The SJTU X-LANCE System for Speaker Recognition Challenge VoxSRC2022 and CNSRC2022

X-LANCE Lab, Shanghai Jiao Tong University

Presenter: Zhengyang Chen

Supervisor: Yanmin Qian

2022.10.22

Outline

- Introduction of VoxSRC 2022 and CNSRC 2022
- System Description
- Results and Analysis

Introduction of VoxSRC 2022 and CNSRC 2022

Comparison between VoxSRC and CNSRC

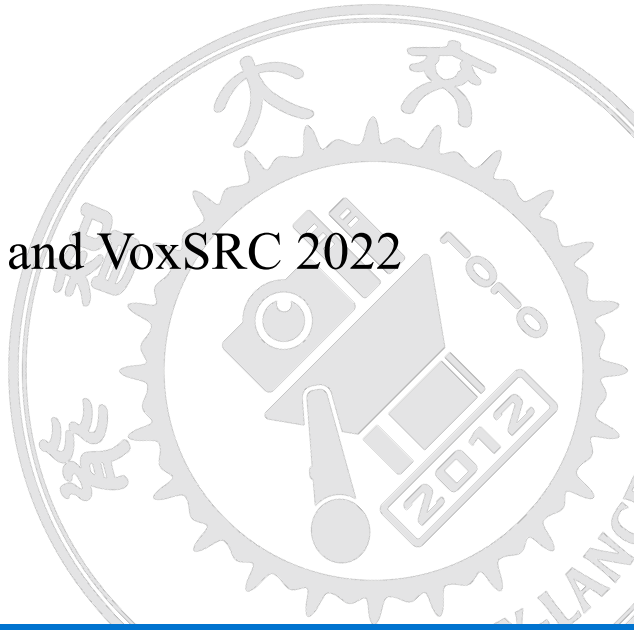
- VoxSRC (Voxceleb Speaker Recognition Challenge)
  - First held in 2019
  - Mainly based on Voxceleb
- CNSRC (CN-Celeb Speaker Recognition Challenge)
  - First held in 2022
  - Mainly based on CN-Celeb

Introduction of VoxSRC 2022 and CNSRC 2022

Comparison between VoxSRC and CNSRC

- VoxSRC (Voxceleb Speaker Recognition Challenge)
  - First held in 2019
  - Mainly based on Voxceleb
- CNSRC (CN-Celeb Speaker Recognition Challenge)
  - First held in 2022
  - Mainly based on CN-Celeb

We achieved 1st place and 3rd place for CNSRC 2022 and VoxSRC 2022 fixed speaker verification track respectively.



Introduction of VoxSRC 2022 and CNSRC 2022

Comparison between VoxSRC 2022 and CNSRC 2022 Speaker Verification Fixed

Challenge	Training Data				
	Data Source	Speaker #	Data Amount	Language	Genre Number
VoxSRC 2022	VoxCeleb2 dev	5994	2360 hrs	Mainly English	1 (interview)
CNSRC 2022	CN-Celeb	2687	1312 hrs	Chinese	11

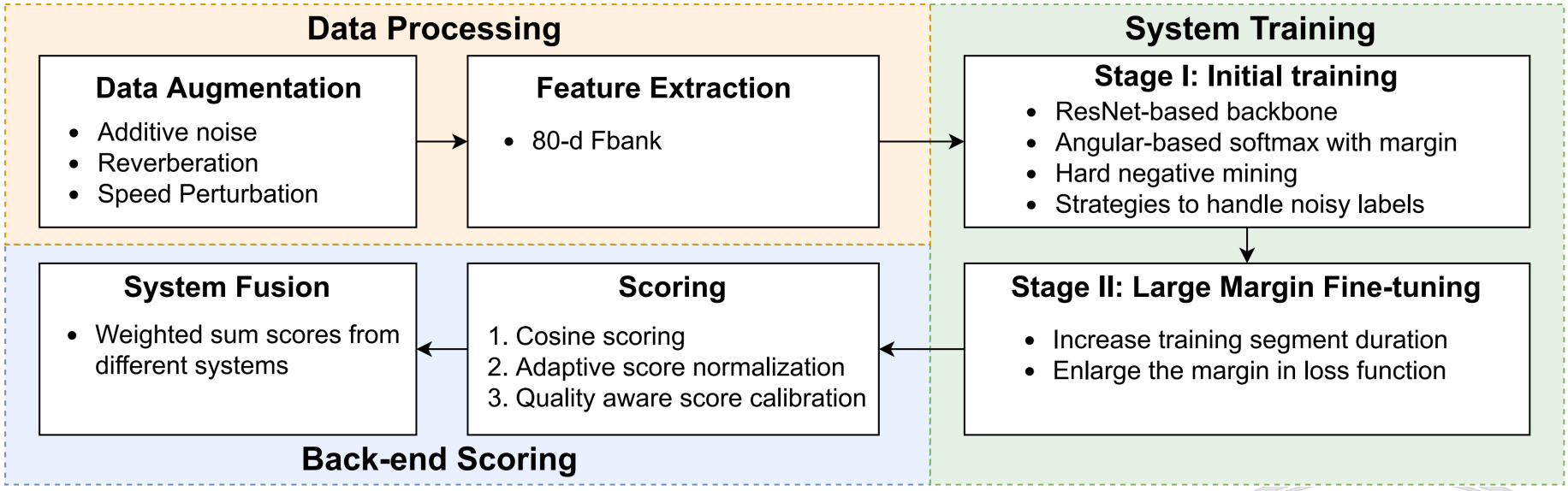
Comparison between training data.

Challenge	Evaluation Data & Trial		
	Average Duration	Focus	Multi Enrollment
VoxSRC 2022	7.4 s	Cross-Age/Channel	No
CNSRC 2022	7.7 s	Cross-Genre	Yes

Comparison between evaluation setup.

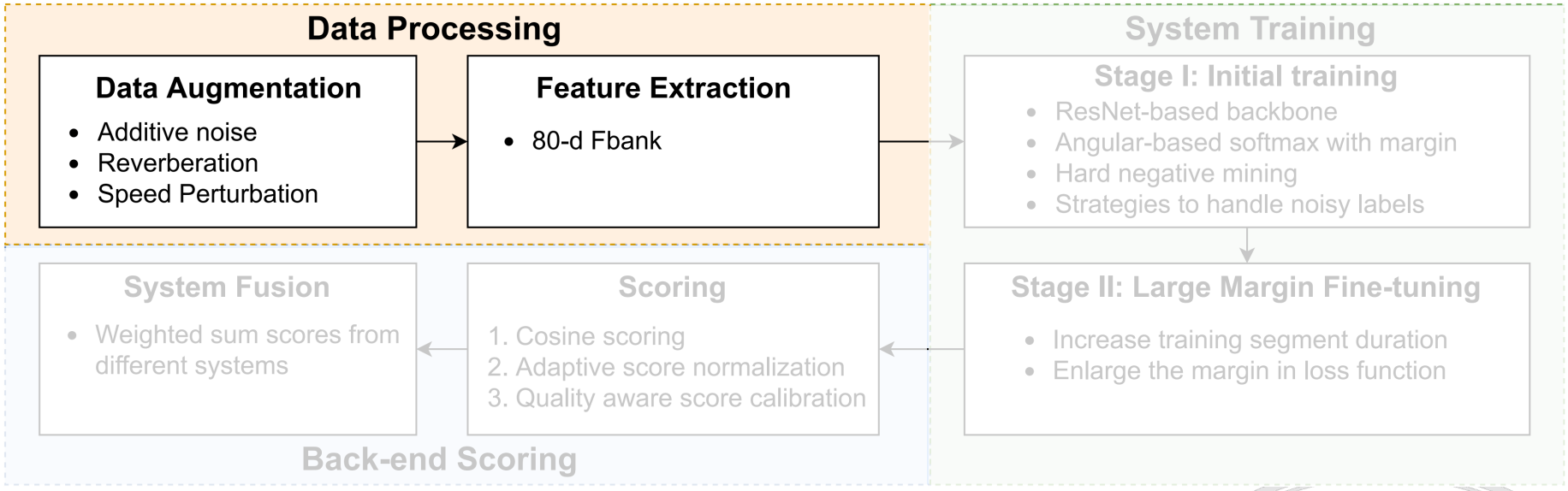
System Description

System Outline



System Description

Data Processing



Data Augmentation

- Additive noise & Reverberation
- Speed Perturbation [1]
  - Speed up the audio with ratio 1.1
  - Slow down the audio with ratio 0.9
  - The augmented audio is considered from a new speaker (pitch is changed).

