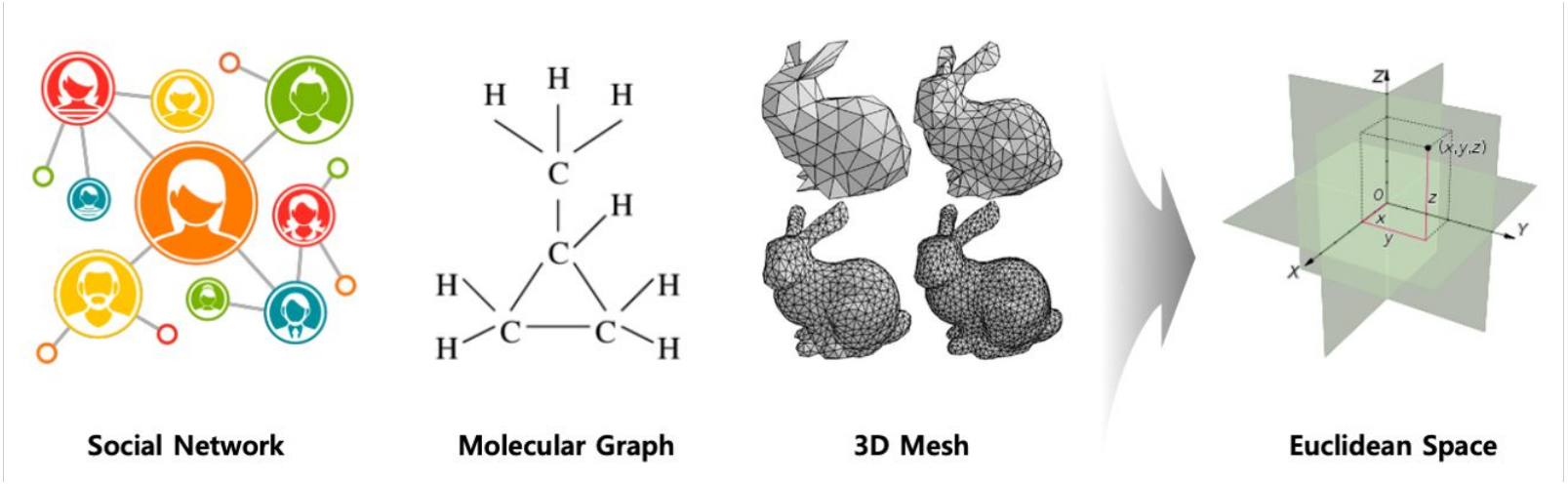
http://snap.stanford.edu/proj/embeddings-www/

Motivation / Advantage of GNN

Graph data

- Represent complex data by Non-Euclidean modeling

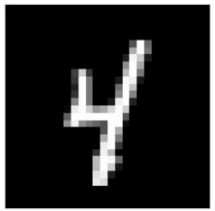
  → Comprehensive understanding

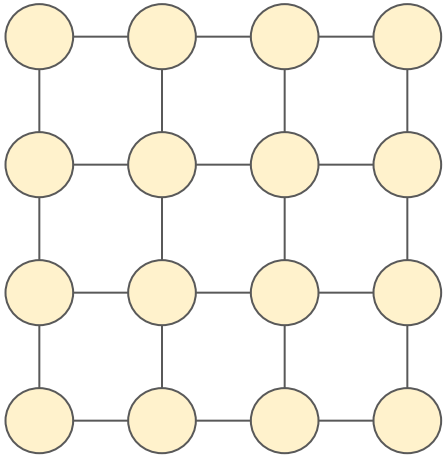


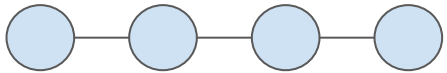
Motivation / Advantage of GNN

Why now we focus on graph?

All data can be represented as graph!

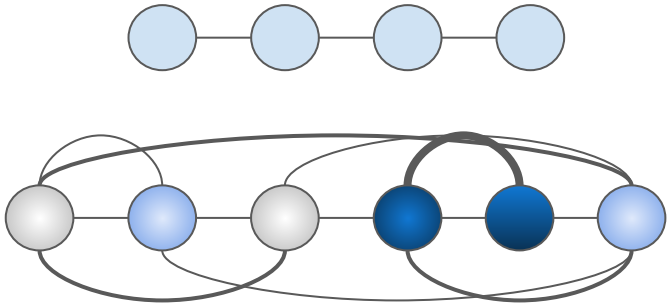






Motivation / Advantage of GNN

Not just simple sequence but complicate connections

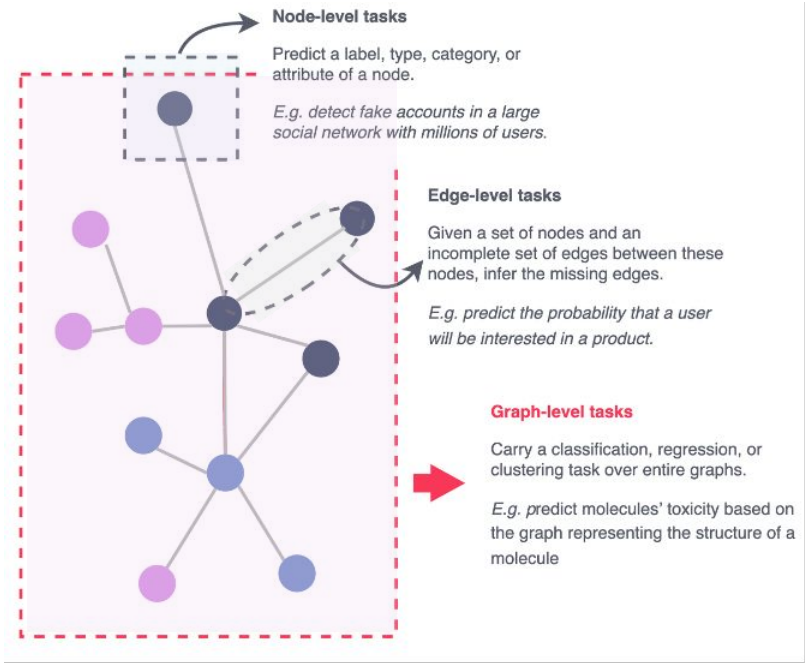


Silence, frequency, amplitude …

Use cases of GNN

uses GNN?

Use cases of GNN

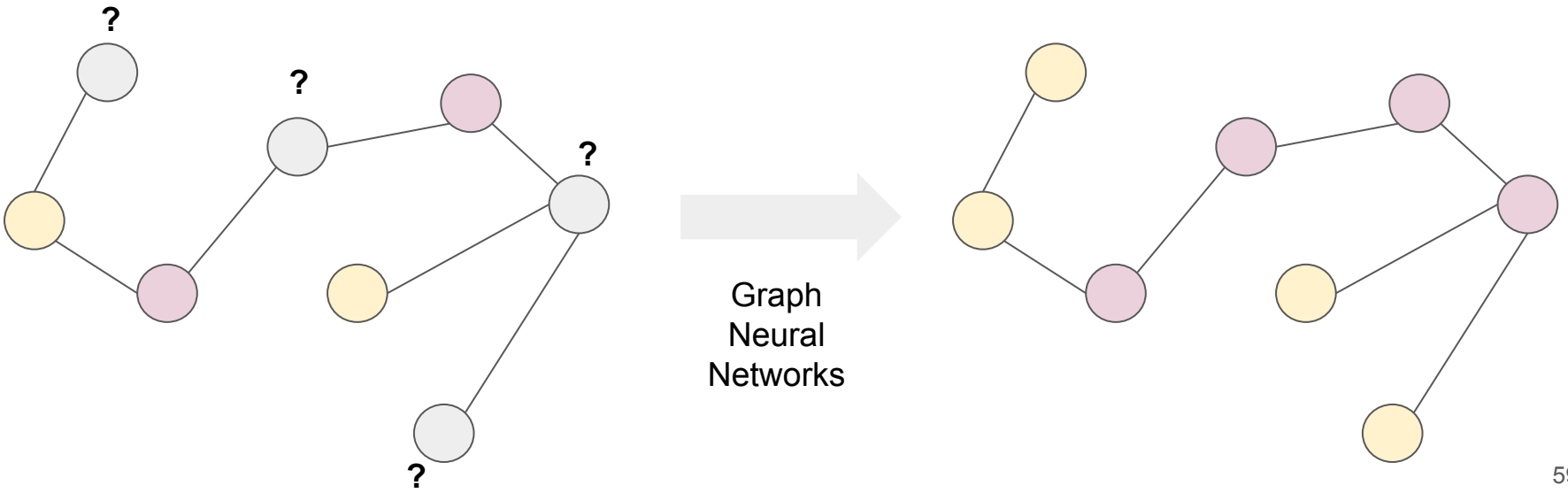


https://blog.dataiku.com/graph-neural-networks-part-three

Use cases of GNN

Node classification (Node-level)

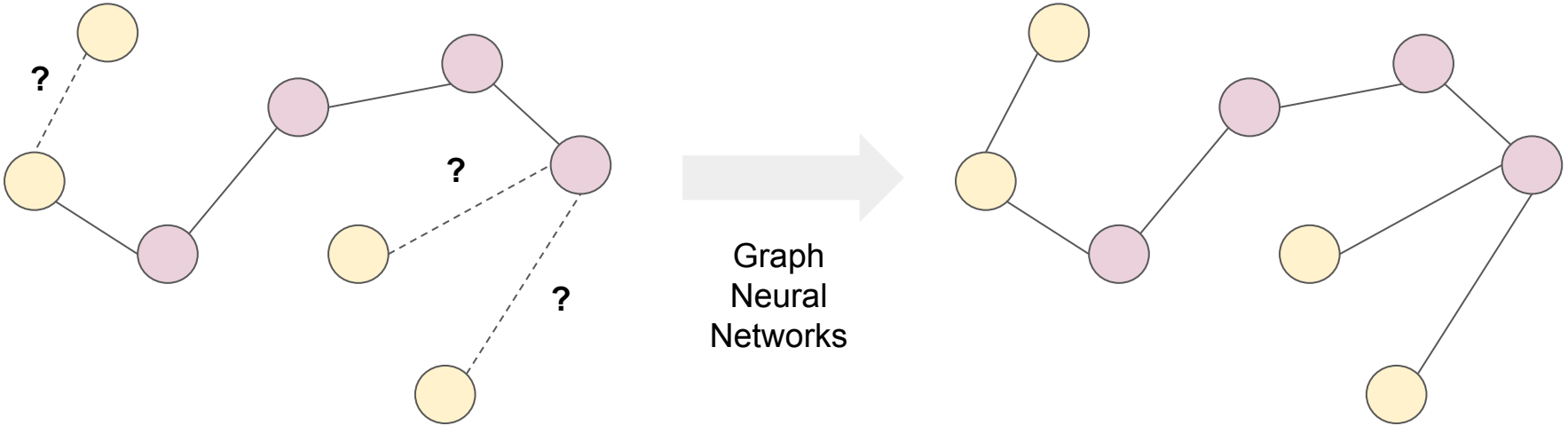
: predict the types of given nodes



Use cases of GNN

Edge/link prediction (Edge-level)

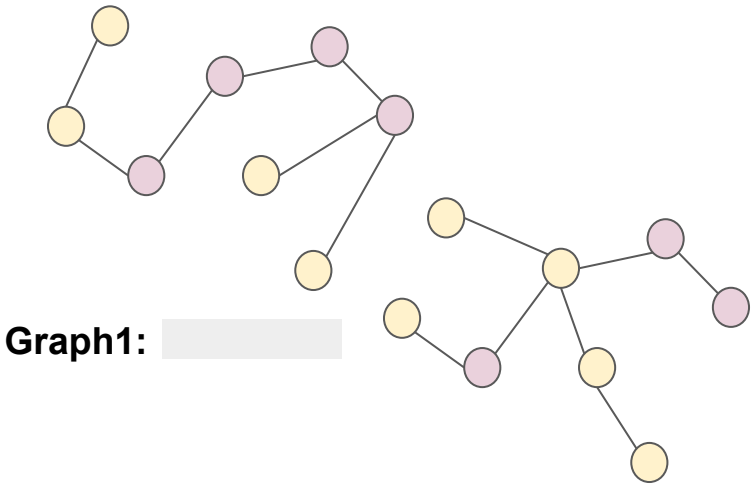
: predict whether given nodes are linked or not



Use cases of GNN

Graph classification (Graph-level)

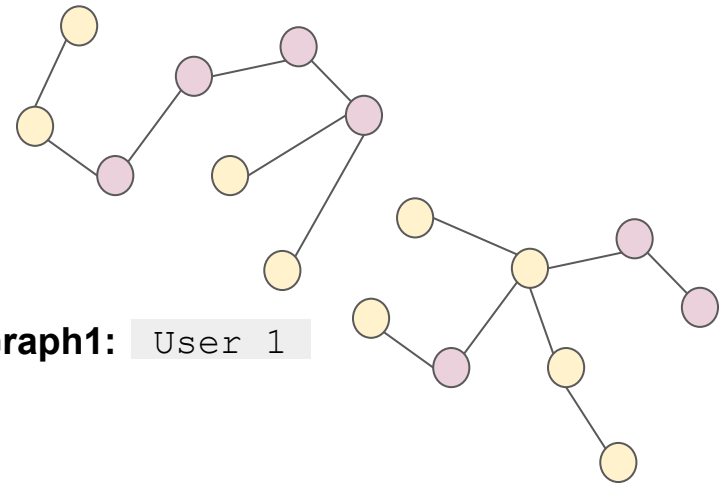
: classify graph itself into given classes



Graph1:

Graph2:

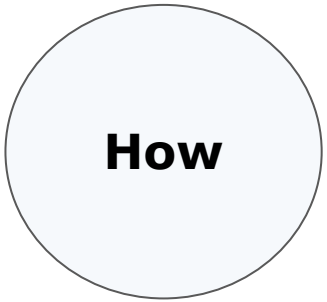
Graph Neural Networks



Graph1: `User 1`

Graph2: `User 9`

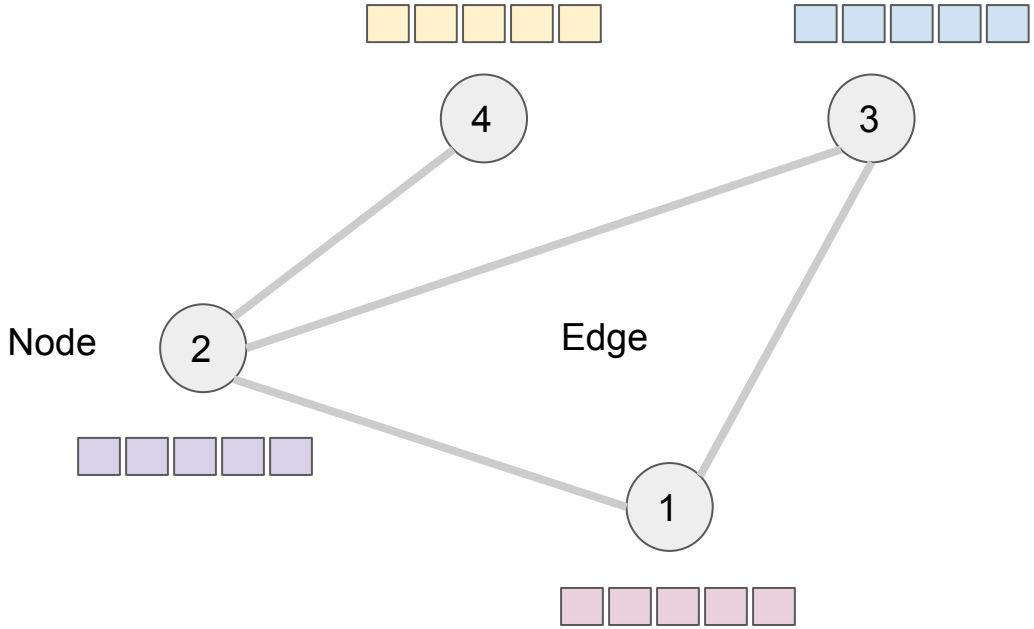
Basics of graph



Basics of graph

Graph structure basics

- Graph construction
- Graph notation
- Node feature matrix
- Adjacency matrix
- Degree matrix
- Laplacian matrix



Graph construction

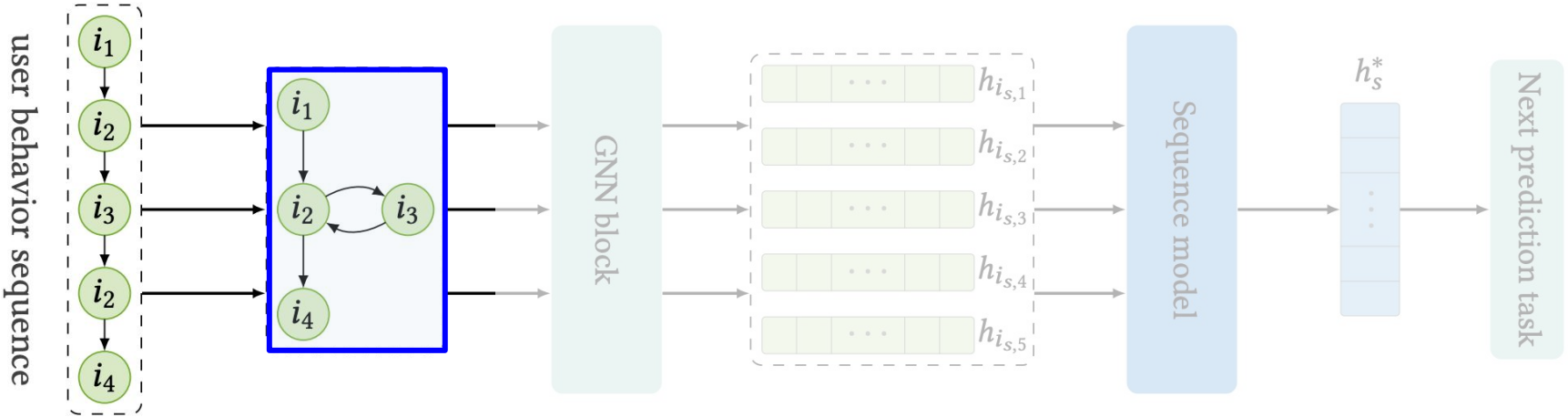
Relationship ***between*** data

Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2), 129-150.

Graph construction

Relationship ***between*** data

- Recommendation system



Wu, Shiwen, et al. "Graph neural networks in recommender systems: a survey." *ACM Computing Surveys* 55.5 (2022): 1-37.