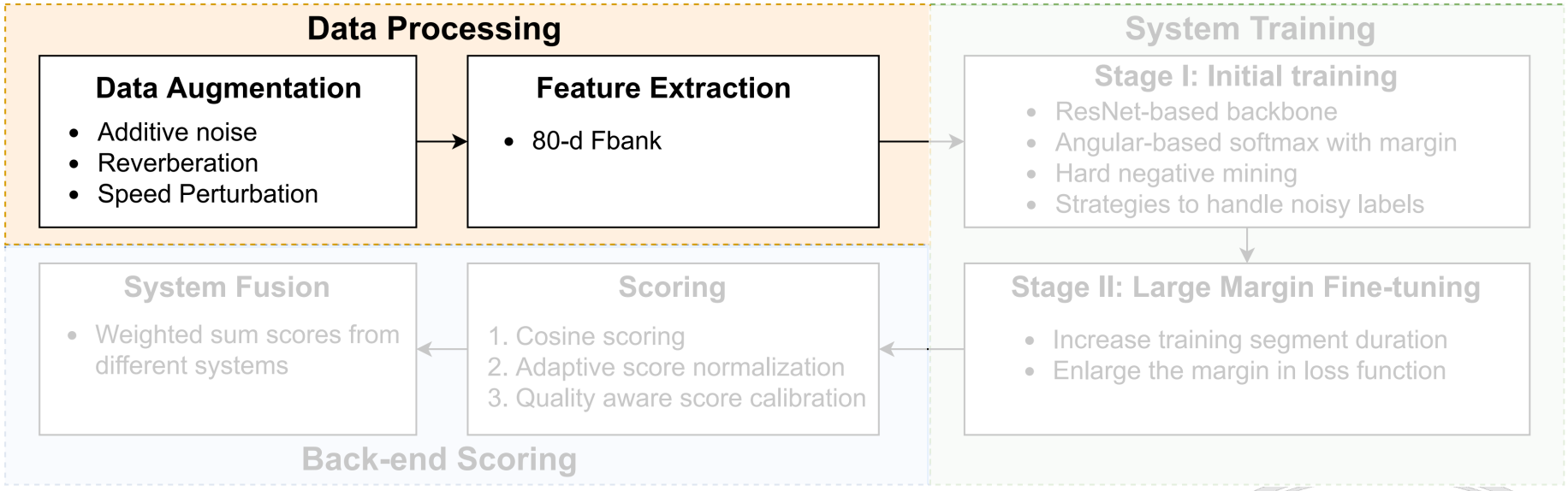
Feature Extraction

- 80-dimensional Fbank feature

[1] Wang, Weiqing, et al. "The dku-dukeece systems for voxceleb speaker recognition challenge 2020." arXiv preprint arXiv:2010.12631 (2020).

System Description

Data Processing



Data Augmentation

- Additive noise & Reverberation
- Speed Perturbation [1]
  - Speed up the audio with ratio 1.1
  - Slow down the audio with ratio 0.9 `sox audio1.wav audio2.wav speed 0.9`
  - The augmented audio is considered from a new speaker (pitch is changed).

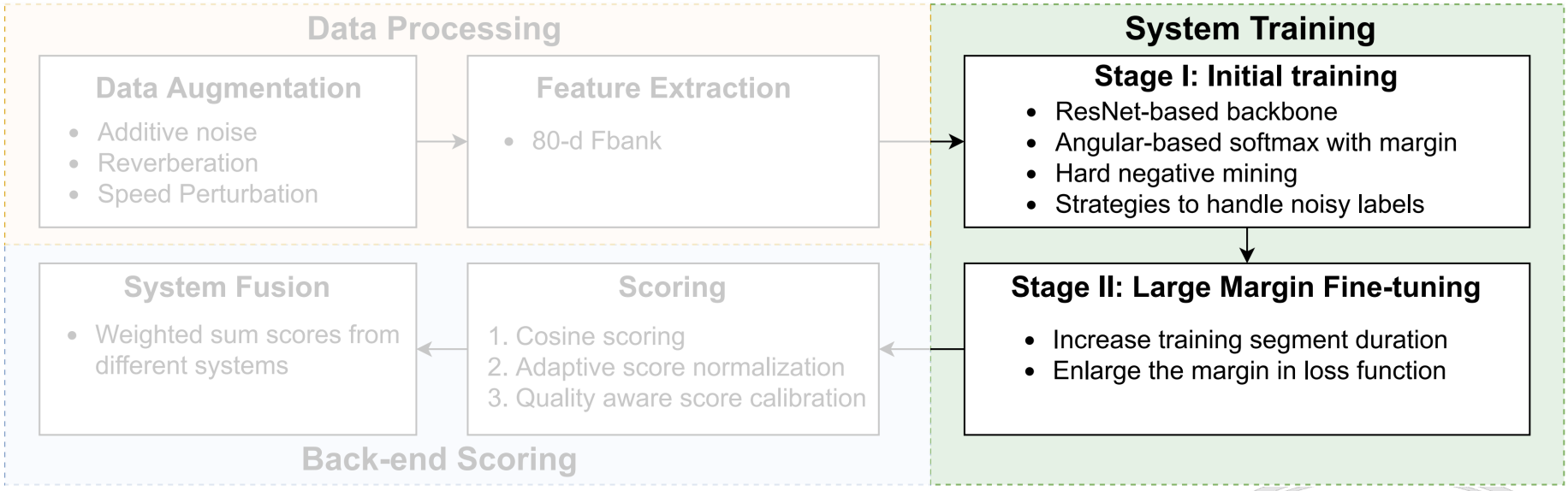
Feature Extraction

- 80-dimensional Fbank feature

[1] Wang, Weiqing, et al. "The dku-dukeece systems for voxceleb speaker recognition challenge 2020." arXiv preprint arXiv:2010.12631 (2020).

System Description

System Training



Stage I: Initial training

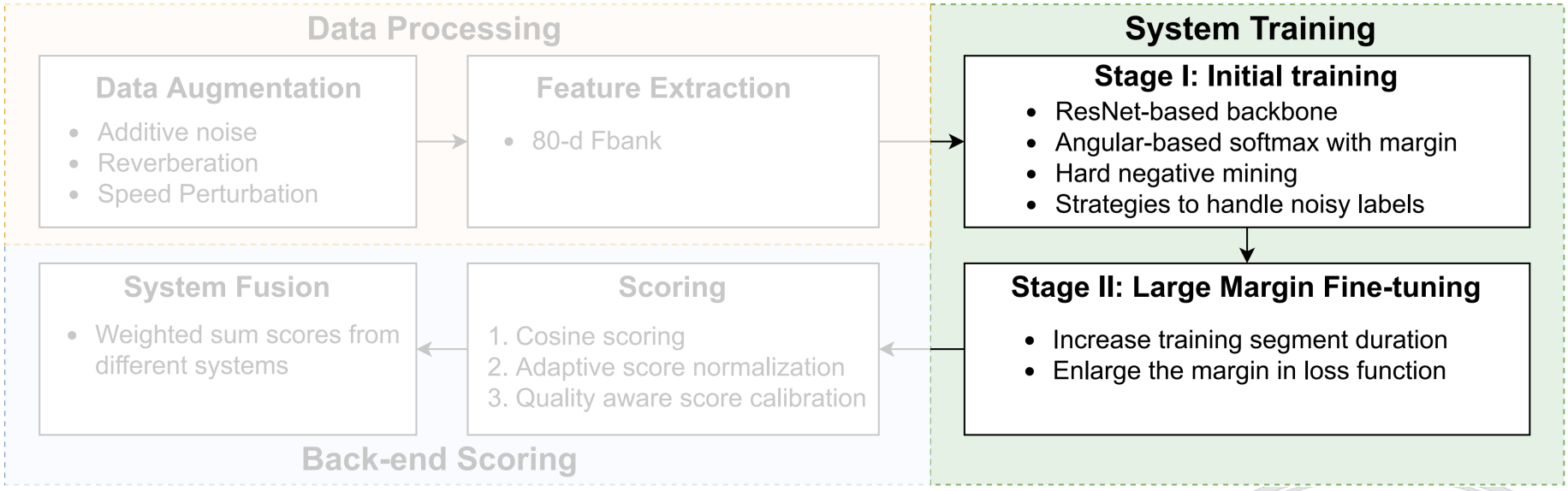
- ResNet-based backbone
  - Design some deeper resnet [2]

Layer name	Structure	Output
Input	–	80 × Frame Num × 1
Conv2D-1	3 × 3, Stride 1	80 × Frame Num × 32
ResNetBlock-1	$\begin{bmatrix} 1\times1, 32 \\ 3\times3, 32 \\ 1\times1, 128 \end{bmatrix} \times N_1$, Stride 1	80 × Frame Num × 128
ResNetBlock-2	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times N_2$, Stride 2	40 × Frame Num//2 × 256
ResNetBlock-3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times N_3$, Stride 2	20 × Frame Num//4 × 512
ResNetBlock-4	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times N_4$, Stride 2	10 × Frame Num//8 × 1024
StatisticPooling	–	20 × 1024
Flatten	–	20480
Emb Layer	–	256

[2] Chen, Zhengyang, et al. "The SJTU X-LANCE Lab System for CNSRC 2022." arXiv preprint arXiv:2206.11699 (2022).