Graph construction

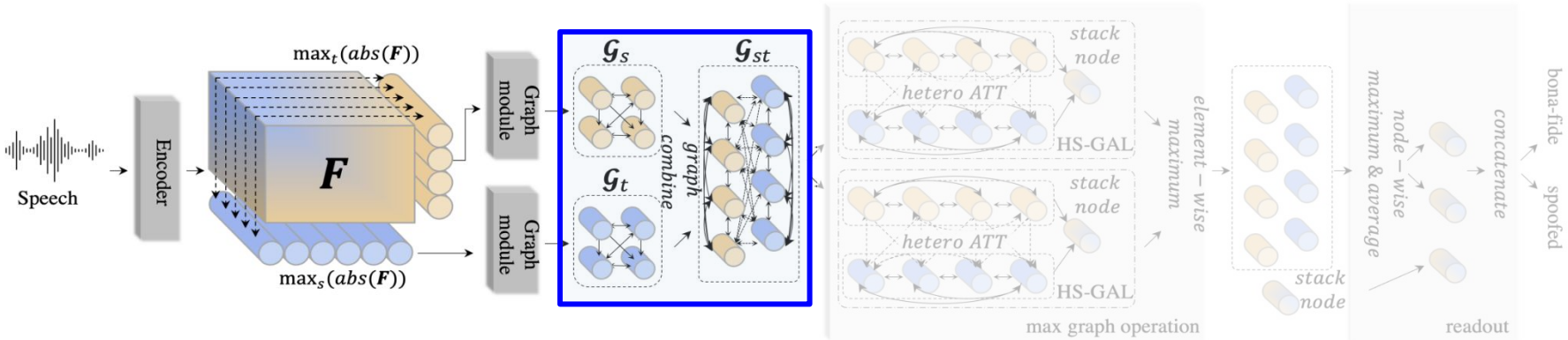
Relationship of features ***within*** data

Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2), 129-150.

Graph construction

Relationship of features ***within*** data

- Audio anti-spoofing



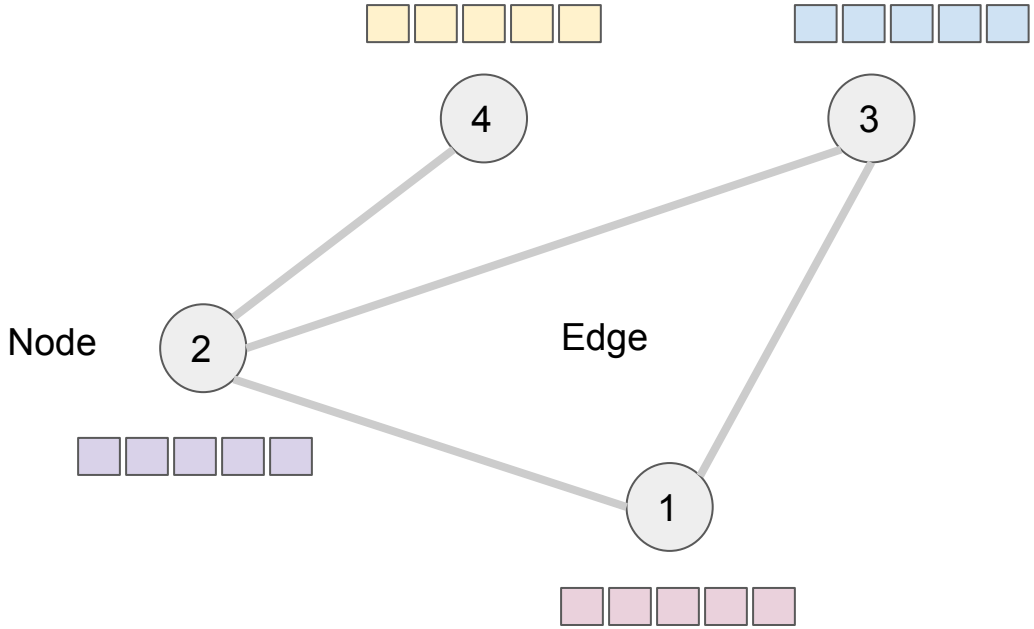
J. Jung, et al. "Aasist: Audio anti-spoofing using integrated spectro-temporal graph attention networks." in Proc. ICASSP, 2022.

Basics of graph

Graph notation

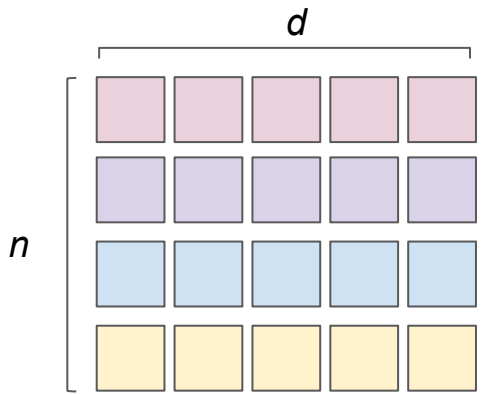
$V = \{1, 2, 3, 4\}$

$E = \{(1, 2), (1, 3), (2, 3), (2, 4)\}$

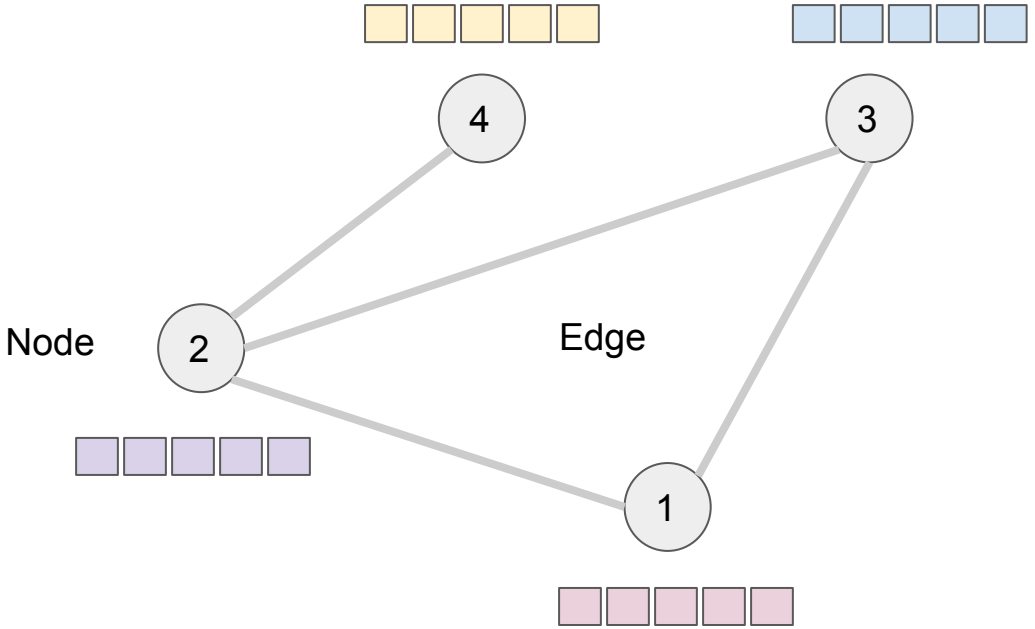


Basics of graph

Node feature matrix



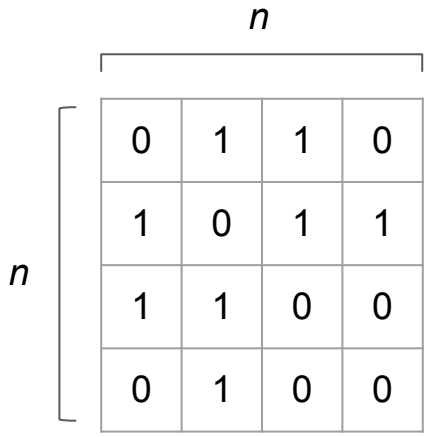
$$X \in R^{n \times d}$$



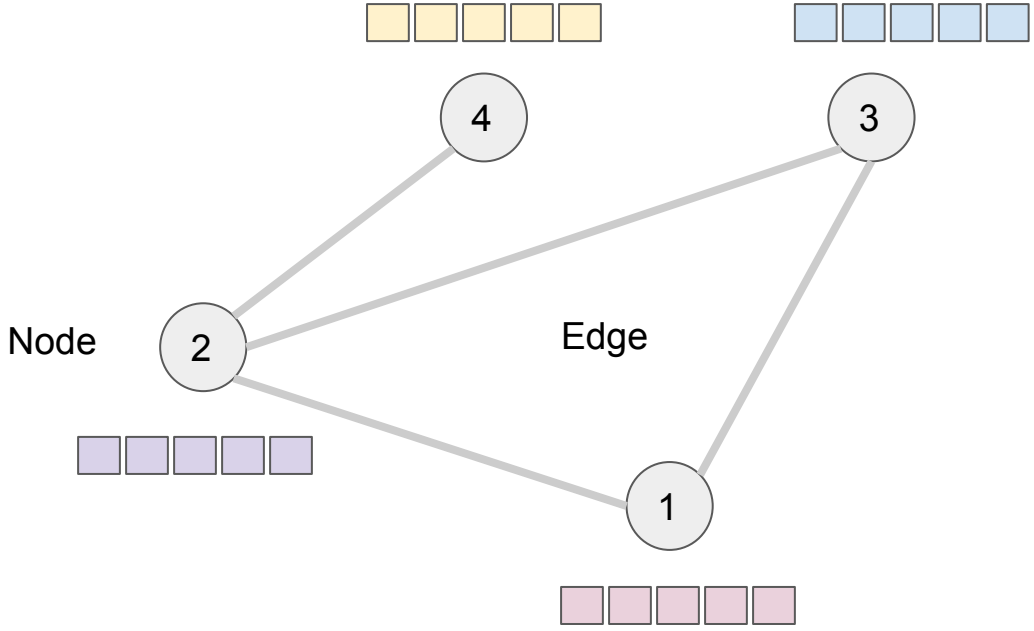
`represents node feature itself`

Basics of graph

Adjacency matrix



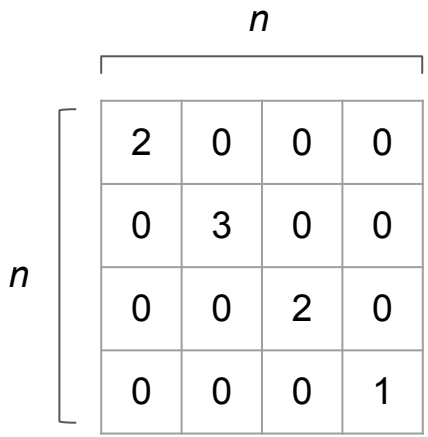
$$X \in A^{n \times n} \qquad A_{ij} = \begin{cases} 1 \; for \; edge \; (i,j) \\ 0 \; otherwise \end{cases}$$



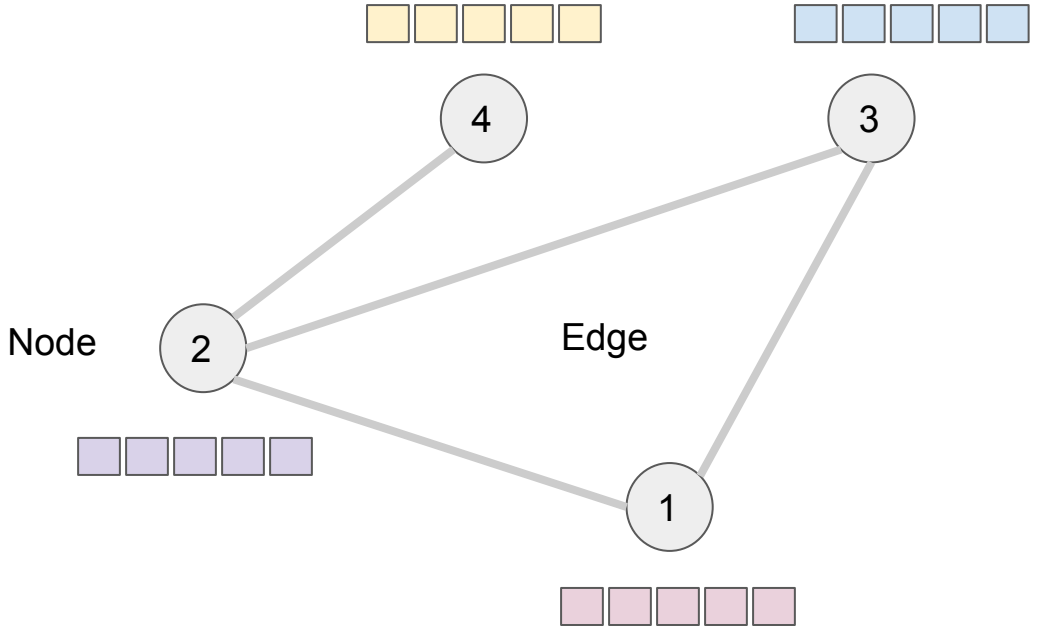
represents the connection/relationship between vertices

Basics of graph

Degree matrix



$$X \in D^{n \times n}$$



represents information about the degree of each vertex

Basics of graph

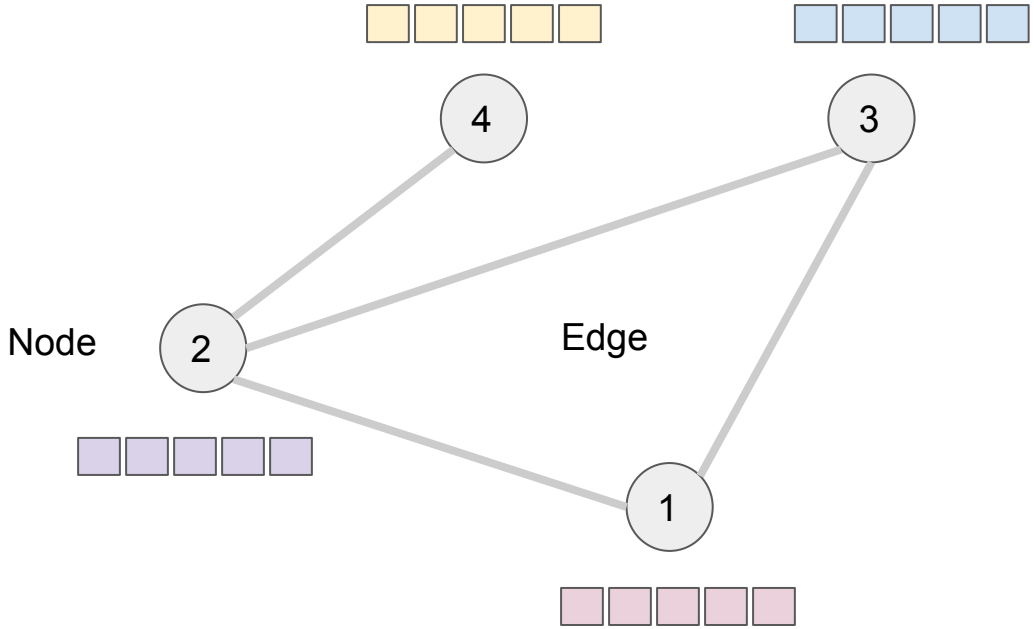
Laplacian matrix (D-A)