System Description

System Training



Stage I: Initial training

- ResNet-based backbone
  - Design some deeper resnet [2]

Table 3: **Configuration for different deep ResNet.**

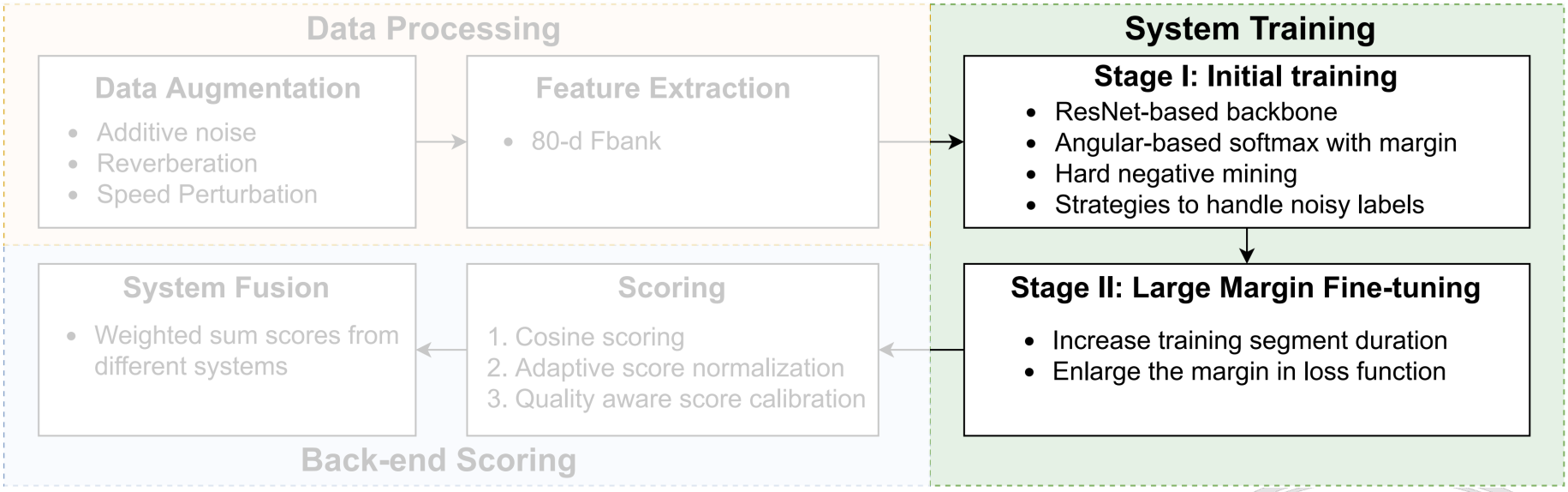
Deep ResNet Name	(N_1, N_2, N_3, N_4)
ResNet152	$(3, 8, 36, 3)$
ResNet221	$(6, 16, 48, 3)$
ResNet293	$(10, 20, 64, 3)$

Layer name	Structure	Output
Input	–	80 × Frame Num × 1
Conv2D-1	3 × 3, Stride 1	80 × Frame Num × 32
ResNetBlock-1	$\begin{bmatrix} 1\times1, 32 \\ 3\times3, 32 \\ 1\times1, 128 \end{bmatrix} \times N_1$, Stride 1	80 × Frame Num × 128
ResNetBlock-2	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times N_2$, Stride 2	40 × Frame Num//2 × 256
ResNetBlock-3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times N_3$, Stride 2	20 × Frame Num//4 × 512
ResNetBlock-4	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times N_4$, Stride 2	10 × Frame Num//8 × 1024
StatisticPooling	–	20 × 1024
Flatten	–	20480
Emb Layer	–	256

[2] Chen, Zhengyang, et al. "The SJTU X-LANCE Lab System for CNSRC 2022." arXiv preprint arXiv:2206.11699 (2022).

System Description

System Training



Stage I: Initial training

- ResNet-based backbone
  - Design some deeper resnet

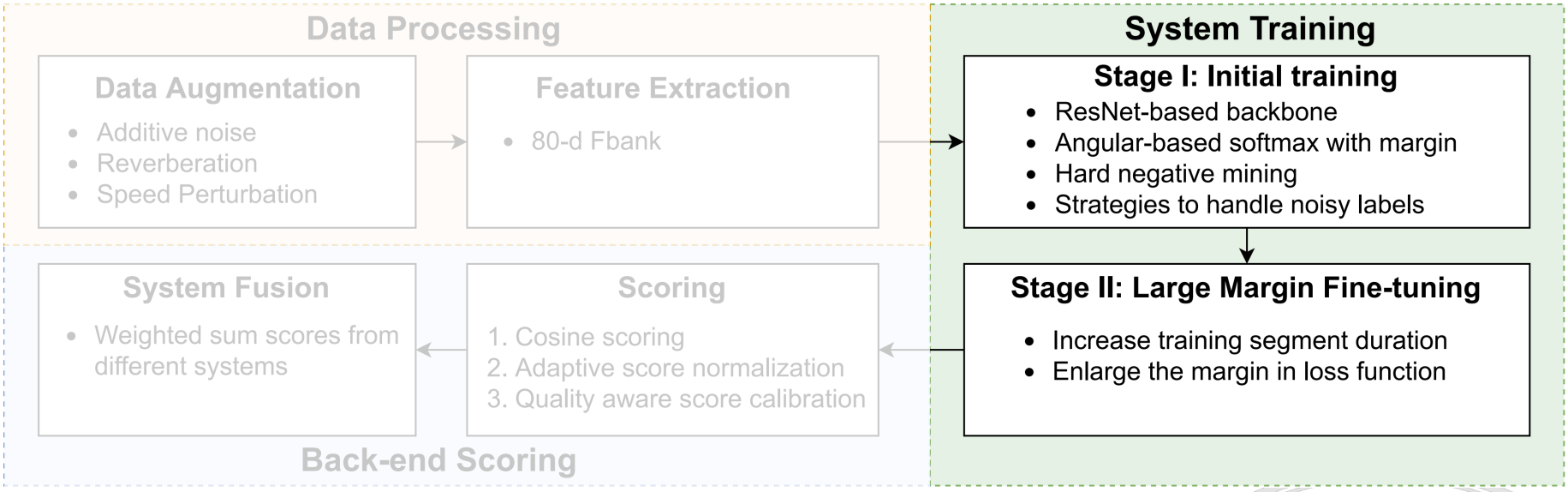
Can be replaced by more advanced attention based pooling, like multi-query and multi-head attention pooling (MQHHA) [2].

Layer name	Structure	Output
Input	–	80 × Frame Num × 1
Conv2D-1	3 × 3, Stride 1	80 × Frame Num × 32
ResNetBlock-1	$\begin{bmatrix} 1\times1, 32 \\ 3\times3, 32 \\ 1\times1, 128 \end{bmatrix} \times N_1$, Stride 1	80 × Frame Num × 128
ResNetBlock-2	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times N_2$, Stride 2	40 × Frame Num//2 × 256
ResNetBlock-3	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times N_3$, Stride 2	20 × Frame Num//4 × 512
ResNetBlock-4	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times N_4$, Stride 2	10 × Frame Num//8 × 1024
StatisticPooling	–	20 × 1024
Flatten	–	20480
Emb Layer	–	256

[3] Zhao, Miao, et al. "Multi-Query Multi-Head Attention Pooling and Inter-Topk Penalty for Speaker Verification." ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022.