n

n

2	-1	-1	0
-1	3	-1	-1
-1	-1	2	0
0	-1	0	1

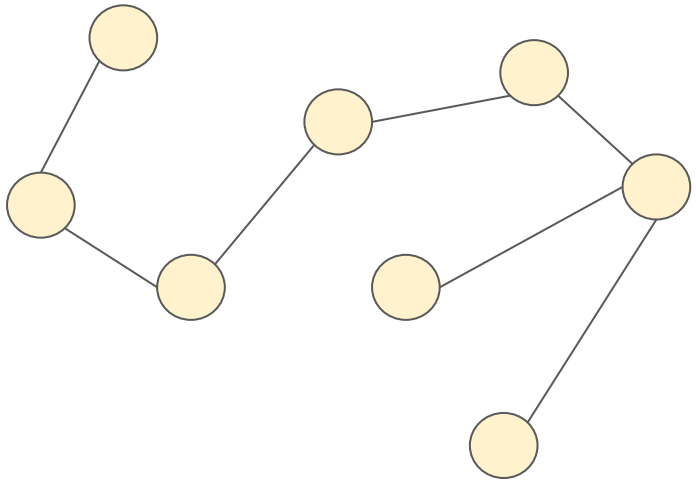
$$X \in L^{n \times n}$$



used for eigen decomposition

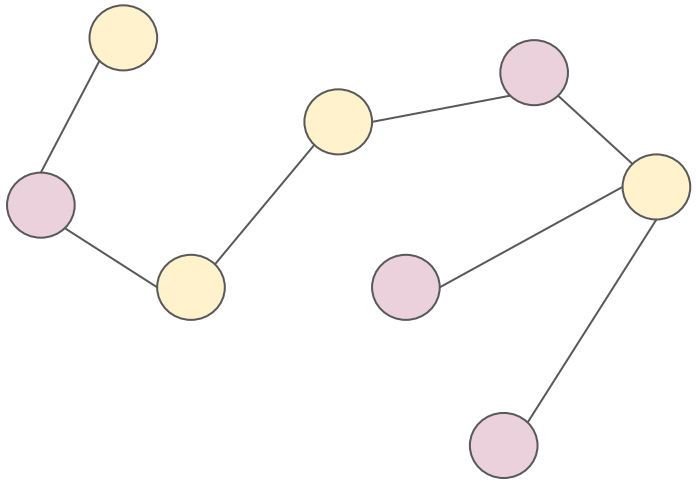
Types of graph

Homogeneous graph



Types of graph

Heterogeneous graph



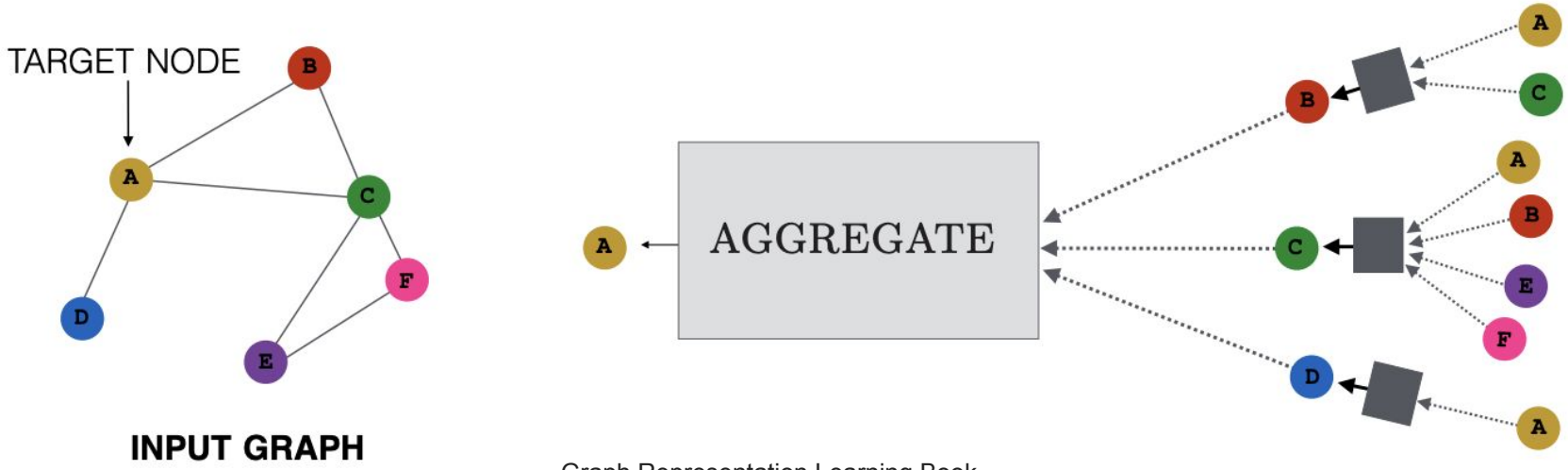
Basics of GNN operations

Graph operation basics

- Message passing
- Graph pooling
- Readout

Basics of GNN operations

Message passing

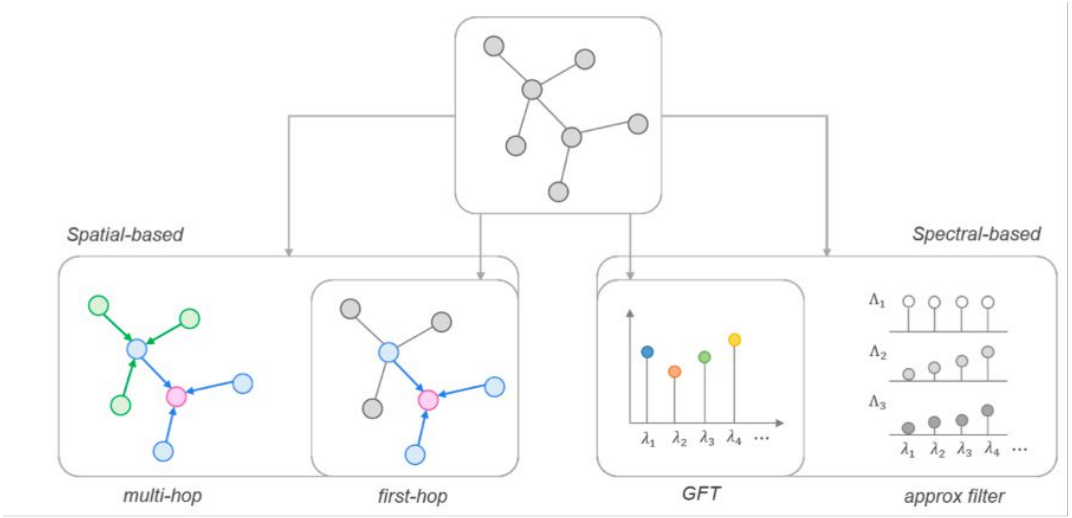


Graph Representation Learning Book

Basics of GNN operations

Message passing

- Graph Convolutional Network
  - Spectral
  - **Spatial (GAT)**

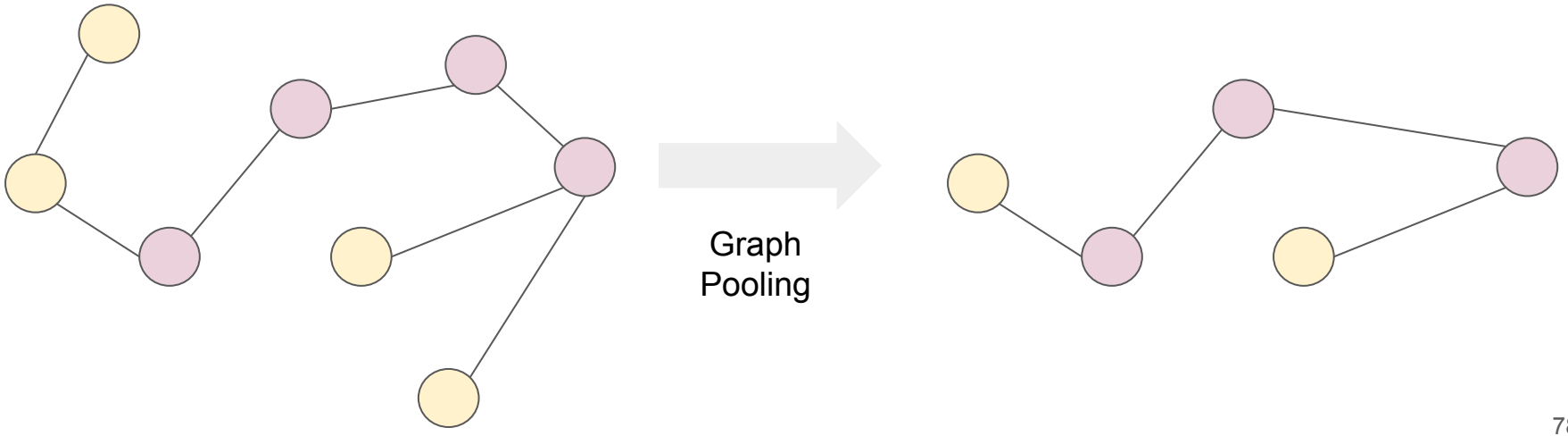


https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

Basics of GNN operations

Graph pooling

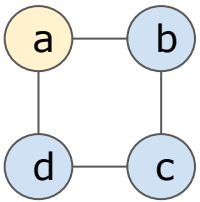
- Reduce the dimensionality of representations
- Avoid overfitting

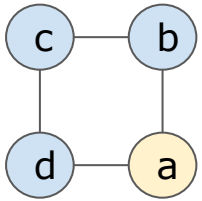


Basics of GNN operations

Graph pooling

- Graph specialized pooling
  - Graph architecture does not have fixed positions for its nodes
  - The conventional poolings for CNNs are difficult to apply to graph

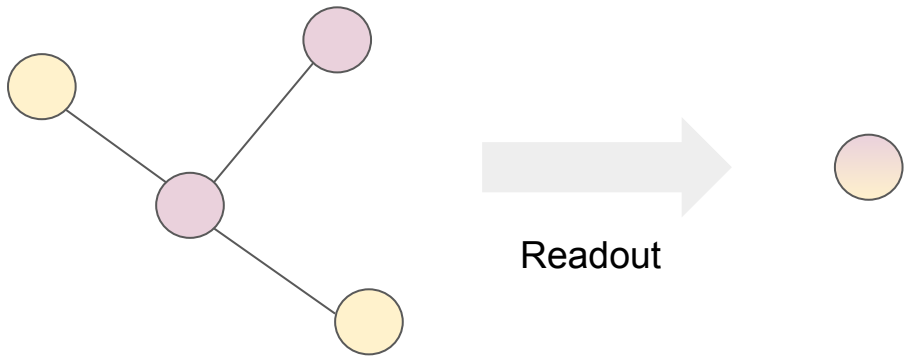




Basics of GNN operations

Readout operation

- Mean
- Max
- Min
- STD



Brief introduction of GAT

Graph attention networks (GAT) (Veličković 2018)

Graph neural networks **+ Attention mechanism**

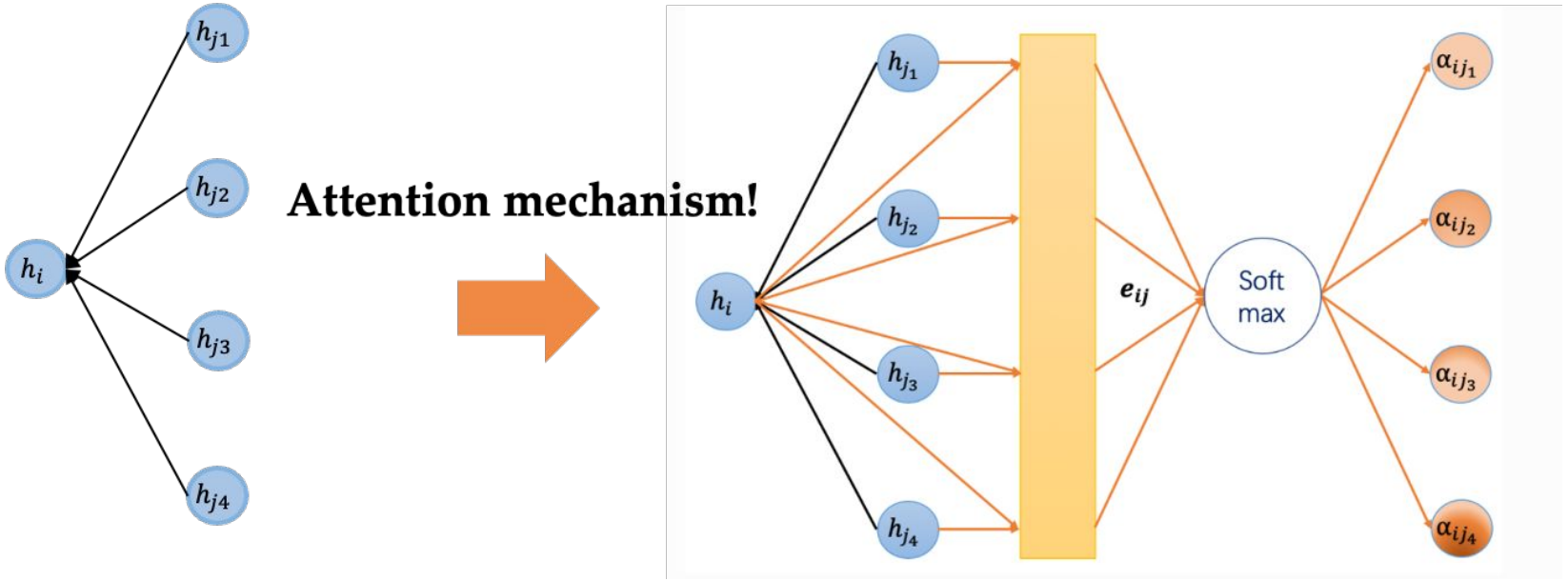
Higher score,

Closer node relationship

Veličković, Petar, et al. "Graph attention networks." In Proc. *ICLR* (2018).

Brief introduction of GAT

Graph attention networks



https://docs.dgl.ai/en/0.8.x/tutorials/models/1_gnn/9_gat.html

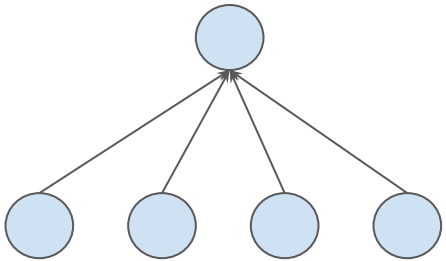
GCN vs GAT

GCN vs GAT

Difference : How the information is aggregated

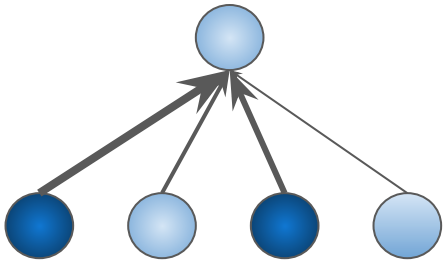
GCN

Normalized sum of the node features of neighbors



GAT

Weighted sum of the node features of neighbors

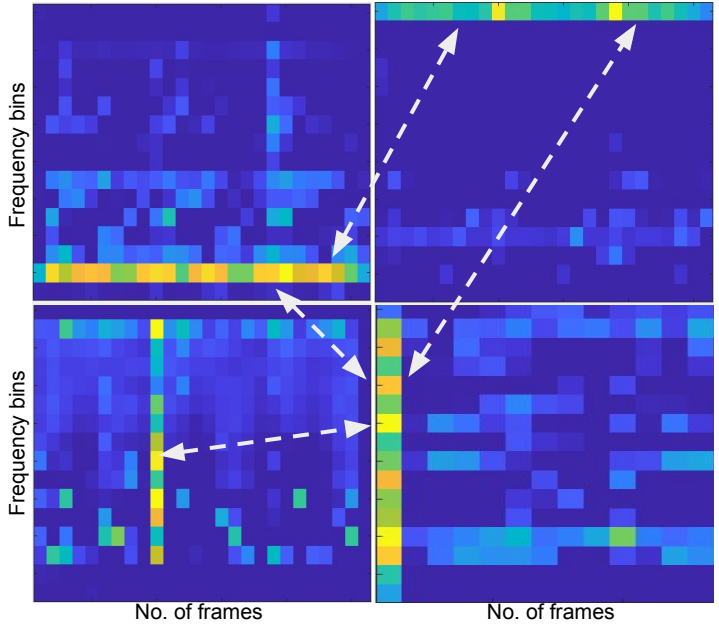


Session 1

- Introduction to speech anti-spoofing
- Introduction to graph attention networks
- **Graph attention networks for speech anti-spoofing**

GNN to model relationship between spectral and temporal representation

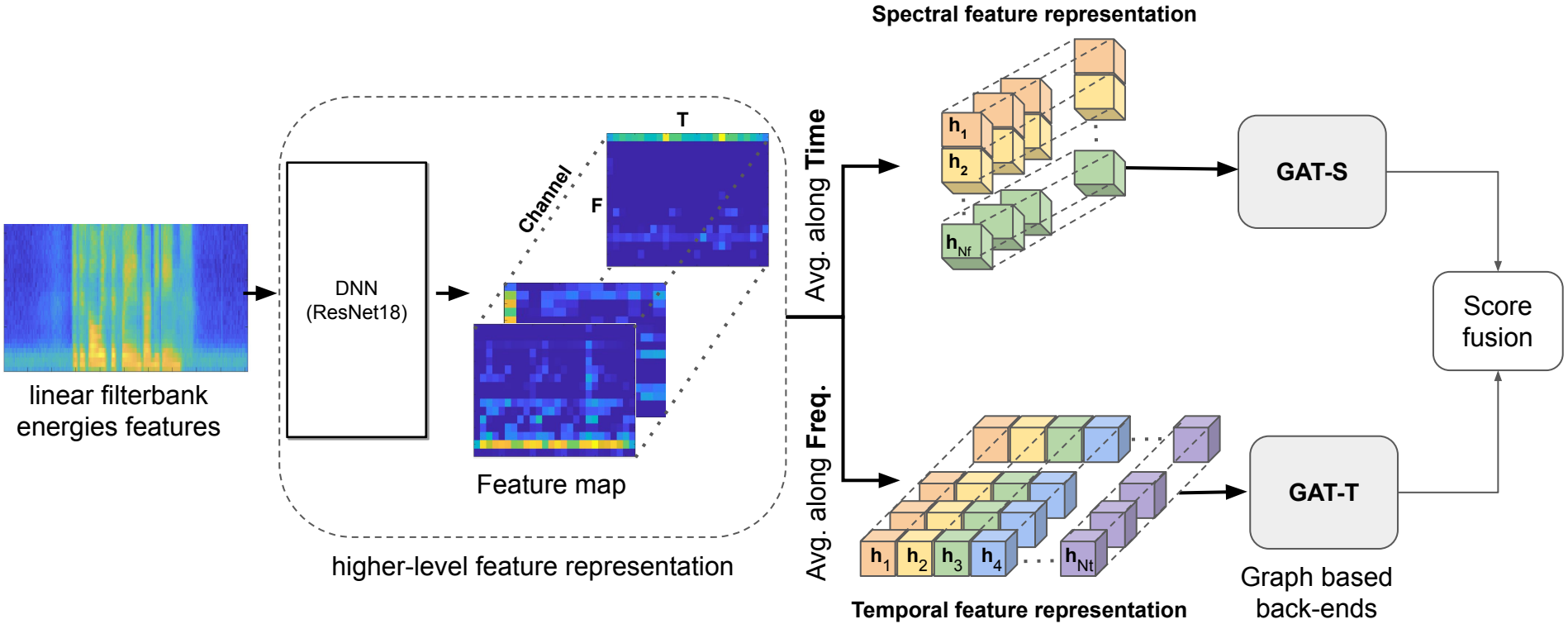
- Spoofing artefacts lie in **specific spectral subbands or temporal frames**[Yang 2019, Tak 2020, Chettri 2020,Tak 2021]
- modelling the relationship between the evidence spanning different **sub-bands and temporal intervals**
- to leverage the potential of **GNNs** for **modeling relationships** in spectral and temporal domain[Velickovic 2018]



feature-map representations from deep residual network

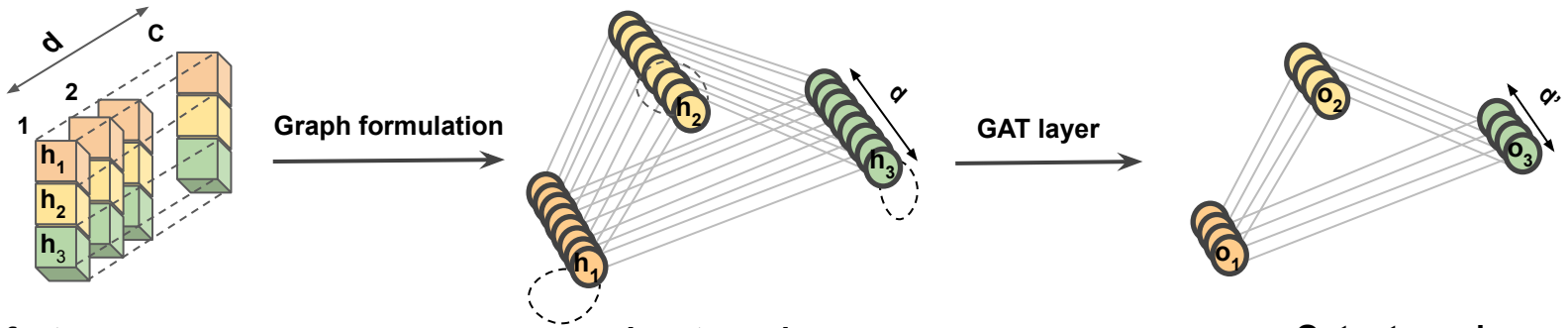
J. Yang et al., “Significance of subband features for synthetic speech detection,” IEEE Transactions on Information Forensics and Security, 2019.
H. Tak et al., “An explainability study of the constant Q cepstral coefficient spoofing countermeasure for automatic speaker verification,” in Proc. Speaker Odyssey, 2020.
B. Chettri et al., “Subband Modeling for Spoofing Detection in Automatic Speaker Verification,” in Proc. Speaker Odyssey Workshop, 2020.
H. Tak et al., “Spoofing Attack Detection using the Non-linear Fusion of Sub-band Classifiers,” in Proc. Interspeech, 2021.
P. Velickovic et al., “Graph attention networks,” in Proc. ICLR, 2018.