System Description

System Training



Stage I: Initial training

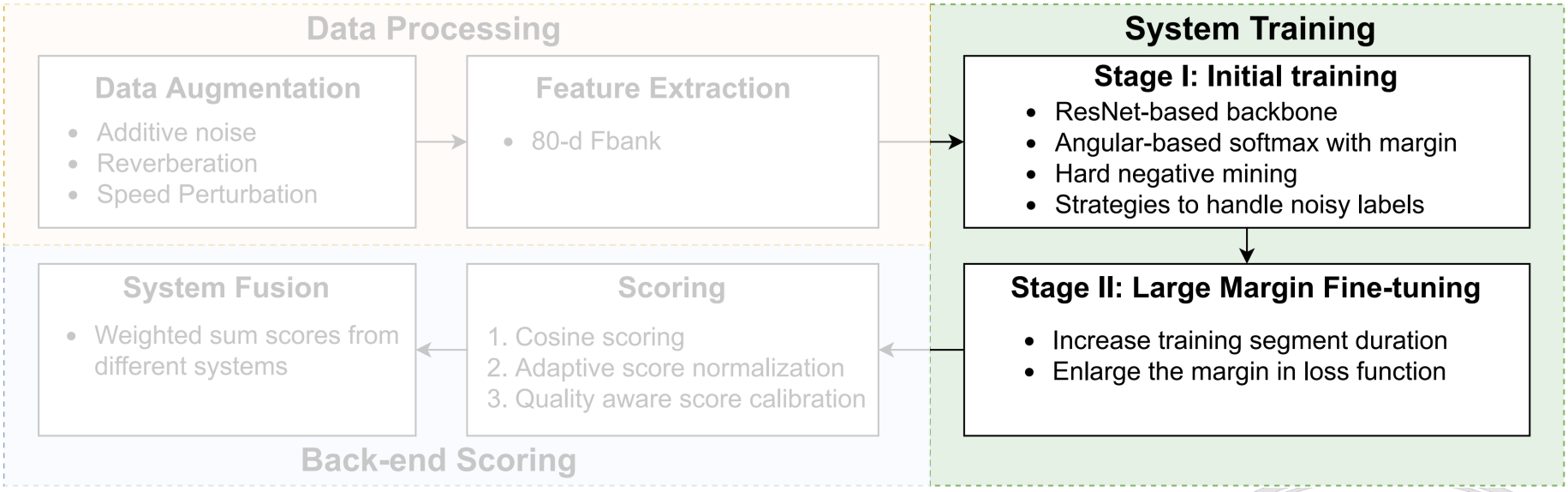
- ResNet-based backbone
- Angular-based softmax with margin
  - m1: multiplicative margin
  - m2 (0.2): additive angular margin (AAM)
  - m3 (0.2): additive margin (AM)

$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp^{s\cdot f(\theta_{y_i})}}{\exp^{s\cdot f(\theta_{y_i})}+\sum_{j=1,j\neq y_i}^{C}\exp^{s\cdot\cos(\theta_j)}}$$

$$f(\theta_{y_i})=\cos(m_1\theta_{y_i}+m_2)-m_3$$

System Description

System Training



Stage I: Initial training

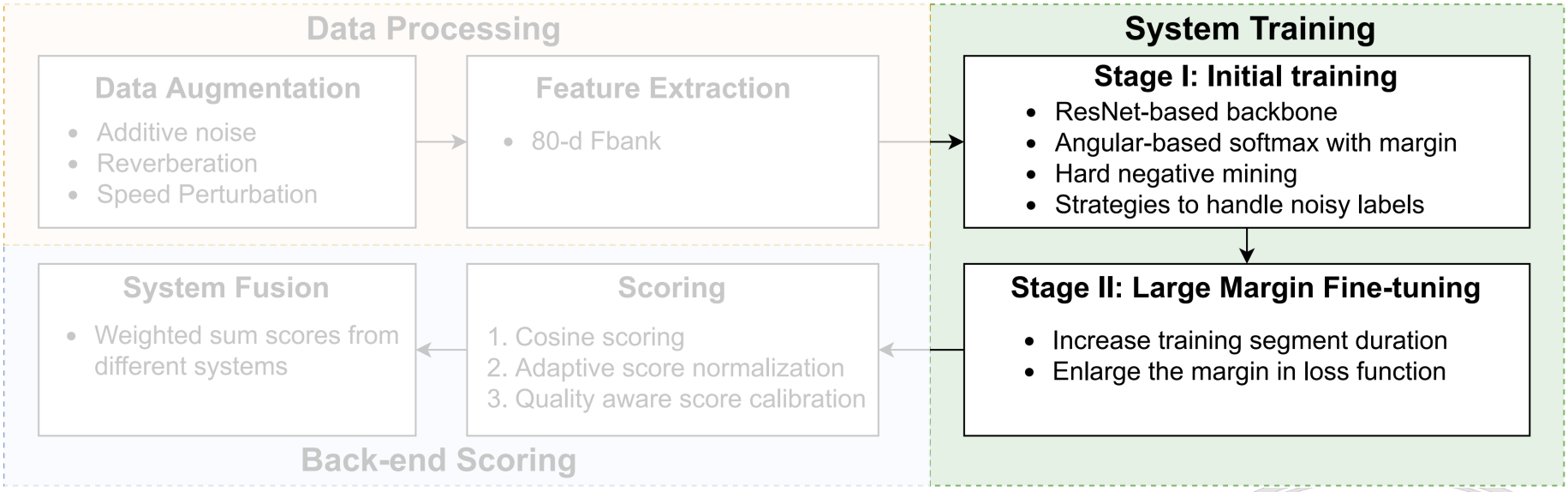
- ResNet-based backbone
- Angular-based softmax with margin
- Hard negative mining
  - Inter-Topk [3]
  - HEM (Hard example mining)

$$-\log \frac{e^{s\cos\left(\theta_{i,y_i}+m\right)}}{e^{s\cos\left(\theta_{i,y_i}+m\right)}+\sum_{j=1,j\neq y_i}^{N} e^{s\phi(\theta_{i,j})}}$$

[3] Zhao, Miao, et al. "Multi-Query Multi-Head Attention Pooling and Inter-Topk Penalty for Speaker Verification." ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022.

System Description

System Training



Stage I: Initial training

- ResNet-based backbone
- Angular-based softmax with margin
- Hard negative mining
  - Inter-Topk [3]
  - HEM (Hard example mining)

$$-\log \frac{e^{s\cos\left(\theta_{i,y_i}+m\right)}}{e^{s\cos\left(\theta_{i,y_i}+m\right)}+\sum_{j=1,j\neq y_i}^{N} e^{s\phi(\theta_{i,j})}}$$

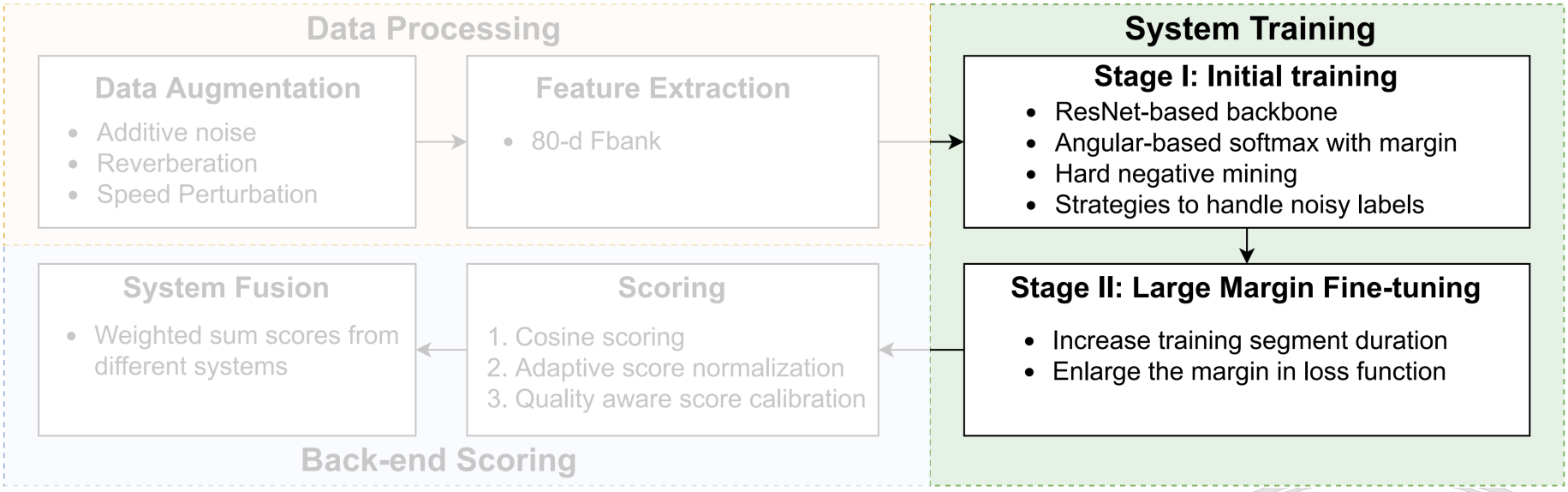
$$\phi(\theta_{i,j}) = \begin{cases} cos\theta_{i,j} + m' & j \in \underset{1\leq n\leq C, n\neq y_i}{arg\,topK}\ (cos\theta_{i,n}) \\ cos\theta_{i,j} & Others. \end{cases} \quad \texttt{Inter-Topk}$$

$$\phi\left(\theta_{i,j}\right) = \begin{cases} \cos(\theta_{i,j} - m') & \text{If } \cos\theta_{i,j} > \cos\left(\theta_{i,y_i} + m\right) \\ \cos\theta_{i,j} & \text{Otherwise.} \end{cases} \quad \texttt{HEM}$$

[3] Zhao, Miao, et al. "Multi-Query Multi-Head Attention Pooling and Inter-Topk Penalty for Speaker Verification." ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022.