Graph attention network for anti-spoofing



H. Tak et al., “Graph attention network for anti-spoofing,” in Proc. Interspeech, 2021.

Graph construction



feature-map

Higher-dim feature representations

Input graph

$\mathcal{G}$ (N, $\mathcal{E}$, h)

Output graph

$\mathcal{G}'$(N, $\mathcal{E}$, o)

Input: a set of input node features

$h = \{ h_1, h_2, ..., h_N \}$, $h_n \in \mathbb{R}^d$, $n \in N$

Output: a set of new node features including neighboring node information

$o = \{ o_1, o_2, ..., o_N \}$, $o_n \in \mathbb{R}^{d'}$

$\mathcal{G}$ = fully-connected graph
N = set of nodes
$\mathcal{E}$ = edges between neighbouring node pairs
h = input node feature representation
d = input node feature dimensionality
o = output node feature representation
d' = target node feature dimensionality

GAT layer

GAT mainly includes 3 steps:

1) Compute attention weight: $$\alpha_{u,n} = \frac{\exp(W_{\text{att}}(\mathrm{h}_n \odot \mathrm{h}_u))}{\sum_{w \in \mathcal{M}(n) \cup \{n\}} \exp(W_{\text{att}}(\mathrm{h}_n \odot \mathrm{h}_w))}$$

where $W_{\text{att}} \in \mathbb{R}^{1 \times d'}$ is the learnable attention weights, $h_{\text{n}} \in \mathbb{R}^d$ and $h_{\text{u}} \in \mathbb{R}^d$ are neighboring node pairs.

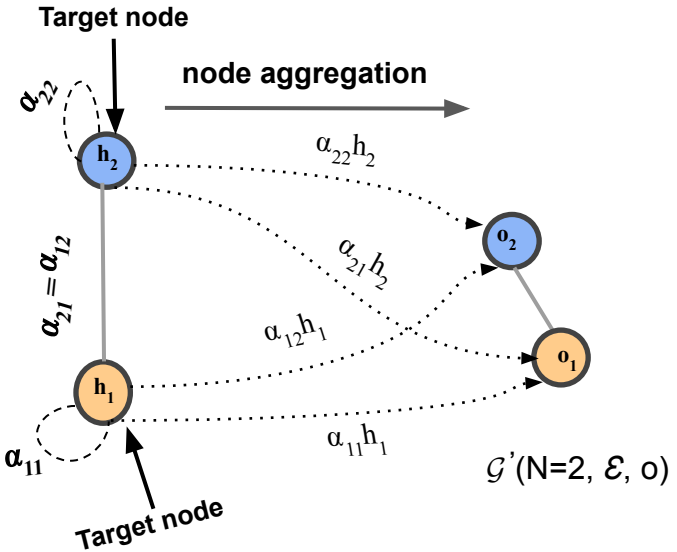
2) Node aggregation process: $$m_n = \sum_{u \in \mathcal{M}(n) \cup \{n\}} \alpha_{u,n} \mathrm{h}_u$$

where $\mathcal{M}(n)$ refers to the set of neighboring nodes for node n and $\alpha_{u,n}$ refers to the attention weight between nodes n and u.

3) Output node computation: $$\mathrm{o}_n = \mathrm{SeLU}(\mathrm{BN}(W_{\text{map}}(m_n) + W_{\text{res}}(\mathrm{h}_n)))$$

where $W_{\text{map}} \in \mathbb{R}^{d' \times d}$ is learnable linear transformation which projects the aggregated information for each node n to the target dimensionality d' using a fully-connected linear layer.

GAT setup example



$\mathcal{G}$ (N=2, $\mathcal{E}$, h)

$\mathcal{G}$'(N=2, $\mathcal{E}$, o)

$$\alpha_{u,n} = \frac{\exp(W_{\text{att}}(\mathrm{h}_n \odot \mathrm{h}_u))}{\sum_{w \in \mathcal{M}(n) \cup \{n\}} \exp(W_{\text{att}}(\mathrm{h}_n \odot \mathrm{h}_w))}$$

$h_{\mathrm{n}} = h_{\mathrm{u}} = h_{\mathrm{w}} \in \mathbb{R}^d$, where *d* (input node feature dim.) = 1

$W_{\text{att}} \in \mathbb{R}^{1 \times d'}$, where *d'* (output node feature dim.) = 1

---

n = 2 (h_2 node), u (neighboring node) = {1,2} and w = {1,2}

$$\alpha_{1,2} = \frac{\exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_1))}{\exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_1)) + \exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_2))}$$

$$\alpha_{2,2} = \frac{\exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_2))}{\exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_1)) + \exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_2))}$$

---

n = 1 (h_1 node) and u = {2,1}

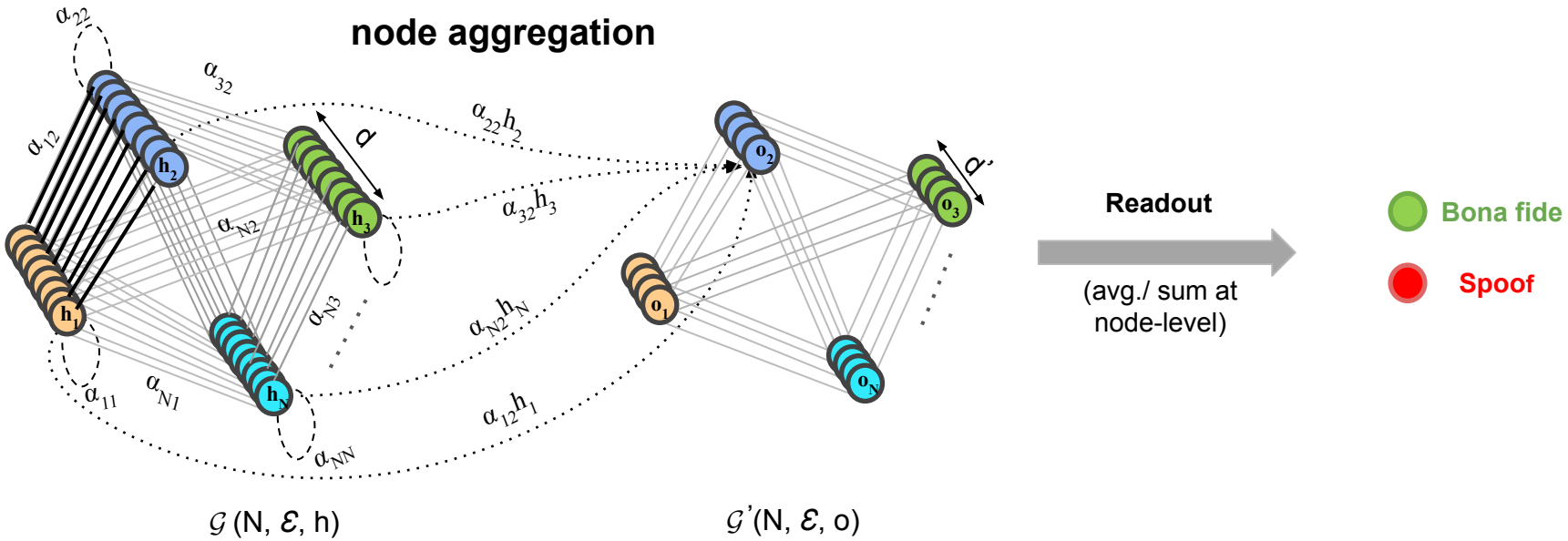
$\alpha_{2,1} = \alpha_{1,2}$

$$\alpha_{1,1} = \frac{\exp(W_{\text{att}}(\mathrm{h}_1 \odot \mathrm{h}_1))}{\exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_1)) + \exp(W_{\text{att}}(\mathrm{h}_2 \odot \mathrm{h}_2))}$$

Node aggregation → $m_n = \sum_{u \in \mathcal{M}(n) \cup \{n\}} \alpha_{u,n} \mathrm{h}_u \rightarrow m_1 = \alpha_{2,1}\mathrm{h}_2 + \alpha_{1,1}\mathrm{h}_1 \quad m_2 = \alpha_{1,2}\mathrm{h}_1 + \alpha_{2,2}\mathrm{h}_2$

Output nodes → $\mathrm{o}_1 = \mathrm{SeLU}(\mathrm{BN}(W_{\text{map}}(m_1) + W_{\text{res}}(\mathrm{h}_1))) \quad \mathrm{o}_2 = \mathrm{SeLU}(\mathrm{BN}(W_{\text{map}}(m_2) + W_{\text{res}}(\mathrm{h}_2)))$

GAT framework



Results

Performance comparisons using ASVspoof 2019 LA dataset

CM Systems	Pooled min t-DCF	Pooled EER (%)
GAT-temporal (GAT-T)	0.089	4.71
GAT-spectral (GAT-S)	0.091	4.48
Score fusion	**0.084**	**4.05**
Resnet18-SP	0.114	6.82
ResNet18-ASP	0.127	6.22
Resnet18-SAP	0.138	7.11

Comparison with other conventional attention mechanisms

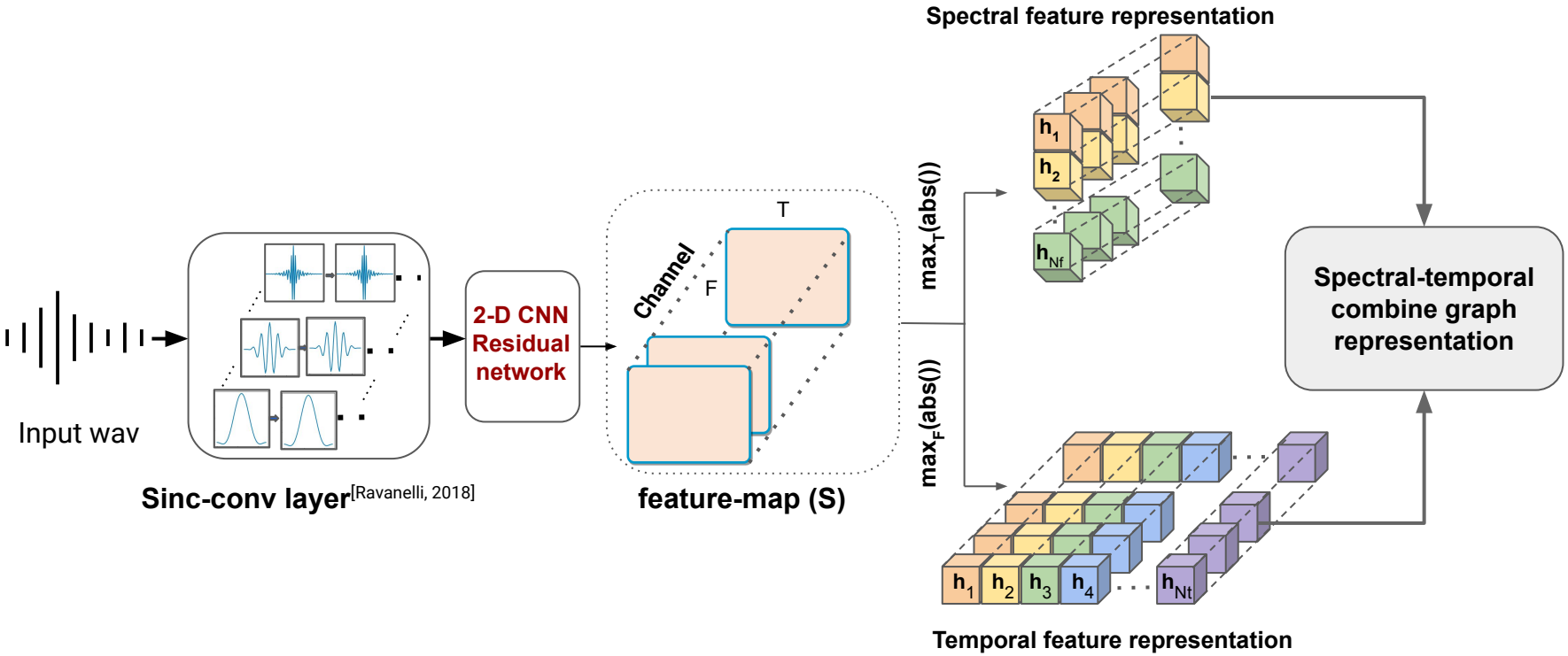
- **Limitations**
  - spectral and temporal relationship is separated → no communication

a single E2E graph model might be useful

SP[Snyder 2017] : Statistical pooling
ASP[Okabe 2018] : Attentive statistical pooling
SAP[Okabe 2018] : Self-attentive pooling

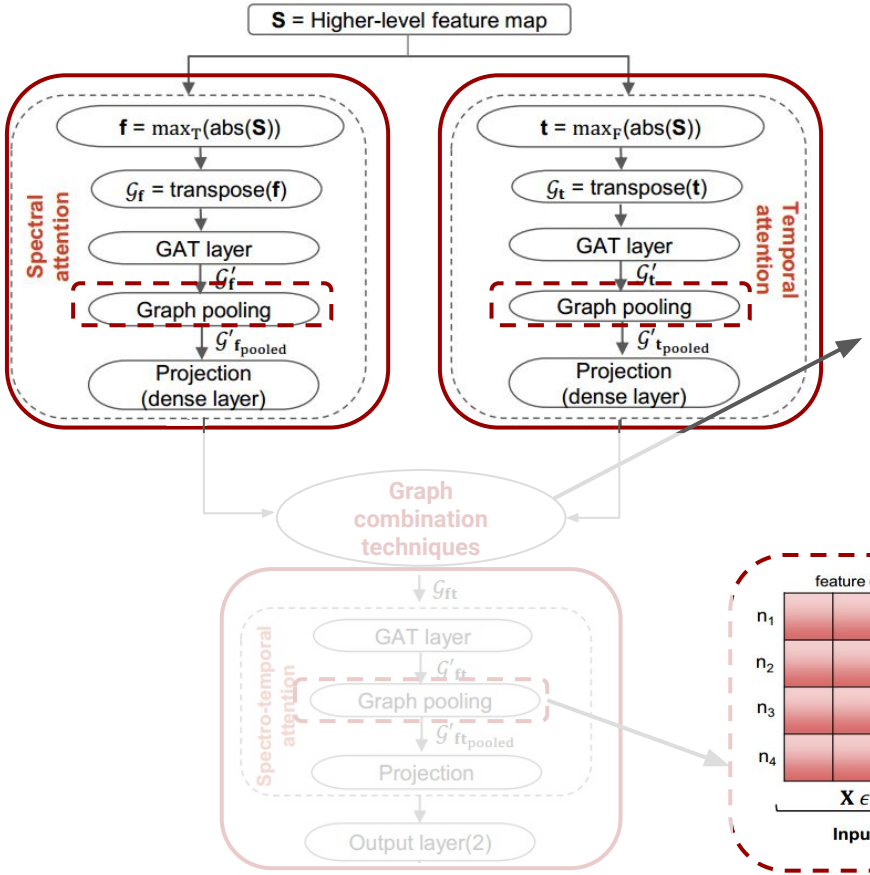
D. Snyder et al., “Deep neural network embeddings for text-independent speaker verification.” in Proc. INTERSPEECH, 2017.
K. Okabe et al., “Attentive statistics pooling for deep speaker embedding,” in Proc. INTERSPEECH, 2018.
K. He et al., “Deep residual learning for image recognition,” in Proc. CVPR, 2016.
H. Tak et al., “Graph attention network for anti-spoofing,” in Proc. INTERSPEECH, 2021.

End-to-end spectro-temporal graph attention network



M. Ravanelli et al., “Speaker Recognition from Raw Waveform with SincNet,” in Proc. SLT 2018.

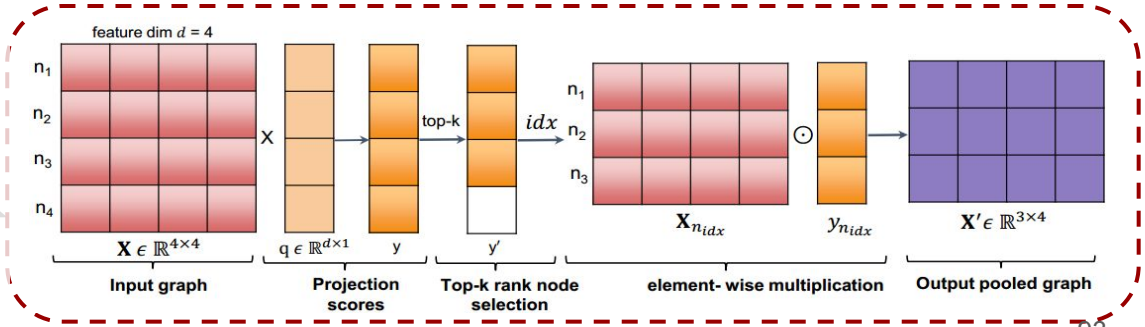
Spectro-temporal GAT (RawGAT-ST)



Standard graph combination techniques in GNN's

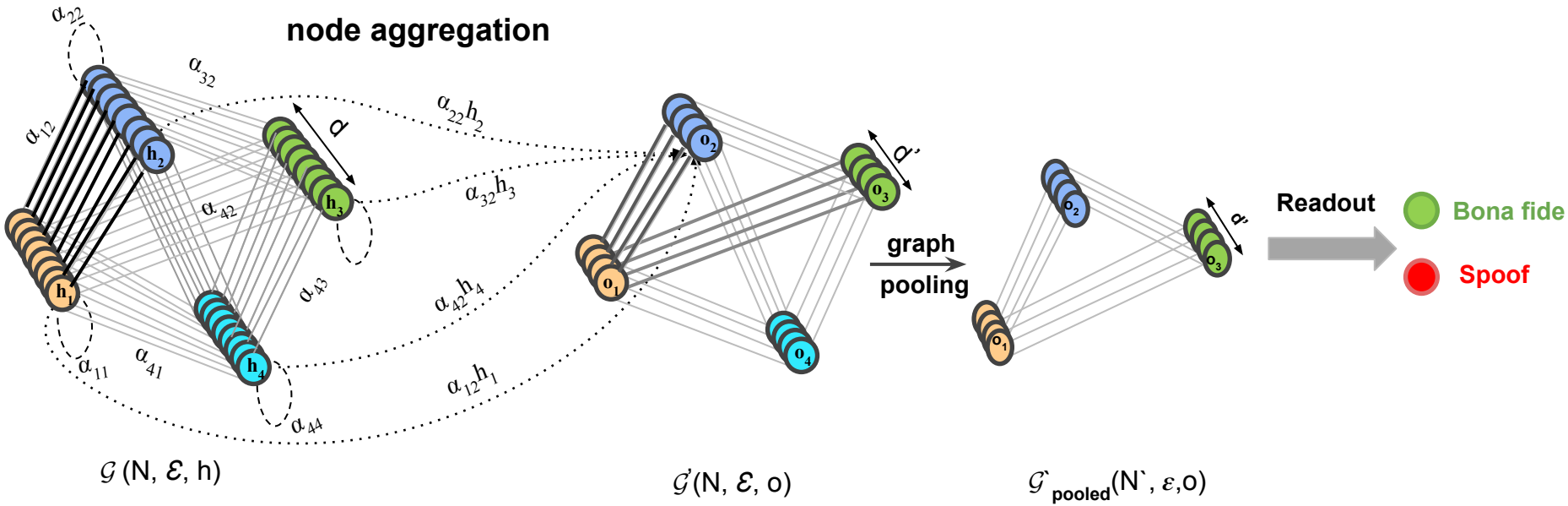
1. Element-wise multiplication (GAT-S ⊙ GAT-T)
2. Element-wise addition (GAT-S ⊕ GAT-T)
3. Concatenation (GAT-S || GAT-T)

Graph pooling layer[Gao 2019]



H. Gao et al., "Graph u-nets," in international conference on machine learning (PMLR), 2019.

RawGAT-ST framework

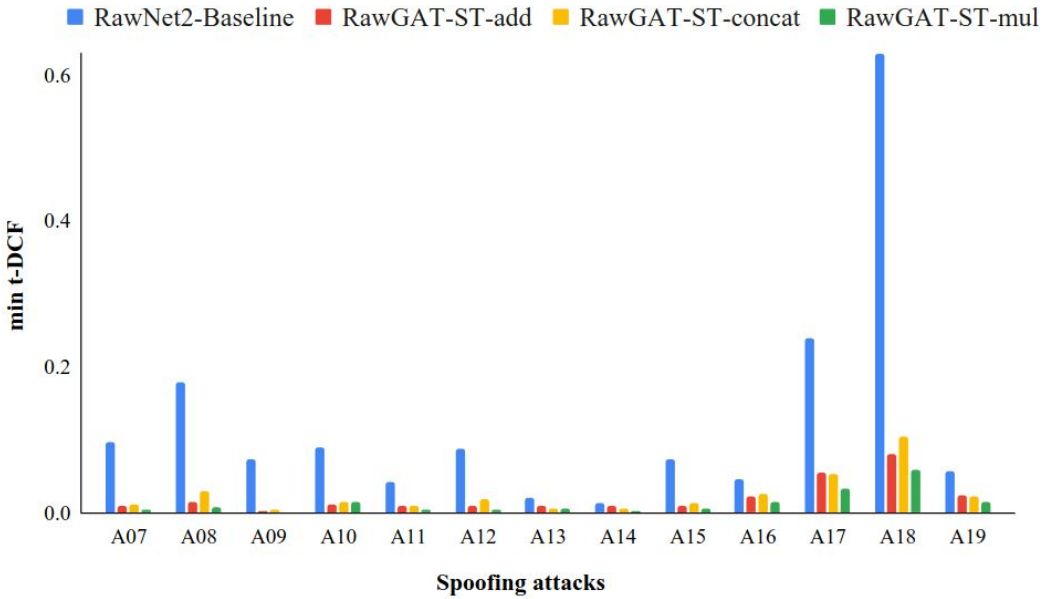


H. Tak et al., "End-to-End Spectro-Temporal Graph Attention Networks for Speaker Verification Anti-Spoofing and Speech Deepfake Detection," in ASVspoof 2021 workshop, 2021.

Results

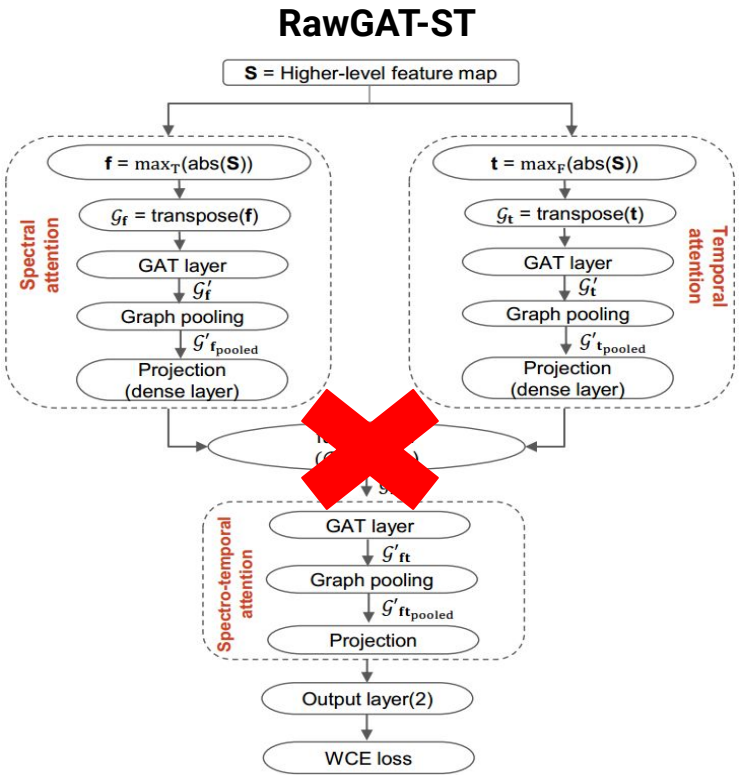
Performance using ASVspoof 2019 LA dataset

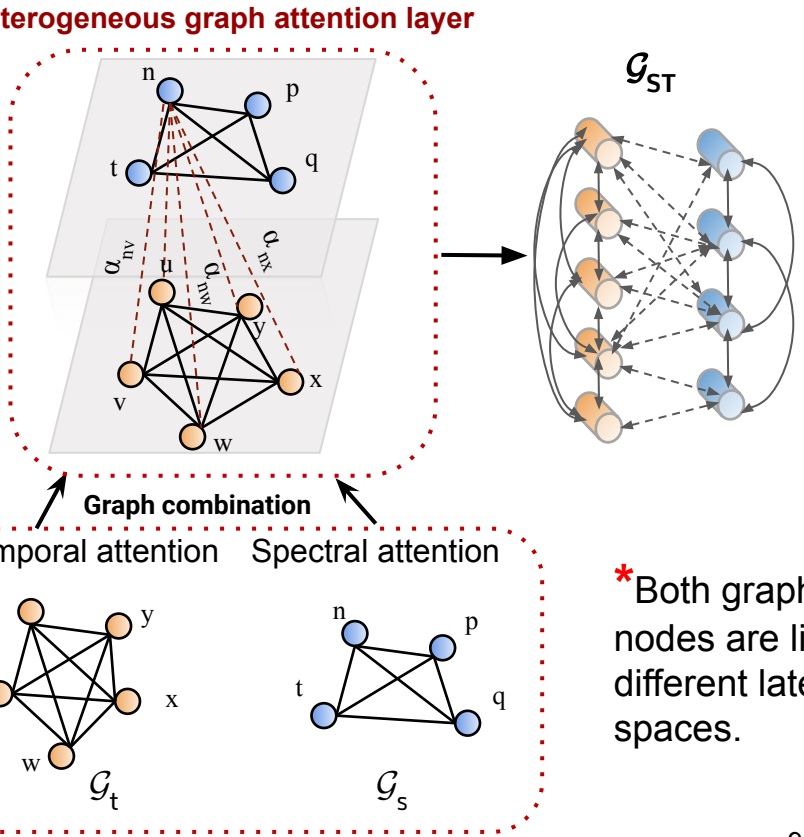
CM systems	min t-DCF	EER (%)
RawNet2[Tak 2022]	0.1547	5.54
GAT-S+GAT-T (score-fusion)	0.084	4.05
RawGAT-ST-mul	**0.0335**	**1.06**
RawGAT-ST-add	0.0373	1.15
RawGAT-ST-concat	0.0388	1.23



H. Tak et al.,"End-to-end anti-spoofing with RawNet2," in Proc. ICASSP, 2021.

Motivation for integrated spectro-temporal graph attention network

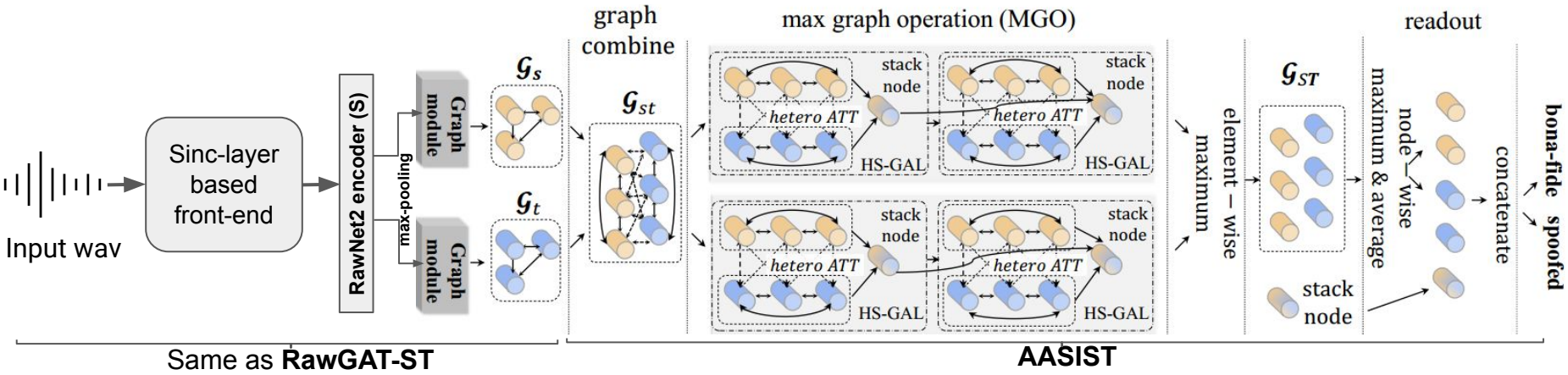




*Both graph nodes are lie in different latent spaces.

X. Wang et al., "Heterogeneous Graph Attention Network," in The World Wide Web Conference, 2019.

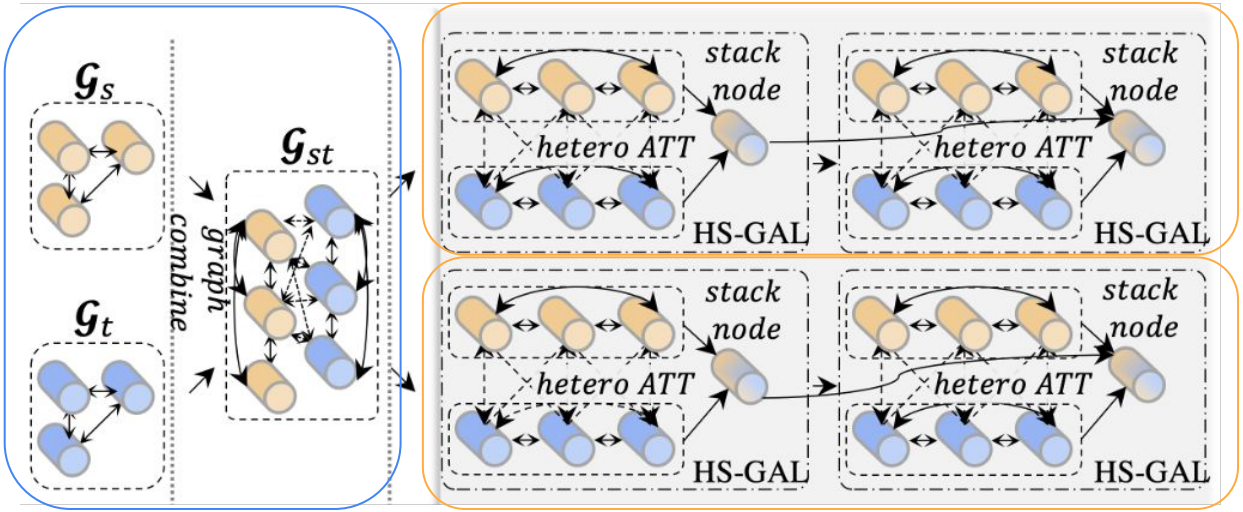
An integrated spectro-temporal graph attention network (AASIST)



X. Wang et al., “Heterogeneous Graph Attention Network,” in The World Wide Web Conference, 2019.
J. Jung et al., “AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks,” in proc. ICASSP, 2022.

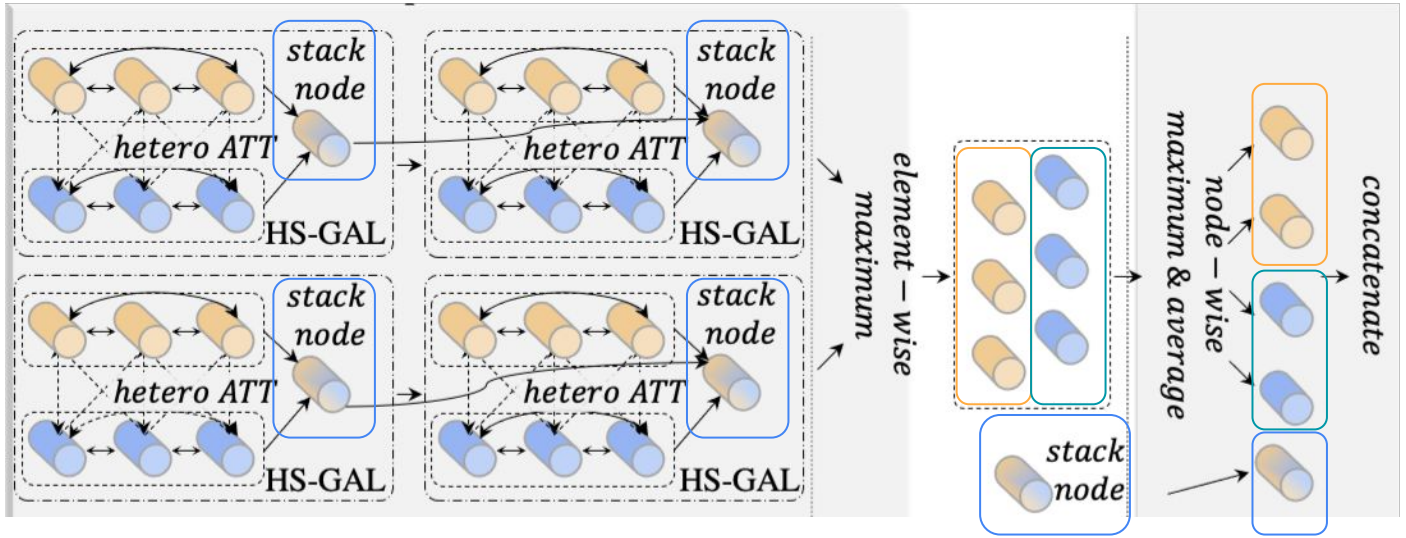
An integrated spectro-temporal graph attention network (AASIST)

- **Graph combination**: construct one spectral and one temporal graphs → combine
- **Heterogeneous extension of GAT:** projection → heterogeneous attention map → node aggregation
- **Max graph operation**: two identical branches (separate weights) → element-wise maximum of two outputs



Techniques in AASIST

- **Stack node**: receives information from all other nodes, then passed on
- **Max & average pool readout with stack node**: concatenate five vectors – max & average × 2 + stack node



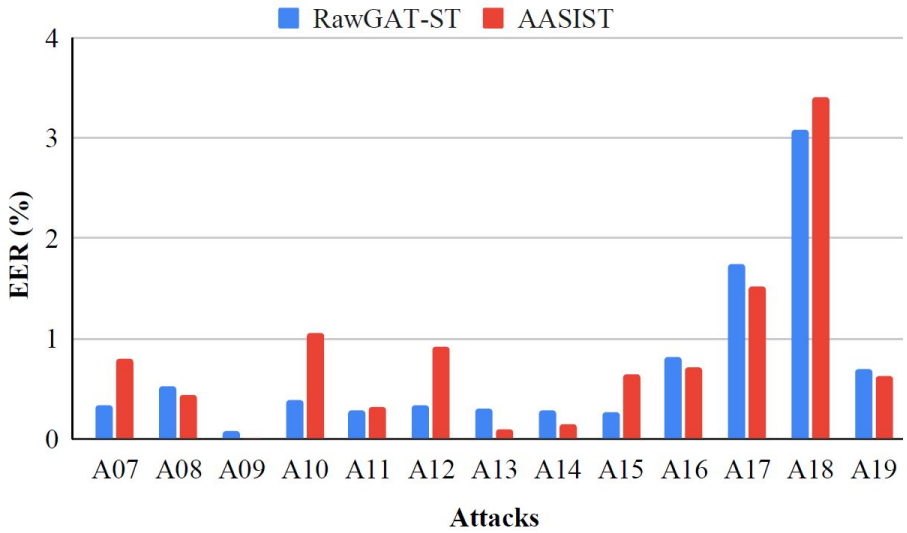
Results

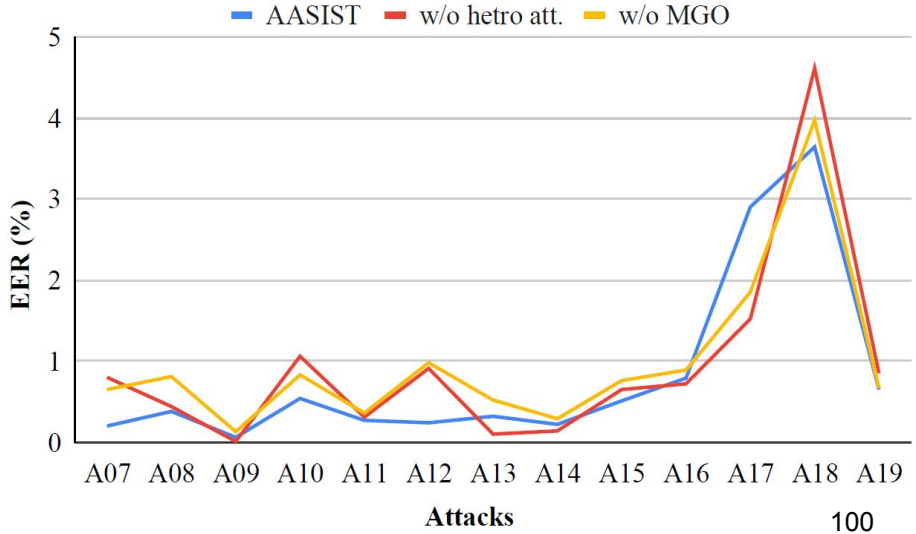
Performance comparison using ASVspoof 2019 LA dataset

CM systems	min t-DCF	EER (%)
RawGAT-ST	0.0335	1.06
AASIST	**0.0270**	**0.83**

Ablation study

CM systems	min t-DCF	EER (%)
w/o heterogenous att. layer	0.0384	1.37
w/o MGO	0.0378	1.19