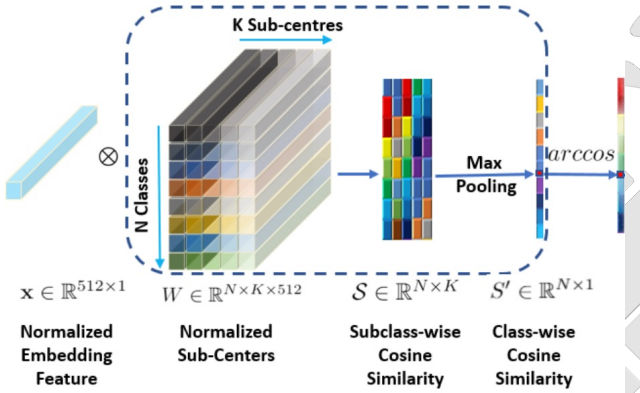
System Description

System Training



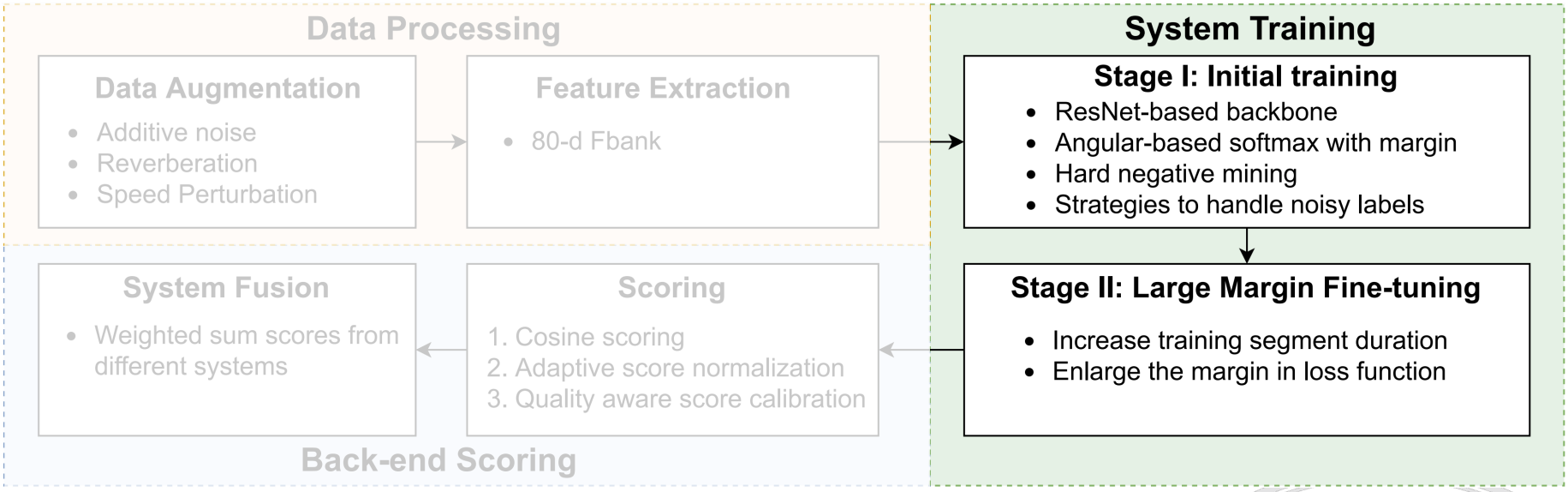
Stage I: Initial training

- ResNet-based backbone
- Angular-based softmax with margin
- Hard negative mining
- Handle noisy labels
  - Sub-center



System Description

System Training



Stage II: Large Margin Fine-tuning [4]

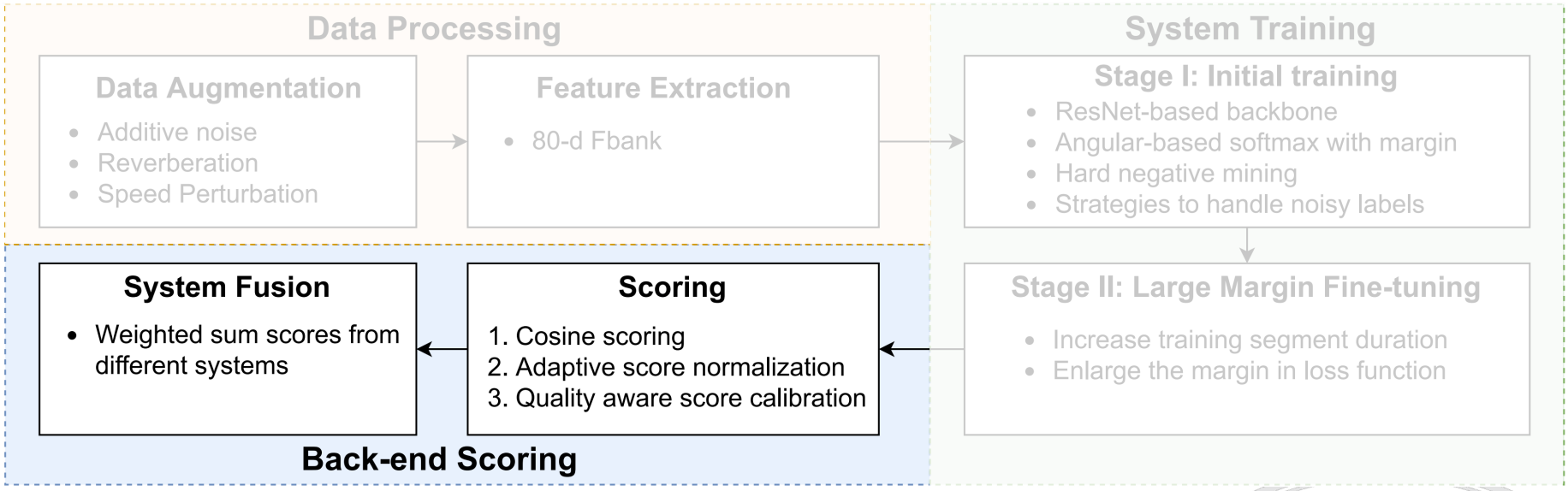
- Enlarge the margin in loss function
  - 0.2 to 0.5
- Increase training segment duration
  - 2s to 6s
- Abandon the speed perturb augmentation

$$-\log \frac{e^{s\cos(\theta_{i,y_i}+m)}}{e^{s\cos(\theta_{i,y_i}+m)}+\sum_{j=1,j\neq y_i}^{N} e^{s\phi(\theta_{i,j})}}$$

[4] Thienpondt, Jenthe, Brecht Desplanques, and Kris Demuynck. "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification." ICASSP 2021.

System Description

Back-end Scoring



Scoring

1. Cosine scoring
2. Adaptive score normalization
   - Estimate imposter cohort from training set
3. Quality aware score calibration [4]
   - Quality measure function: embedding magnitude, utterance duration, mean value of imposter cohort

$$l(s) = w_s s + \mathbf{w}_q^T \mathbf{q} + b$$

[4] Thienpondt, Jenthe, Brecht Desplanques, and Kris Demuynck. "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification." ICASSP 2021.

Comparison between Augmentation Methods

Table 1. Voxceleb EER (%) results comparison between different augmentation methods. N+R: additive noise and reverberation. Perturb (S): audio perturbation by changing audio speed. Perturb (P): audio perturbation by changing audio pitch. Perturb (S+P): audio perturbation by changing audio speed and pitch.

Aug Type	Vox1-O	Vox1-E	Vox1-H
N + R	0.947	1.098	2.002
N + R + SpecAugment	1.064	1.091	2.004
N + R + Perturb (S)	0.973	1.113	2.033
N + R + Perturb (P)	0.867	**0.984**	1.781
N + R + Perturb (S + P)	0.861	0.996	**1.767**
N + R + Perturb (S + P) *	**0.803**	1.001	1.777

*: applying speed perturbation with ratio {0.8, 0.9, 1.0, 1.1, 1.2}

- Combining SpecAugment with additive noise and reverberation cannot obtain further benefits.
- The improvement brought by speed perturbation mainly comes from changing pitch.

Comparison between Different Features

Table 2. Voxceleb EER (%) results comparison between different acoustic features.

Acoustic Feature	Vox1-O	Vox1-E	Vox1-H
MFCC-80d	1.292	1.351	2.436
Fbank-40d	0.941	1.147	2.104
Fbank-80d	**0.861**	0.996	1.767
Fbank-96d	0.931	**0.953**	**1.741**
Fbank-80d + Pitch	0.862	0.984	1.783

- Fbank feature perform much better than the MFCC feature.
- The higher the feature dimension, the better the performance.
- Additional pitch information cannot bring further improvement.