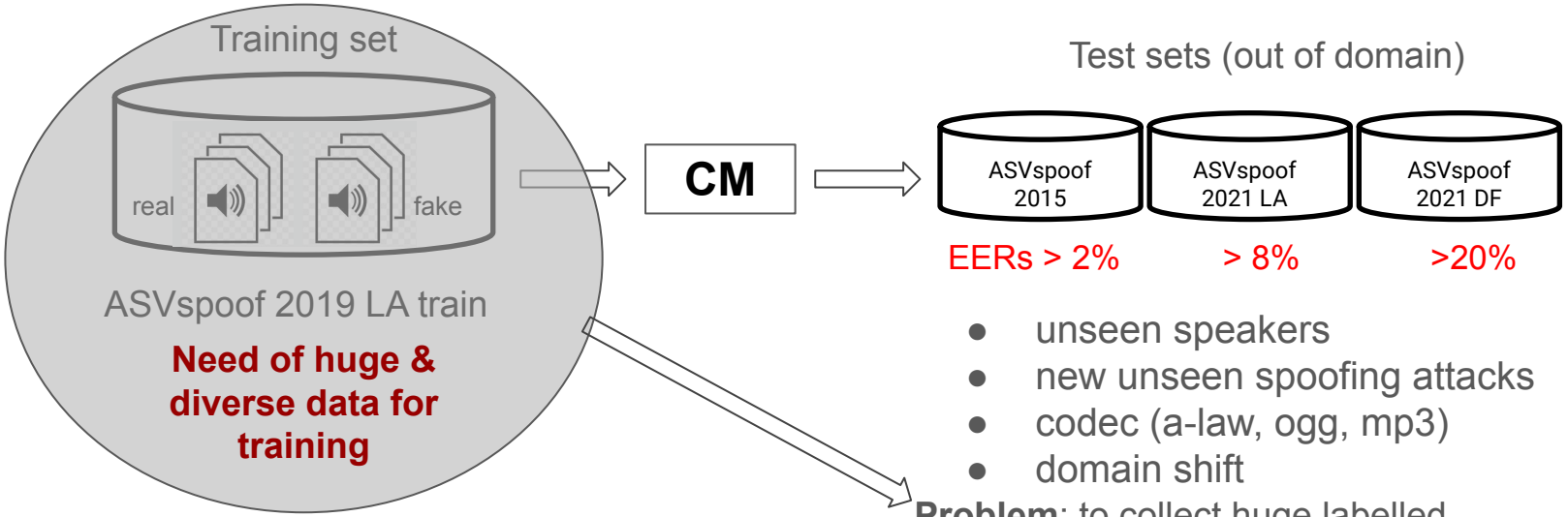


J. Jung et al., “AASIST: Audio Anti-Spoofing using Integrated Spectro-Temporal Graph Attention Networks,” in proc. ICASSP, 2022.

Session 2

- Introduction to speech anti-spoofing
- Introduction to graph attention networks
- Graph attention networks for speech anti-spoofing
- **Self-supervised learning for speech anti-spoofing**

CM generalisation



- unseen speakers
- new unseen spoofing attacks
- codec (a-law, ogg, mp3)
- domain shift

Problem: to collect huge labelled data is time-consuming

Solution : many **self-supervised (SSL) pre-trained models** are available and can be used off the shelf for anti-spoofing

Questions:

Does CM generalise across different databases ?

How to improve generalisation ?

D. Paul et al., “Generalization of Spoofing Countermeasures: A Case Study with ASVspoof 2015 and BTAS 2016 Corpora,” in Proc. ICASSP 2017.
R. K Das et al., “Assessing the Scope of Generalized Countermeasures for Anti-Spoofing,” in Proc. ICASSP 2020.
N. M. Muller et al., “Does Audio Deepfake Detection Generalize?”, in Proc. INTERSPEECH 2022.

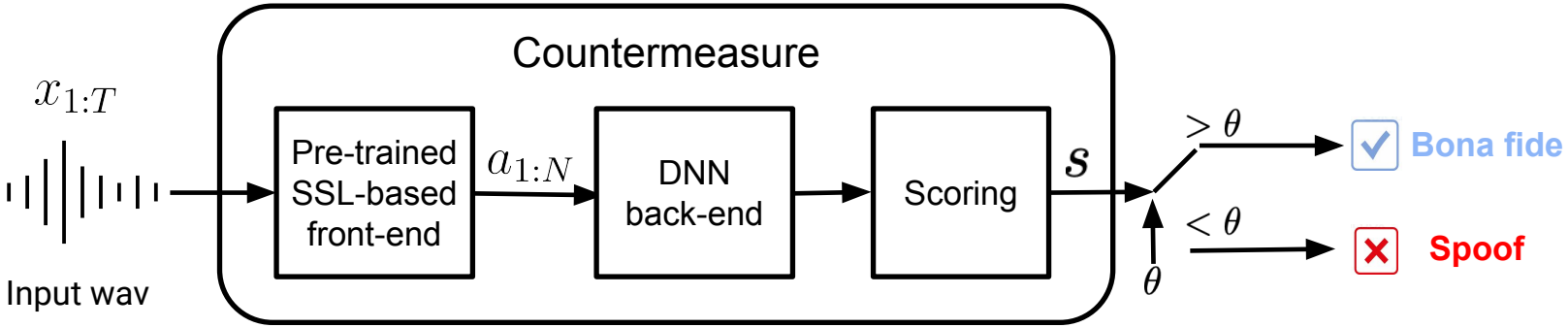
Self-supervised learning

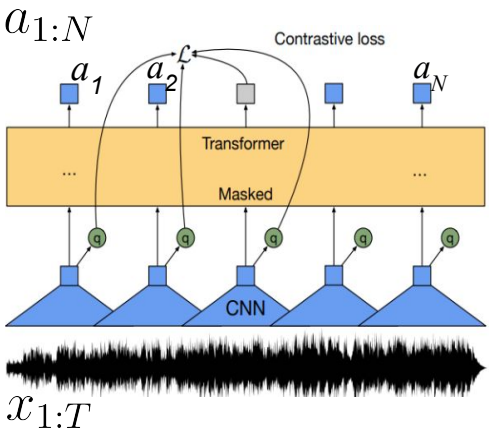
- **What** is it ?
  - Self-supervised learning (SSL) uses information from input data as the label to learn feature representations useful for downstream tasks

- **Why** do we need it ?
  - The age of “representation learning”! (Pre-training - Fine tuning pipeline)
  - To improve feature representation for CM generalisation
  - To reduce domain-shift when in-domain data is limited

- **How** can we use it for anti-spoofing ?
  - That is in this tutorial

SSL-based countermeasure





Wav2vec 2.0

$x_{1:T}$ — Transformer (CNN ($x_{1:T}$)) → $a_{1:N}$

$(x_1, x_2, ..., x_T) \in \mathbb{R}^{1xT}$

$(a_1, a_2, ..., a_N) \in \mathbb{R}^{D \times N}$

T = temporal segments

D = SSL pre-trained model dim. (1024)

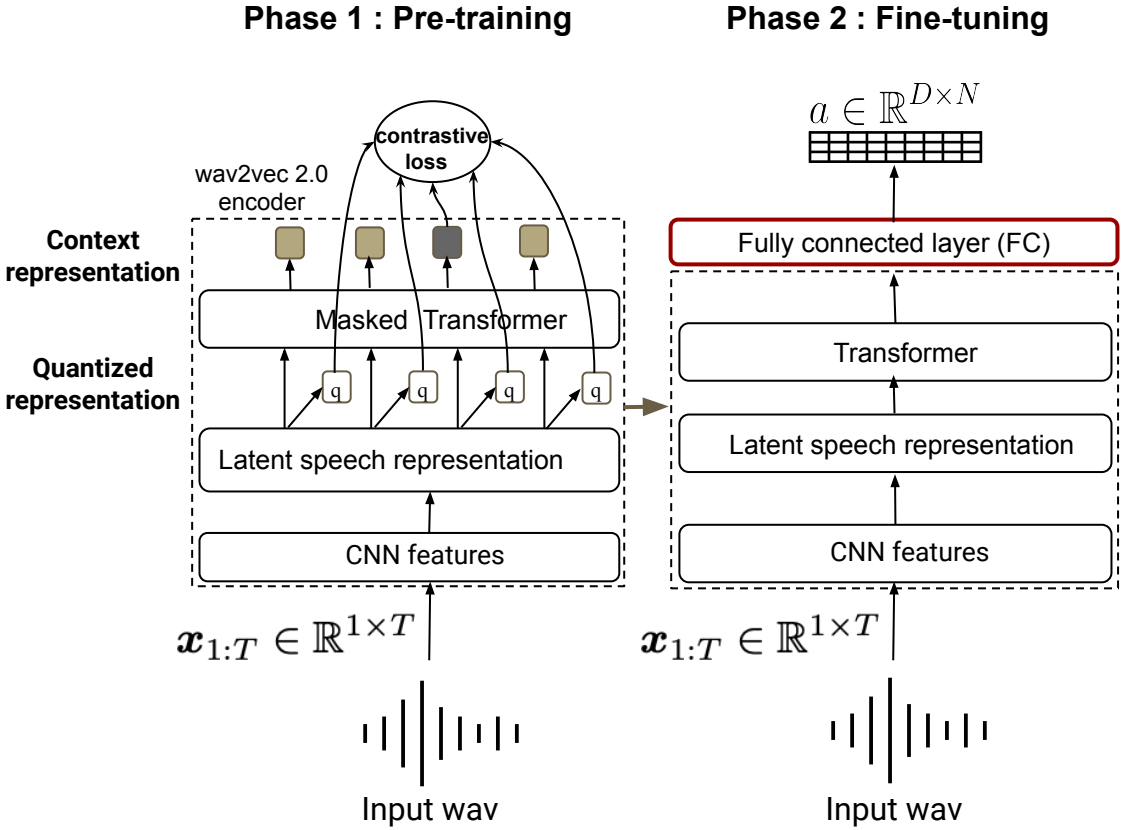
N = no. of frames (20 ms CNN stride)

SSL framework

Two stages in the framework:

1. Use SSL to pre-train an upstream model for general purpose task.

2. Downstream task uses the learned representation from a pre-trained model (fixed) or fine-tune the pre-trained model for specific downstream task.



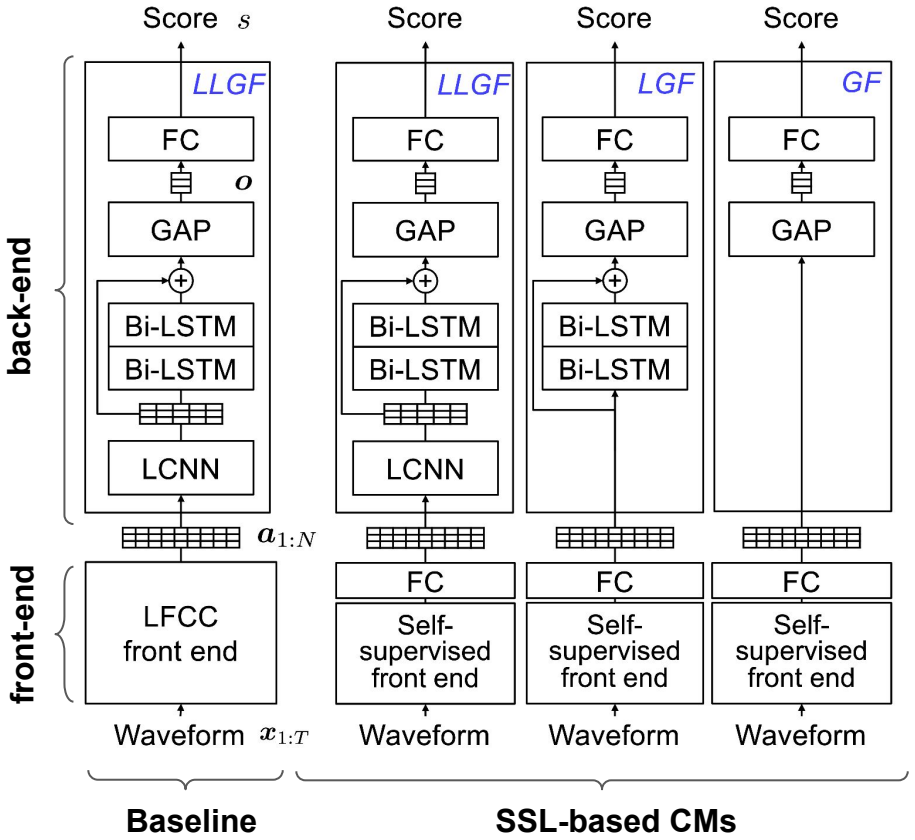
A. Baevski et al., “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” in Proc. NeurIPS 2020.

CM configuration

- Baseline[Wang 2021]
  LLGF → LCNN + LSTM + GAP (global average pooling) + FC (linear layer)
- SSL front-end: **wav2vec 2.0 XLSR**[Baevski 2020]
- **Different back-ends** setup for SSL front-end

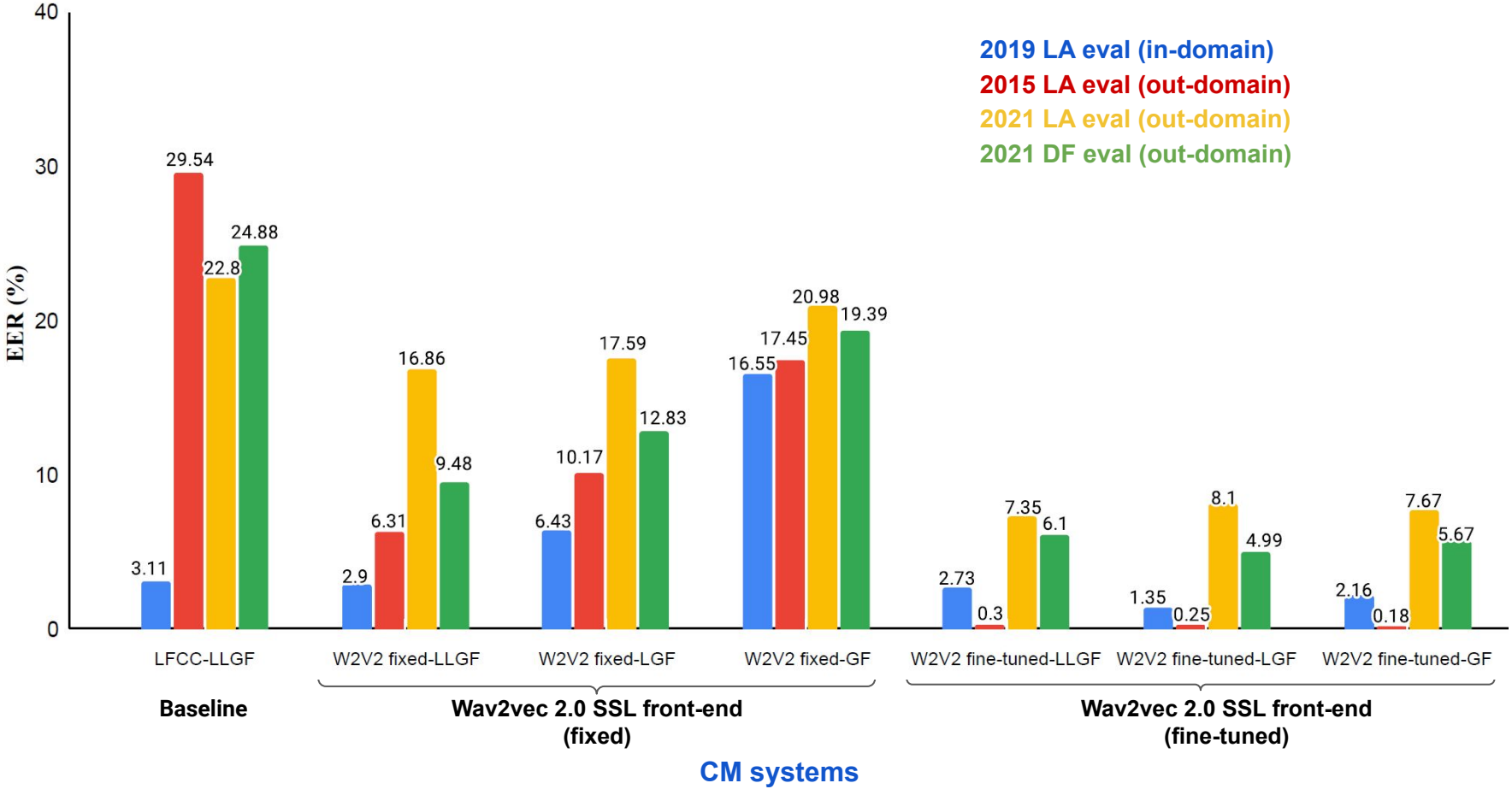
Complex to simple back-ends

LLGF → LCNN+LSTM+GAP+FC
LGF → LSTM+GAP+FC
GF → GAP+FC



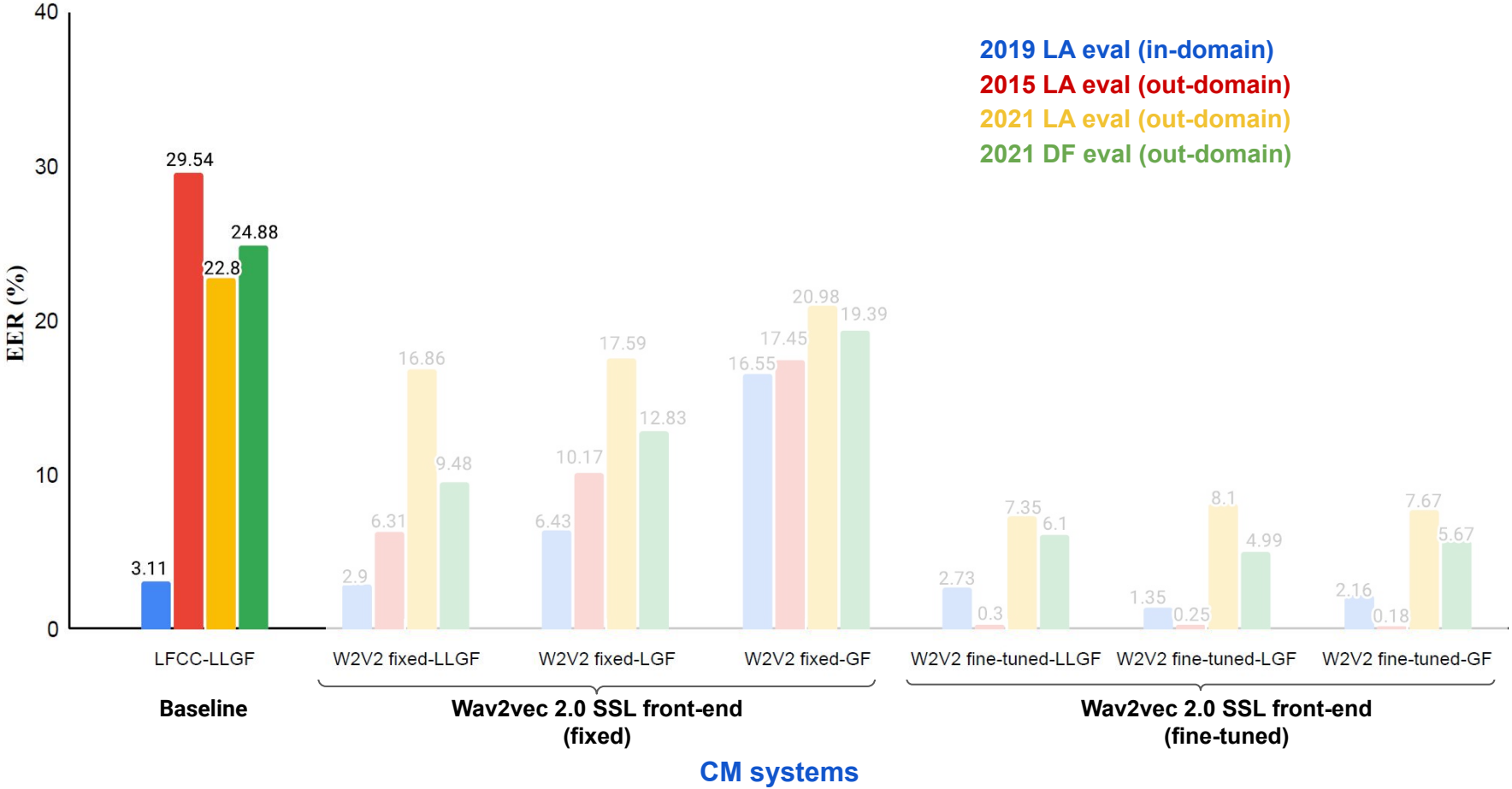
X. Wang et al., "A comparative study on recent neural spoofing countermeasures for synthetic speech detection," in Proc. Interspeech, 2021.
A. Baevski et al., "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in Proc. NeurIPS 2020.
X. Wang et al., "Investigating self-supervised front ends for speech spoofing countermeasures," in Proc. Speaker Odyssey, 2022.

Results



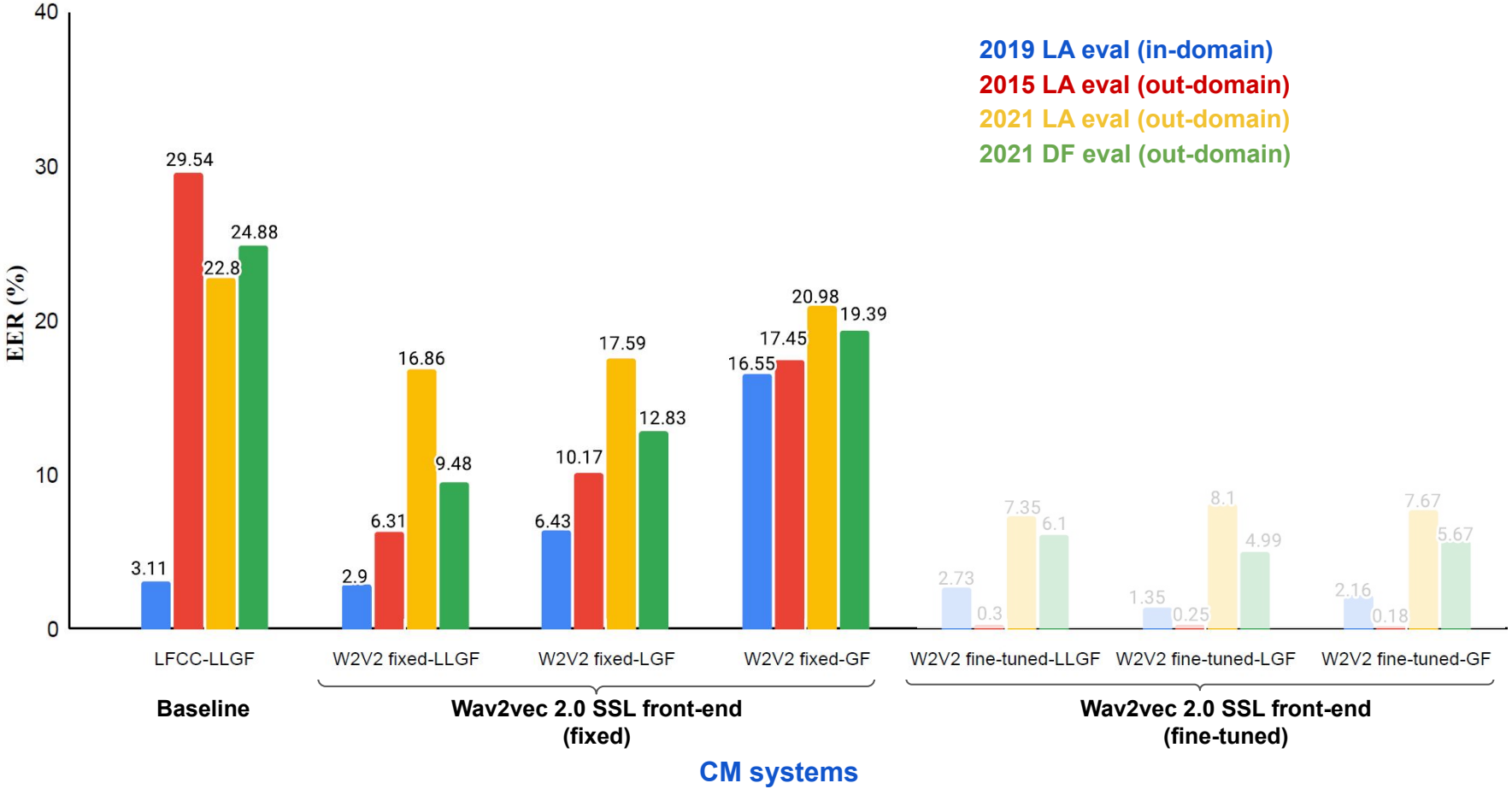
X. Wang et al., “Investigating self-supervised front ends for speech spoofing countermeasures,” in Proc. Speaker Odyssey, 2022.

Results



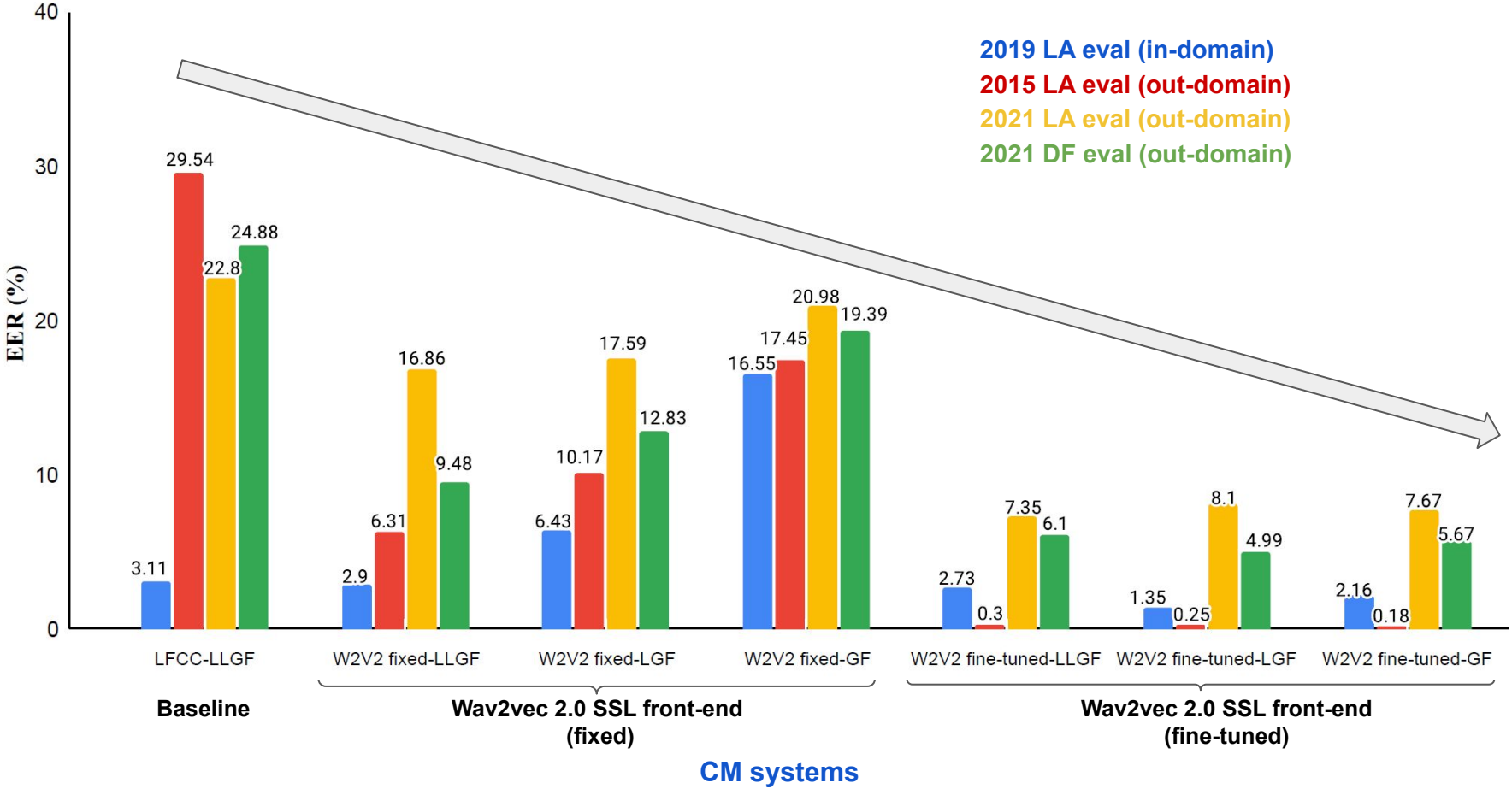
X. Wang et al., “Investigating self-supervised front ends for speech spoofing countermeasures,” in Proc. Speaker Odyssey, 2022.

Results



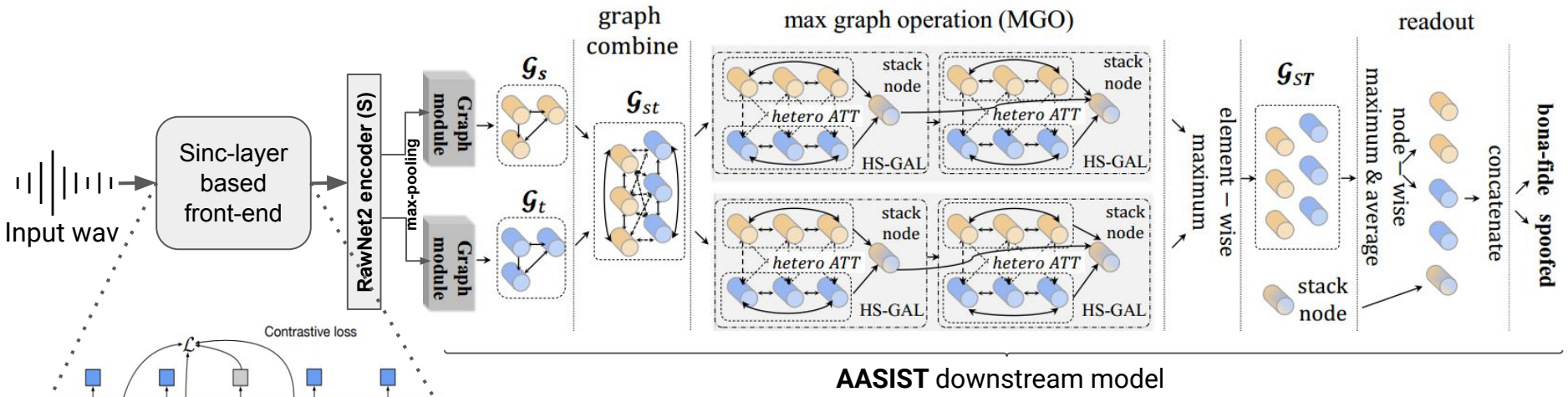
X. Wang et al., “Investigating self-supervised front ends for speech spoofing countermeasures,” in Proc. Speaker Odyssey, 2022.

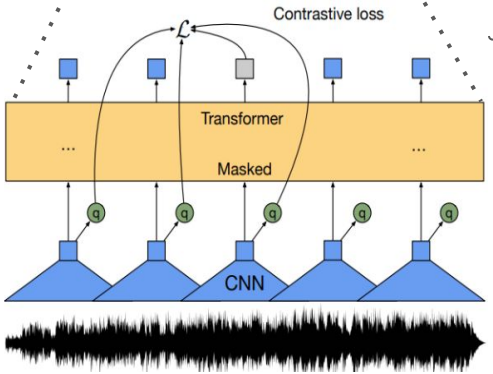
Results



X. Wang et al., “Investigating self-supervised front ends for speech spoofing countermeasures,” in Proc. Speaker Odyssey, 2022.

Self-supervised front-end with graph neural network

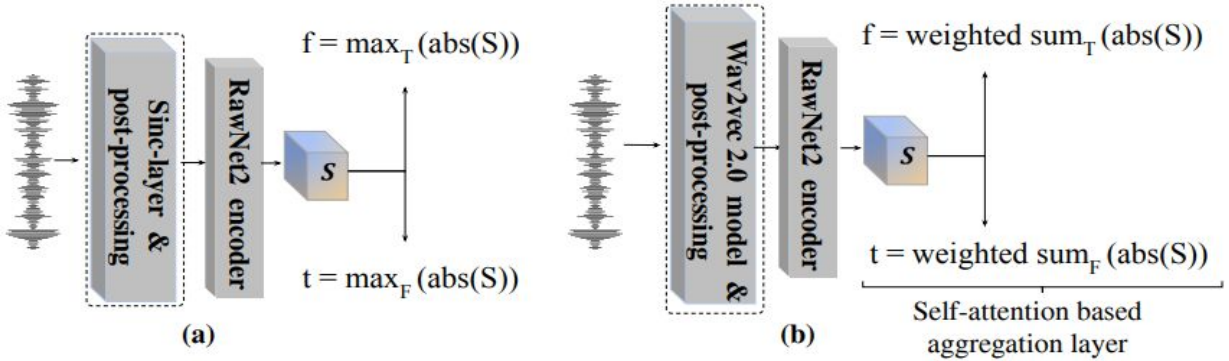




- **wav2vec 2.0 XLSR**[Baevski 2020] is a self-supervised pre-trained model using massive amount of bona fide data
  - multi-lingual, speakers, background noises, codec
- perform **fine-tuning** on ASVspoof 2019 LA bona fide & spoofed data for downstream task

A. Baevski et al., “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” in Proc. NeurIPS 2020.

Front-ends



Conventional DSP based front-end

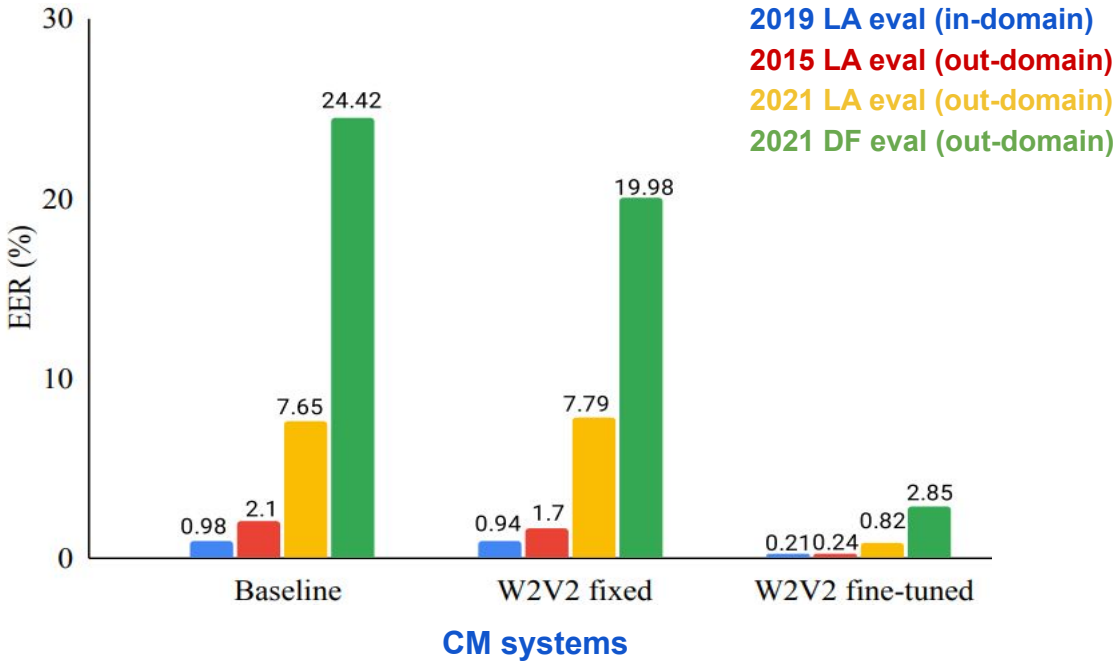
Self-supervised front-end

Self-attentive aggregation layer[Okabe 2018] : to extract more attentive/relevant spectral and temporal features.

K. Okabe et al. “Attentive Statistics Pooling for Deep Speaker Embedding,” in Proc. Interspeech 2018.
X. Wang et al., “Investigating self-supervised front ends for speech spoofing countermeasures,” in Proc. Speaker Odyssey, 2022.
H. Tak et al., “Automatic speaker verification spoofing and deepfake detection using Wav2vec 2.0 and data augmentation,” in Proc. Speaker Odyssey, 2022.