Comparison between Different Soring Methods

Table 3. Voxceleb EER (%) results comparison between different scoring methods. PLDA is trained on Voxceleb2 dev set.

Scoring Method	Vox1-O	Vox1-E	Vox1-H
PLDA	1.633	1.723	2.857
Cosine	1.058	1.147	2.087
+ AS-Norm	0.920	1.048	1.874
++ Score Calibration	**0.861**	**0.996**	**1.767**

- The angular based softmax can optimize the embedding in a hyper-space and cosine scoring is more suitable than PLDA.
- AS-Norm and score calibration can make further improvement.

Different Loss Functions and Training Strategies

Table 4. Voxceleb and VoxSRC EER (%) results comparison between different loss functions. The LM-FT is performed on the model with Inter-Topk loss.

Loss Functions	Vox1-O	Vox1-E	Vox1-H	VoxSRC21-val
AM	0.840	0.987	1.796	3.453
AAM	0.861	0.996	1.767	3.450
+ Sub-center	0.824	0.985	1.733	3.340
++ HEM	**0.782**	0.970	1.684	**3.060**
++ Inter-TopK	**0.782**	**0.936**	**1.658**	3.153
LM-FT	0.649	0.797	1.394	2.429

- Noisy label detection and hard negative penalty can further improve the performance.
- Large margin fine-tuning (LM-FT) can improve the system significantly.

Comparisons between Different Backbones

Table 5. Voxceleb and VoxSRC results comparison between different embedding extractor backbones. '-64' denotes the ResNet with base channel number 64 and our default setup is 32. Large margin fine-tuning is applied for all the systems. The ResNet models with statistic pooling are trained with AAM loss and the ResNet models with MQMHA pooling are trained with AAM + Sub-center + Inter-Topk loss.

Index	Model Type	Pooling	Param #	Vox1-O		Vox1-E		Vox1-H		VoxSRC21-val		VoxSRC22-val	
				$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.05}$	EER	$DCF_{0.05}$	EER
S1	ECAPA (c=512)	ASP	6.20M	0.0901	0.771	0.1051	0.962	0.1670	1.784	0.1911	3.540	0.1676	2.499
S2	ECAPA (c=1024)	ASP	14.7M	0.0802	0.606	0.0889	0.813	0.1556	1.594	0.1841	3.267	0.1641	2.398
S3	ResNet34	Statistic	6.63M	0.0635	0.654	0.0920	0.824	0.1424	1.456	0.1434	2.574	0.1456	2.085
S4	ResNet101	Statistic	15.9M	0.0404	0.505	0.0666	0.651	0.1101	1.163	0.1162	2.025	0.1154	1.691
S5	ResNet152	Statistic	19.8M	0.0341	0.415	0.0623	0.606	0.1027	1.101	0.1189	2.052	0.1077	1.569
S6	ResNet34	MQMHA	8.61M	0.0471	0.675	0.0843	0.771	0.1353	1.369	0.1417	2.411	0.1408	1.955
S7	ResNet34-c64	MQMHA	27.8M	0.0510	0.638	0.0771	0.756	0.1275	1.353	0.1429	2.604	0.1375	2.001
S8	ResNet101	MQMHA	23.8M	0.0442	0.425	0.0615	0.602	0.1003	1.051	0.1050	1.885	0.1029	1.551
S9	ResNet101-c64	MQMHA	68.7M	0.0335	0.388	0.0575	0.576	0.0964	1.044	0.1124	1.961	0.1034	1.566
S10	ResNet152	MQMHA	27.7M	**0.0321**	0.378	0.0549	0.552	0.0898	0.980	**0.0967**	**1.765**	0.1036	1.457
S11	ResNet221	MQMHA	31.6M	0.0357	**0.330**	**0.0539**	**0.535**	**0.0855**	**0.966**	0.1009	1.795	**0.0976**	**1.420**
S1-S11	Fusion	-	-	0.0282	0.303	0.0483	0.491	0.0795	0.892	0.0931	1.645	0.0898	1.330

Comparisons between Different Backbones

Table 5. Voxceleb and VoxSRC results comparison between different embedding extractor backbones. '-64' denotes the ResNet with base channel number 64 and our default setup is 32. Large margin fine-tuning is applied for all the systems. The ResNet models with statistic pooling are trained with AAM loss and the ResNet models with MQMHA pooling are trained with AAM + Sub-center + Inter-Topk loss.

Index	Model Type	Pooling	Param #	Vox1-O		Vox1-E		Vox1-H		VoxSRC21-val		VoxSRC22-val	
				$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.05}$	EER	$DCF_{0.05}$	EER
S1	ECAPA (c=512)	ASP	6.20M	0.0901	0.771	0.1051	0.962	0.1670	1.784	0.1911	3.540	0.1676	2.499
S2	ECAPA (c=1024)	ASP	14.7M	0.0802	0.606	0.0889	0.813	0.1556	1.594	0.1841	3.267	0.1641	2.398
S3	ResNet34	Statistic	6.63M	0.0635	0.654	0.0920	0.824	0.1424	1.456	0.1434	2.574	0.1456	2.085
S4	ResNet101	Statistic	15.9M	0.0404	0.505	0.0666	0.651	0.1101	1.163	0.1162	2.025	0.1154	1.691
S5	ResNet152	Statistic	19.8M	0.0341	0.415	0.0623	0.606	0.1027	1.101	0.1189	2.052	0.1077	1.569
S6	ResNet34	MQMHA	8.61M	0.0471	0.675	0.0843	0.771	0.1353	1.369	0.1417	2.411	0.1408	1.955
S7	ResNet34-c64	MQMHA	27.8M	0.0510	0.638	0.0771	0.756	0.1275	1.353	0.1429	2.604	0.1375	2.001
S8	ResNet101	MQMHA	23.8M	0.0442	0.425	0.0615	0.602	0.1003	1.051	0.1050	1.885	0.1029	1.551
S9	ResNet101-c64	MQMHA	68.7M	0.0335	0.388	0.0575	0.576	0.0964	1.044	0.1124	1.961	0.1034	1.566
S10	ResNet152	MQMHA	27.7M	**0.0321**	0.378	0.0549	0.552	0.0898	0.980	**0.0967**	**1.765**	0.1036	1.457
S11	ResNet221	MQMHA	31.6M	0.0357	**0.330**	**0.0539**	**0.535**	**0.0855**	**0.966**	0.1009	1.795	**0.0976**	**1.420**
S1-S11	Fusion	-	-	0.0282	0.303	0.0483	0.491	0.0795	0.892	0.0931	1.645	0.0898	1.330

- ResNet-based model shows higher performance upper bound than ECAPA-TDNN [5].
- ResNet is more robust and performs better in the hard trials.

Model	Vox1-O	Vox1-E	Vox1-H
ECAPA (c=512)	0.979	1.226	2.311
ECAPA (c=1024)	0.846	1.024	2.017
ResNet34	0.899	1.126	2.048

Results under the training setup in [4]

[5] Desplanques, Brecht, Jenthe Thienpondt, and Kris Demuynck. "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification." arXiv preprint arXiv:2005.07143 (2020).

Comparisons between Different Backbones

Table 5. Voxceleb and VoxSRC results comparison between different embedding extractor backbones. '-64' denotes the ResNet with base channel number 64 and our default setup is 32. Large margin fine-tuning is applied for all the systems. The ResNet models with statistic pooling are trained with AAM loss and the ResNet models with MQMHA pooling are trained with AAM + Sub-center + Inter-Topk loss.

Index	Model Type	Pooling	Param #	Vox1-O		Vox1-E		Vox1-H		VoxSRC21-val		VoxSRC22-val	
				$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.05}$	EER	$DCF_{0.05}$	EER
S1	ECAPA (c=512)	ASP	6.20M	0.0901	0.771	0.1051	0.962	0.1670	1.784	0.1911	3.540	0.1676	2.499
S2	ECAPA (c=1024)	ASP	14.7M	0.0802	0.606	0.0889	0.813	0.1556	1.594	0.1841	3.267	0.1641	2.398
S3	ResNet34	Statistic	6.63M	0.0635	0.654	0.0920	0.824	0.1424	1.456	0.1434	2.574	0.1456	2.085
S4	ResNet101	Statistic	15.9M	0.0404	0.505	0.0666	0.651	0.1101	1.163	0.1162	2.025	0.1154	1.691
S5	ResNet152	Statistic	19.8M	0.0341	0.415	0.0623	0.606	0.1027	1.101	0.1189	2.052	0.1077	1.569
S6	ResNet34	MQMHA	8.61M	0.0471	0.675	0.0843	0.771	0.1353	1.369	0.1417	2.411	0.1408	1.955
S7	ResNet34-c64	MQMHA	27.8M	0.0510	0.638	0.0771	0.756	0.1275	1.353	0.1429	2.604	0.1375	2.001
S8	ResNet101	MQMHA	23.8M	0.0442	0.425	0.0615	0.602	0.1003	1.051	0.1050	1.885	0.1029	1.551
S9	ResNet101-c64	MQMHA	68.7M	0.0335	0.388	0.0575	0.576	0.0964	1.044	0.1124	1.961	0.1034	1.566
S10	ResNet152	MQMHA	27.7M	**0.0321**	0.378	0.0549	0.552	0.0898	0.980	**0.0967**	**1.765**	0.1036	1.457
S11	ResNet221	MQMHA	31.6M	0.0357	**0.330**	**0.0539**	**0.535**	**0.0855**	**0.966**	0.1009	1.795	**0.0976**	**1.420**
S1-S11	Fusion	-	-	0.0282	0.303	0.0483	0.491	0.0795	0.892	0.0931	1.645	0.0898	1.330

- Deep ResNet is better than the shallow one.