CM results using graph based back-end

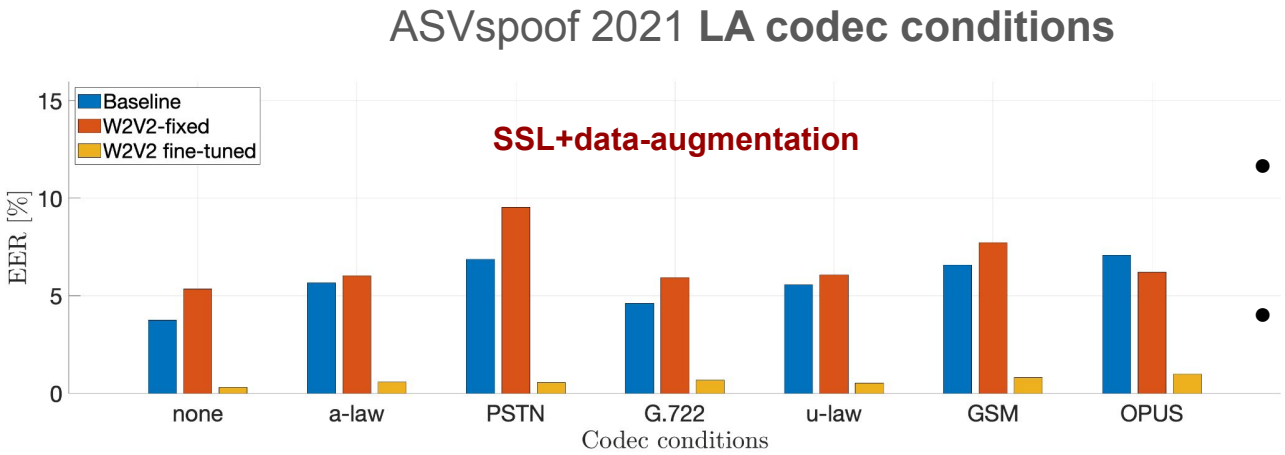
Performance comparison on **multiple test sets** to check CM generalisability



- **SSL** front-end generalies better than baseline (AASIST).
- Fine-tuning the SSL front-end helps.

H. Tak, “End-to-End Modeling for Speech Spoofing and Deepfake Detection,” 2023 (PhD thesis).

Benefit of data-augmentation with SSL front-end



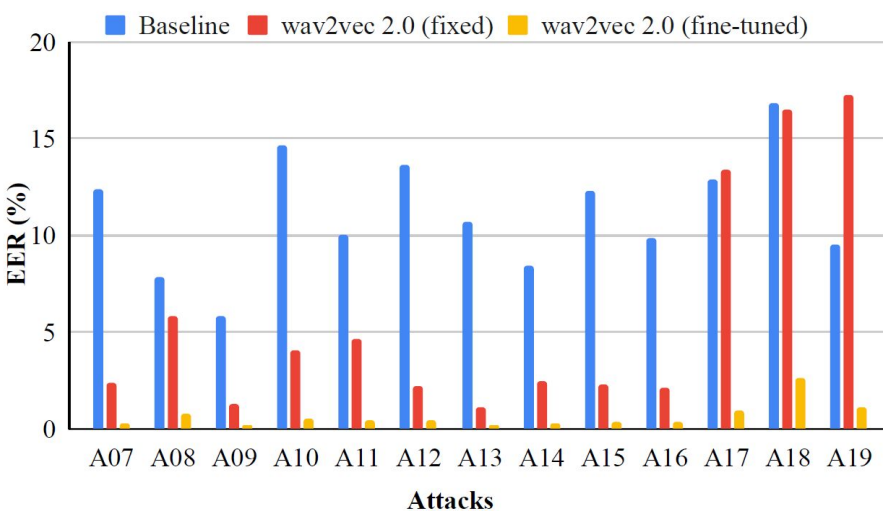
- **SSL** front-end generalises well to different seen/**unseen codec and attack conditions**
- combined use of **SSL+RawBoost**[Tak 2022] is beneficial for reliable spoofing detection

- Similar trends also observed for **ASVspoof 2021 DF** task which contains different codecs (mp3, ogg..)

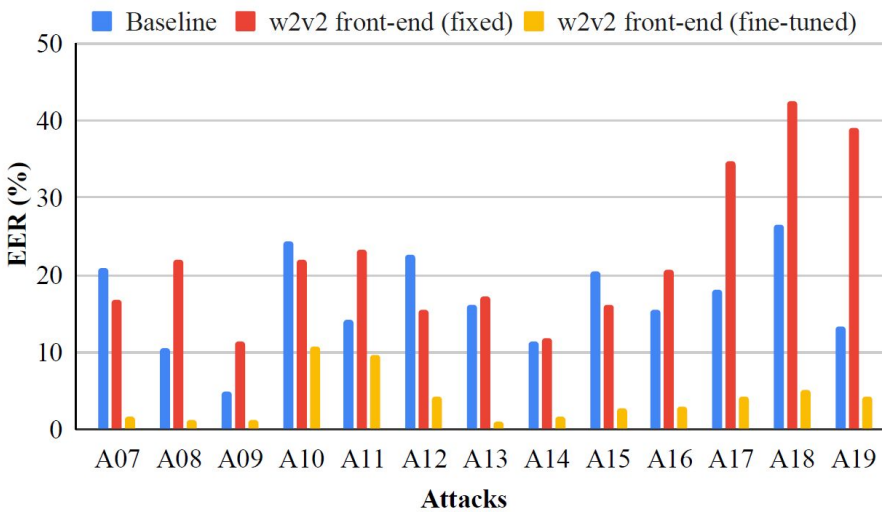
H. Tak et al., “RawBoost: A Raw Data Boosting and Augmentation Method applied to Automatic Speaker Verification Anti-Spoofing,” in Proc. ICASSP, 2022.

Breakdown results: Attack-wise

ASVspoof 2021 LA attack-wise breakdown performance



ASVspoof 2021 DF attack-wise breakdown performance



- **SSL-based CM** achieve superior performance and generalised better than baseline (AASIST).

H. Tak et al., “Automatic speaker verification spoofing and deepfake detection using Wav2vec 2.0 and data augmentation,” in Proc. Speaker Odyssey, 2022.

Findings

- **Fine-tuning SSL front-end** improves CM generalisation
- **More sophisticated back-end**[Tak 2022] can further enhance the CM performance
- **Data-augmentation techniques**[Martin-Donas 2022, Tak 2022] are complementary to SSL front-end for spoofed speech detection

H. Tak et al., “Automatic speaker verification spoofing and deepfake detection using Wav2vec 2.0 and data augmentation,” in Proc. Speaker Odyssey, 2022.
J. M. Martın-Donas et al., “The Vicomtech Audio Deepfake Detection System based on Wav2Vec2 for the 2022 ADD Challenge,” in Proc. ICASSP, 2022.
H. Tak et al., “RawBoost: A Raw Data Boosting and Augmentation Method applied to Automatic Speaker Verification Anti-Spoofing,” accepted in ICASSP, 2022.

Limitations

- Computationally expensive
  - Need for light-weight model for real-world applications
- The best CM model works well on the ASVspoof databases, how about the performance on a unseen test set (in the wild dataset)?

Best CM performance on unseen test set

Q : What is the **Best CM** performance on **in the wild dataset**[Muller 2022] ?

In the wild dataset is more challenging

- more diverse acoustic characteristics
- fake audios are generated using publicly available sources such as social networks and popular video sharing platforms
- 50 English-speaking celebrities and politicians

➔ **10.46% EER** on in the wild dataset

The best CM system is not well generalised to an unseen test set.

N. M Muller et al., “Does Audio ¨ Deepfake Detection Generalize?,” in Proc. Interspeech, 2022.

Conclusion & remarks

- E2E GNNs demonstrate robust performance to in-domain unknown attacks
- SSL models further bring robustness towards out-of-domain evaluations
- **Open questions & issues**
  - Generalization capabilities to in the wild test sets
  - GNNs seems work very well for anti-spoofing, what's next ?
  - Explainability and Interpretability: GNNs provide interpretability by allowing inspection of the learned representations and the influence of different nodes or edges in the graph.

Conclusion & remarks

Code & implementation

- **End-to-end spectro-temporal graph attention network (RawGAT-ST)**
  github.com/eurecom-asp/RawGAT-ST-antispoofing
- **AASIST**
  github.com/clovaai/aasist
- **SSL-based CM solution for anti-spoofing & deepfake detection**
  github.com/TakHemlata/SSL_Anti-spoofing
  github.com/nii-yamagishilab/project-NN-Pytorch-scripts

Thank you for your attention

Databases

- ASVspoof challenge datasets (logical access)
  - **ASVspoof 2015: http://dx.doi.org/10.7488/ds/298**
  - **ASVspoof 2019: https://doi.org/10.7488/ds/2555**
  - **ASVspoof 2021: https://doi.org/10.5281/zenodo.4837263**
- Others
  - **In-the-wild**: https://deepfake-demo.aisec.fraunhofer.de/in_the_wild
  - **WaveFake**: https://zenodo.org/record/5642694
  - **FAD:** https://zenodo.org/record/8122764
  - **AVSpoof**: https://www.idiap.ch/en/dataset/avspoof
  - **LibriSeVoc**: https://github.com/csun22/Synthetic-Voice-Detection-Vocoder-Artifacts

Thank you for your attention

Where to download ?

https://github.com/Jungjee/INTERSPEECH2023_T6

- Tutorial slides
- Hands-on notebook
- Pytorch code and scripts

ASVspoof5 (2023 edition on-going challenge)

ASVspoof
Automatic Speaker Verification and Spoofing Countermeasures Challenge

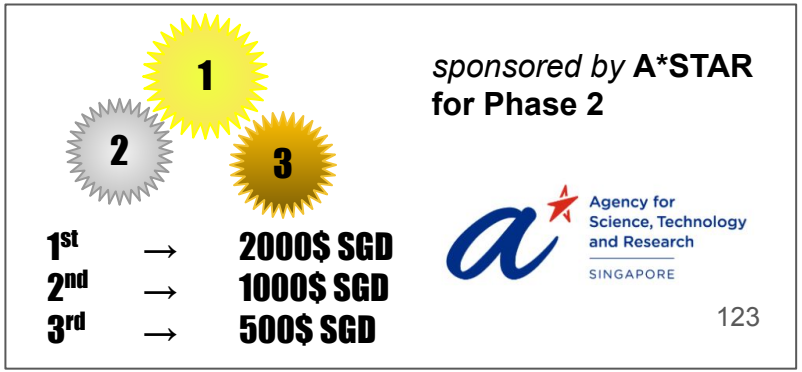
Call For Spoofed/Speech DeepFake Data Contributors

Evaluation plan
www.asvspoof.org/file/ASVspoof5___Evaluation_Plan.pdf

www.asvspoof.org/
info@asvspoof.org

Spoofed data generation (Phase 1): Ongoing
Phase 1 submission: September 22, 2023
Deepfake detection (Phase 2): beginning last quarter, 2023

- Phase 1: building TTS/VC and creating spoofed data
  - feedbacks from surrogate ASV/CM models are available
  - source databases: MLS (English subset), CommonVoice (English subset)
- Phase 2: spoofing detection
  - as in ASVspoof 2021, spoofing detection of degraded-quality data
  - both CM-only and CM+ASV tasks



References

[N. Evans, et al., 2013] Spoofing and countermeasures for automatic speaker verification. in Proc. Interspeech, 2013.
[T. Kinnunen, et al., 2018] t-DCF: A Detection Cost Function for the Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification, in Proc. Speaker Odyssey, 2018.
[X. Wang, et al., 2020] ASVspoof 2019: a large-scale public database of synthetized, converted and replayed speech, Computer Speech & Language, vol. 64, 2020.
[P. Veličković, et al., 2018] Graph Attention Networks, in Proc. International Conference on Learning Representations (ICLR), 2018.
[Z. Wu, et al., 2020] A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 2020.
[H. Yuan, et al., 2022] Explainability in graph neural networks: A taxonomic survey. IEEE transactions on pattern analysis and machine intelligence (T-PAMI), 2022.
[H. Tak, et al., 2021] Graph Attention Network for Anti-spoofing, in Proc. Interspeech 2021.
[J. Jung, et al., 2022] AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks, in Proc. ICASSP, 2022.
[X. Wang, et al., 2022] Investigating self-supervised front ends for speech spoofing countermeasures, in Proc. Speaker Odyssey, 2022.
[H. Tak, et al., 2022] Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation, in Proc. Speaker Odyssey, 2022.

References

[B. Pellom, et al., 1999] An Experimental Study of Speaker Verification Sensitivity to Computer Voice-Altered Imposters. In Proc. ICASSP, 2:837–840. 1999.
[T. Masuko, et al., 2000] Imposture Using Synthetic Speech against Speaker Verification Based on Spectrum and Pitch. In Proc. ICSLP. 2000.
[P. De Leon, et al., 2012] Evaluation of Speaker Verification Security and Detection of HMM-Based Synthetic Speech. IEEE Transactions on Audio, Speech, and Language Processing 20 (8). IEEE: 2280–2290.
[L.W. Chen, et al., 2010] Speaker Verification against Synthetic Speech. In Proc. ISCSLP, 309–312. 2010.
[A. Martin, et al., 1997] The DET Curve in Assessment of Detection Task Performance. In Proc. Eurospeech, 1895–1898. 1997.
[T. Kinunnen, et al., 2020] Tandem Assessment of Spoofing Countermeasures and Automatic Speaker Verification: Fundamentals. IEEE/ACM Transactions on Audio, Speech, and Language Processing 28. IEEE: 2195–2210. 2020.
[M. Sahidullah, et al., 2015] A Comparison of Features for Synthetic Speech Detection. In Proc. Interspeech, 2087–2091. 2015.
[N. Zeghidour, et al., 2021] Leaf: A Learnable Frontend for Audio Classification. Proc. ICLR. 2021.
[M. Kamble, et al., 2020] Advances in Anti-Spoofing: From the Perspective of ASVspoof Challenges. APSIPA Transactions on Signal and Information Processing. doi:10.1017/ATSIP.2019.21. 2020.
[X. Wang, et al., 2022] A Practical Guide to Logical Access Voice Presentation Attack Detection. Springer. 2022.
[D. Van Leeuwen and N. Brümmer] An Introduction to Application-Independent Evaluation of Speaker Recognition Systems. In Speaker Classification I, 330–353. Springer. 2007.

References

On text-to-speech synthesis

[A. Hunt and A. Black, 1996] Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database. In Proc. ICASSP, 373–376. 1996.

[K. Yoshimura, et al., 1999] Simultaneous Modeling of Spectrum, Pitch and Duration in {HMM}-Based Speech Synthesis. In Proc. Eurospeech, 2347–2350. 1999.

[K. Tokuda, et al., 2000] Speech Parameter Generation Algorithms for HMM-Based Speech Synthesis. In Proc. ICASSP, 936–939. 2000.

[H. Zen, et al., 2013] Statistical Parametric Speech Synthesis Using Deep Neural Networks. In Proc. ICASSP, 7962–7966. 2013.

[Z. H. Ling, et al., 2013] Modeling Spectral Envelopes Using Restricted Boltzmann Machines and Deep Belief Networks for Statistical Parametric Speech Synthesis. IEEE Trans. ASLP 21 (10). IEEE: 2129–2139. 2013.

[Y. Wang, et al., 2017] Tacotron: Towards End-to-End Speech Synthesis. In Proc. Interspeech, 4006–4010. doi:10.21437/Interspeech.2017-1452. 2017.

[A. Oord, et al., 2016] WaveNet: A Generative Model for Raw Audio. CoRR abs/1609.0. 2016.

[N. Li, 2019] Neural Speech Synthesis with Transformer Network. In Proceedings of the AAAI Conference on Artificial Intelligence, 33:6706–6713. 2019.

[J. Shen, 2018] Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. In Proc. ICASSP, 4779–4783. 2018.

References

On text-to-speech synthesis

- On classical speech synthesis methods (e.g., formant synthesis, concatenative & signal processing)
  - Dutoit, T. *An Introduction to Text-to-speech Synthesis*. (Kluwer Academic Publishers, 1997).
  - Taylor, P. *Text-to-Speech Synthesis*. (Cambridge University Press, 2009).
  - **Chapter 16**, Huang, X., Acero, A., Hon, H.-W. & Reddy, R. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. (Prentice Hall PTR, 2001).
  - **Chapter 8**, Jurafsky, D. & Martin, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. (Prentice Hall PTR, 2000).
- Tutorial & talk
  - Wang, X. Tutorial on Neural statistical parametric speech synthesis, ISCA Odyssey 2020, http://tonywangx.github.io/slide.html#dec-2020 (cover HMM & many seq2seq attention models)
  - King, S., Watts, O., Ronanki, S., Espic, F., Wu, Z. Deep Learning for Text-to-Speech Synthesis, using the Merlin toolkit. Tutorial at Interspeech 2017 (Many details explained) http://speech.zone/courses/one-off/merlin-interspeech2017/
  - Qian, Y, Soong, F. Deep learning for speech generation and synthesis. Tutorial at ISCSLP 2014 https://www.superlectures.com/iscslp2014/tutorial-4-deep-learning-for-speech-generation-and-synthesis
  - Dr. Heigz Zen's talk on DNN or HMM-based statistical parametric speech synthesis http://rtthss2015.talp.cat/ & http://www.sp.nitech.ac.jp/~zen/english/index.php?Publications%2FTalks

References

On text-to-speech synthesis

- Tutorial & talk (continue)
  - Zen, H. *Generative Model-Based Text-to-Speech Synthesis*, CBMM Workshop https://cbmm.mit.edu/video/generative-model-based-text-speech-synthesis
  - Alex Graves: Attention & Memory, **EEML 2020**, https://www.youtube.com/watch?v=POtAGvwnB5s
  - Smola, A. Attention in deep learning, ICML 2019, http://alex.smola.org/talks/ICML19-attention.pdf
  - Alex Graves: *Hallucination with RNNs https://www.youtube.com/watch?v=-yX1SYeDHbg*
- Overview papers on statistical parametric speech synthesis
  - HMM & decision tree
    - Tokuda, K. et al. Speech synthesis based on hidden Markov models. Proc. IEEE 101, 1234–1252 (2013)
    - Zen, H., Tokuda, K. & Black, A. W. Statistical parametric speech synthesis. Speech Commun. 51, 1039–1064 (2009)
  - HMM + DNN
    - Ling, Z. H. et al. Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. IEEE Signal Process. Mag. 32, 35–52 (2015)

References

On text-to-speech synthesis

- Papers on TTS front-end topics
  - Prosody: I found this book and paper useful as introduction.
    - Gussenhoven, C. The phonology of tone and intonation. (Cambridge University Press, 2004).
  - Prosody labelling (e.g., ToBI):
    - Beckman, M. E. & Ayers, G. Guidelines for ToBI labelling. OSU Res. Found. 3, (1997)
    - Silverman, K. E. A. et al. ToBI: a standard for labeling English prosody. in Proc. ICSLP 867–870 (1992).
  - Prediction prosodic labels from text:
    - Hirschberg, J. Pitch accent in context predicting intonational prominence from text. Artif. Intell. 63, 305–340 (1993)
  - Unsupervised front-end:
    - Watts, O. S. Unsupervised learning for text-to-speech synthesis. (University of Edinburgh, 2013)

- Neural waveform models
  - Wang, X.: neural waveform models course http://tonywangx.github.io/slide.html#jul-2021 (with Jupyter notebook!)