Comparisons between Different Backbones

Table 5. Voxceleb and VoxSRC results comparison between different embedding extractor backbones. '-64' denotes the ResNet with base channel number 64 and our default setup is 32. Large margin fine-tuning is applied for all the systems. The ResNet models with statistic pooling are trained with AAM loss and the ResNet models with MQMHA pooling are trained with AAM + Sub-center + Inter-Topk loss.

Index	Model Type	Pooling	Param #	Vox1-O		Vox1-E		Vox1-H		VoxSRC21-val		VoxSRC22-val	
				$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.05}$	EER	$DCF_{0.05}$	EER
S1	ECAPA (c=512)	ASP	6.20M	0.0901	0.771	0.1051	0.962	0.1670	1.784	0.1911	3.540	0.1676	2.499
S2	ECAPA (c=1024)	ASP	14.7M	0.0802	0.606	0.0889	0.813	0.1556	1.594	0.1841	3.267	0.1641	2.398
S3	ResNet34	Statistic	6.63M	0.0635	0.654	0.0920	0.824	0.1424	1.456	0.1434	2.574	0.1456	2.085
S4	ResNet101	Statistic	15.9M	0.0404	0.505	0.0666	0.651	0.1101	1.163	0.1162	2.025	0.1154	1.691
S5	ResNet152	Statistic	19.8M	0.0341	0.415	0.0623	0.606	0.1027	1.101	0.1189	2.052	0.1077	1.569
S6	ResNet34	MQMHA	8.61M	0.0471	0.675	0.0843	0.771	0.1353	1.369	0.1417	2.411	0.1408	1.955
S7	ResNet34-c64	MQMHA	27.8M	0.0510	0.638	0.0771	0.756	0.1275	1.353	0.1429	2.604	0.1375	2.001
S8	ResNet101	MQMHA	23.8M	0.0442	0.425	0.0615	0.602	0.1003	1.051	0.1050	1.885	0.1029	1.551
S9	ResNet101-c64	MQMHA	68.7M	0.0335	0.388	0.0575	0.576	0.0964	1.044	0.1124	1.961	0.1034	1.566
S10	ResNet152	MQMHA	27.7M	**0.0321**	0.378	0.0549	0.552	0.0898	0.980	**0.0967**	**1.765**	0.1036	1.457
S11	ResNet221	MQMHA	31.6M	0.0357	**0.330**	**0.0539**	**0.535**	**0.0855**	**0.966**	0.1009	1.795	**0.0976**	**1.420**
S1-S11	Fusion	-	-	0.0282	0.303	0.0483	0.491	0.0795	0.892	0.0931	1.645	0.0898	1.330

- Deep ResNet is better than the shallow one.
- Making ResNet wider has little effect on the performance.

Comparisons between Different Backbones

Table 5. Voxceleb and VoxSRC results comparison between different embedding extractor backbones. '-64' denotes the ResNet with base channel number 64 and our default setup is 32. Large margin fine-tuning is applied for all the systems. The ResNet models with statistic pooling are trained with AAM loss and the ResNet models with MQMHA pooling are trained with AAM + Sub-center + Inter-Topk loss.

Index	Model Type	Pooling	Param #	Vox1-O		Vox1-E		Vox1-H		VoxSRC21-val		VoxSRC22-val	
				$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.05}$	EER	$DCF_{0.05}$	EER
S1	ECAPA (c=512)	ASP	6.20M	0.0901	0.771	0.1051	0.962	0.1670	1.784	0.1911	3.540	0.1676	2.499
S2	ECAPA (c=1024)	ASP	14.7M	0.0802	0.606	0.0889	0.813	0.1556	1.594	0.1841	3.267	0.1641	2.398
S3	ResNet34	Statistic	6.63M	0.0635	0.654	0.0920	0.824	0.1424	1.456	0.1434	2.574	0.1456	2.085
S4	ResNet101	Statistic	15.9M	0.0404	0.505	0.0666	0.651	0.1101	1.163	0.1162	2.025	0.1154	1.691
S5	ResNet152	Statistic	19.8M	0.0341	0.415	0.0623	0.606	0.1027	1.101	0.1189	2.052	0.1077	1.569
S6	ResNet34	MQMHA	8.61M	0.0471	0.675	0.0843	0.771	0.1353	1.369	0.1417	2.411	0.1408	1.955
S7	ResNet34-c64	MQMHA	27.8M	0.0510	0.638	0.0771	0.756	0.1275	1.353	0.1429	2.604	0.1375	2.001
S8	ResNet101	MQMHA	23.8M	0.0442	0.425	0.0615	0.602	0.1003	1.051	0.1050	1.885	0.1029	1.551
S9	ResNet101-c64	MQMHA	68.7M	0.0335	0.388	0.0575	0.576	0.0964	1.044	0.1124	1.961	0.1034	1.566
S10	ResNet152	MQMHA	27.7M	**0.0321**	0.378	0.0549	0.552	0.0898	0.980	**0.0967**	**1.765**	0.1036	1.457
S11	ResNet221	MQMHA	31.6M	0.0357	**0.330**	**0.0539**	**0.535**	**0.0855**	**0.966**	0.1009	1.795	**0.0976**	**1.420**
S1-S11	Fusion	-	-	0.0282	0.303	0.0483	0.491	0.0795	0.892	0.0931	1.645	0.0898	1.330

- Attention based pooling and Sub-center + Inter-Topk loss can make further improvement

Comparisons between Different Backbones

Table 5. Voxceleb and VoxSRC results comparison between different embedding extractor backbones. '-64' denotes the ResNet with base channel number 64 and our default setup is 32. Large margin fine-tuning is applied for all the systems. The ResNet models with statistic pooling are trained with AAM loss and the ResNet models with MQMHA pooling are trained with AAM + Sub-center + Inter-Topk loss.

Index	Model Type	Pooling	Param #	Vox1-O		Vox1-E		Vox1-H		VoxSRC21-val		VoxSRC22-val	
				$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.05}$	EER	$DCF_{0.05}$	EER
S1	ECAPA (c=512)	ASP	6.20M	0.0901	0.771	0.1051	0.962	0.1670	1.784	0.1911	3.540	0.1676	2.499
S2	ECAPA (c=1024)	ASP	14.7M	0.0802	0.606	0.0889	0.813	0.1556	1.594	0.1841	3.267	0.1641	2.398
S3	ResNet34	Statistic	6.63M	0.0635	0.654	0.0920	0.824	0.1424	1.456	0.1434	2.574	0.1456	2.085
S4	ResNet101	Statistic	15.9M	0.0404	0.505	0.0666	0.651	0.1101	1.163	0.1162	2.025	0.1154	1.691
S5	ResNet152	Statistic	19.8M	0.0341	0.415	0.0623	0.606	0.1027	1.101	0.1189	2.052	0.1077	1.569
S6	ResNet34	MQMHA	8.61M	0.0471	0.675	0.0843	0.771	0.1353	1.369	0.1417	2.411	0.1408	1.955
S7	ResNet34-c64	MQMHA	27.8M	0.0510	0.638	0.0771	0.756	0.1275	1.353	0.1429	2.604	0.1375	2.001
S8	ResNet101	MQMHA	23.8M	0.0442	0.425	0.0615	0.602	0.1003	1.051	0.1050	1.885	0.1029	1.551
S9	ResNet101-c64	MQMHA	68.7M	0.0335	0.388	0.0575	0.576	0.0964	1.044	0.1124	1.961	0.1034	1.566
S10	ResNet152	MQMHA	27.7M	**0.0321**	0.378	0.0549	0.552	0.0898	0.980	**0.0967**	**1.765**	0.1036	1.457
S11	ResNet221	MQMHA	31.6M	0.0357	**0.330**	**0.0539**	**0.535**	**0.0855**	**0.966**	0.1009	1.795	**0.0976**	**1.420**
S1-S11	Fusion	-	-	0.0282	0.303	0.0483	0.491	0.0795	0.892	0.0931	1.645	0.0898	1.330

- Attention based pooling and Sub-center + Inter-Topk loss can make further improvement

Comparisons between Different Backbones

Table 5. Voxceleb and VoxSRC results comparison between different embedding extractor backbones. '-64' denotes the ResNet with base channel number 64 and our default setup is 32. Large margin fine-tuning is applied for all the systems. The ResNet models with statistic pooling are trained with AAM loss and the ResNet models with MQMHA pooling are trained with AAM + Sub-center + Inter-Topk loss.

Index	Model Type	Pooling	Param #	Vox1-O		Vox1-E		Vox1-H		VoxSRC21-val		VoxSRC22-val	
				$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.01}$	EER	$DCF_{0.05}$	EER	$DCF_{0.05}$	EER
S1	ECAPA (c=512)	ASP	6.20M	0.0901	0.771	0.1051	0.962	0.1670	1.784	0.1911	3.540	0.1676	2.499
S2	ECAPA (c=1024)	ASP	14.7M	0.0802	0.606	0.0889	0.813	0.1556	1.594	0.1841	3.267	0.1641	2.398
S3	ResNet34	Statistic	6.63M	0.0635	0.654	0.0920	0.824	0.1424	1.456	0.1434	2.574	0.1456	2.085
S4	ResNet101	Statistic	15.9M	0.0404	0.505	0.0666	0.651	0.1101	1.163	0.1162	2.025	0.1154	1.691
S5	ResNet152	Statistic	19.8M	0.0341	0.415	0.0623	0.606	0.1027	1.101	0.1189	2.052	0.1077	1.569
S6	ResNet34	MQMHA	8.61M	0.0471	0.675	0.0843	0.771	0.1353	1.369	0.1417	2.411	0.1408	1.955
S7	ResNet34-c64	MQMHA	27.8M	0.0510	0.638	0.0771	0.756	0.1275	1.353	0.1429	2.604	0.1375	2.001
S8	ResNet101	MQMHA	23.8M	0.0442	0.425	0.0615	0.602	0.1003	1.051	0.1050	1.885	0.1029	1.551
S9	ResNet101-c64	MQMHA	68.7M	0.0335	0.388	0.0575	0.576	0.0964	1.044	0.1124	1.961	0.1034	1.566
S10	ResNet152	MQMHA	27.7M	**0.0321**	0.378	0.0549	0.552	0.0898	0.980	**0.0967**	**1.765**	0.1036	1.457
S11	ResNet221	MQMHA	31.6M	0.0357	**0.330**	**0.0539**	**0.535**	**0.0855**	**0.966**	0.1009	1.795	**0.0976**	**1.420**
S1-S11	Fusion	-	-	0.0282	0.303	0.0483	0.491	0.0795	0.892	0.0931	1.645	0.0898	1.330

- Attention based pooling and Sub-center + Inter-Topk loss can make further improvement

Table 4: **Results comparison between different scoring methods and different strategies to combine multiple utterances within one enrollment speaker.** The results are from the ResNet34 model after stage I training.

Scoring Method	Enroll Comb	minDCF (0.01)	EER (%)
Cosine	Utt-Concat	0.4391	7.305
Cosine	Emb-Avg	0.4004	6.922
Cosine + ASnorm	Utt-Concat	0.4035	7.085
Cosine + ASnorm	Emb-Avg	**0.3707**	**6.590**
Cosine + ASnorm	Score-Avg	0.4419	6.759

Average the embeddings from different enrollment utterances is the best way to combine multiple enrollment utterances

Results and Analysis

CNSRC 2022 Results

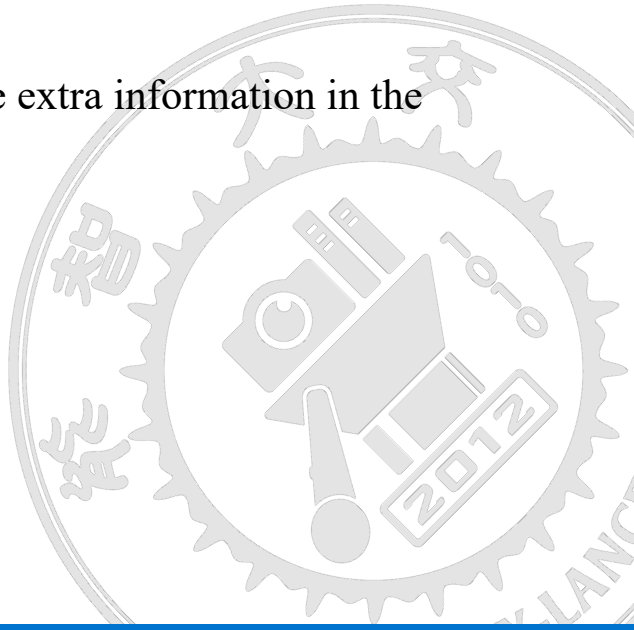
Table 6. Results on CNSRC 2022. Without specific notation, the models using statistic pooling and are trained with AAM loss.

Index	Model	Param #	$DCF_{0.01}$	EER (%)
S1	ResNet34 *	6.63M	0.3958	7.981
S2	ResNet34	6.63M	0.3707	6.590
S3	ResNet34 †	6.63M	0.3593	6.702
S4	ResNet152	19.8M	0.3386	5.762
S5	ResNet221	23.8M	0.3270	**5.543**
S6	ResNet293	28.6M	**0.3202**	5.553
S7	DF-ResNet	14.8M	0.3361	6.279
S8	ResNet34 + LM-FT †	6.63M	0.3458	6.319
S9	ResNet34 + LM-FT	6.63M	0.3543	6.221
S10	ResNet152 + LM-FT	19.8M	0.3251	5.452
S11	ResNet221 + LM-FT	23.8M	0.3179	5.284
S12	ResNet293 + LM-FT	28.6M	**0.3164**	**5.227**
S13	DF-ResNet + LM-FT	14.8M	0.3185	6.117
S9-S13	Fusion	-	0.2975	4.911
-	SpeakerIn System [29]	-	0.3185	5.953
-	STAP System [30]	-	0.3399	5.728

*: did not apply speed perturbation.
†: using MHMQA pooling and AAM + Sub-center + Inter-TopK loss.

Summary

- Data processing:
  - Speed perturbation is very necessary
  - Fbank feature is better than MFCC feature
- Embedding Extractor
  - ResNet-based model has higher performance upper bound than TDNN-based model
- Optimization & Training Strategy
  - Advanced margin-based softmax loss functions are very commonly used.
  - Large margin fine-tuning can improve the system's performance significantly.
- Back-end scoring
  - Quality-aware score calibration is very useful to introduce extra information in the scoring process.



References

[1] Wang, Weiqing, et al. "The dku-dukeece systems for voxceleb speaker recognition challenge 2020." arXiv preprint arXiv:2010.12631 (2020).

[2] Chen, Zhengyang, et al. "The SJTU X-LANCE Lab System for CNSRC 2022." arXiv preprint arXiv:2206.11699 (2022).

[3] Zhao, Miao, et al. "Multi-Query Multi-Head Attention Pooling and Inter-Topk Penalty for Speaker Verification." ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022.

[4] Thienpondt, Jenthe, Brecht Desplanques, and Kris Demuynck. "The idlab voxsrc-20 submission: Large margin fine-tuning and quality-aware score calibration in dnn based speaker verification." ICASSP 2021.

[5] Desplanques, Brecht, Jenthe Thienpondt, and Kris Demuynck. "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification." arXiv preprint arXiv:2005.07143 (2020).

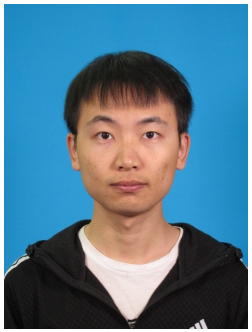
[6] Chen, Zhengyang, et al. "SJTU-AISPEECH System for VoxCeleb Speaker Recognition Challenge 2022." arXiv preprint arXiv:2209.09076 (2022).

[7] Chen, Zhengyang, et al. “BUILD A SRE CHALLENGE SYSTEM: LESSONS FROM VOXSRC 2022 AND CNSRC 2022.” submitted to ICASSP 2023.

Q & A!

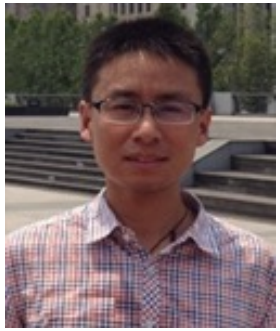
Zhengyang Chen

Bei Liu



Bing Han

Leying Zhang



Yanmin Qian

Houjun Huang

Xu Xiang