

# Speaker Recognition and Diarization for Alexa

Andreas Stolcke

Amazon Alexa AI



# How can Alexa (or Google Home or Siri) benefit from speaker recognition?

- Personalization
  - Music preferences, shopping lists, reminders, ...
  - Recommendations, news updates
- Authentication
  - Access to calendars, emails, ...
  - Commands with financial implications (purchases)
- Diarization
  - Follow human-human conversations, providing context to later requests
  - Resolve pronouns (“Play the movie you recommended to me earlier”)
  - <https://www.aboutamazon.com/news/devices/ai-advances-make-alexa-more-natural-conversational-and-useful>

# How are these applications different from other speaker recognition?

- Small set of target speakers (“household speaker recognition”)
  - Mostly known speakers mixed with possibly unknown speakers
- For stationary devices: relatively stable acoustic environment
- Speech input often includes a wakeword (“Alexa, ...”)
  
- For diarization: need for real-time, streaming processing
  
- Preference for compact models, limited computation

# Roadmap

- Intro: speaker recognition and diarization for digital assistants
- Advances in speaker recognition
  - Robust fusion of embeddings
  - Unsupervised training with graph-based inference
  - Embedding adaptation for households
- Pushing end-to-end neural speaker diarization (EEND)
  - Streaming EEND for unknown number of speakers
  - Advances in Transformer- and Conformer-based EEND

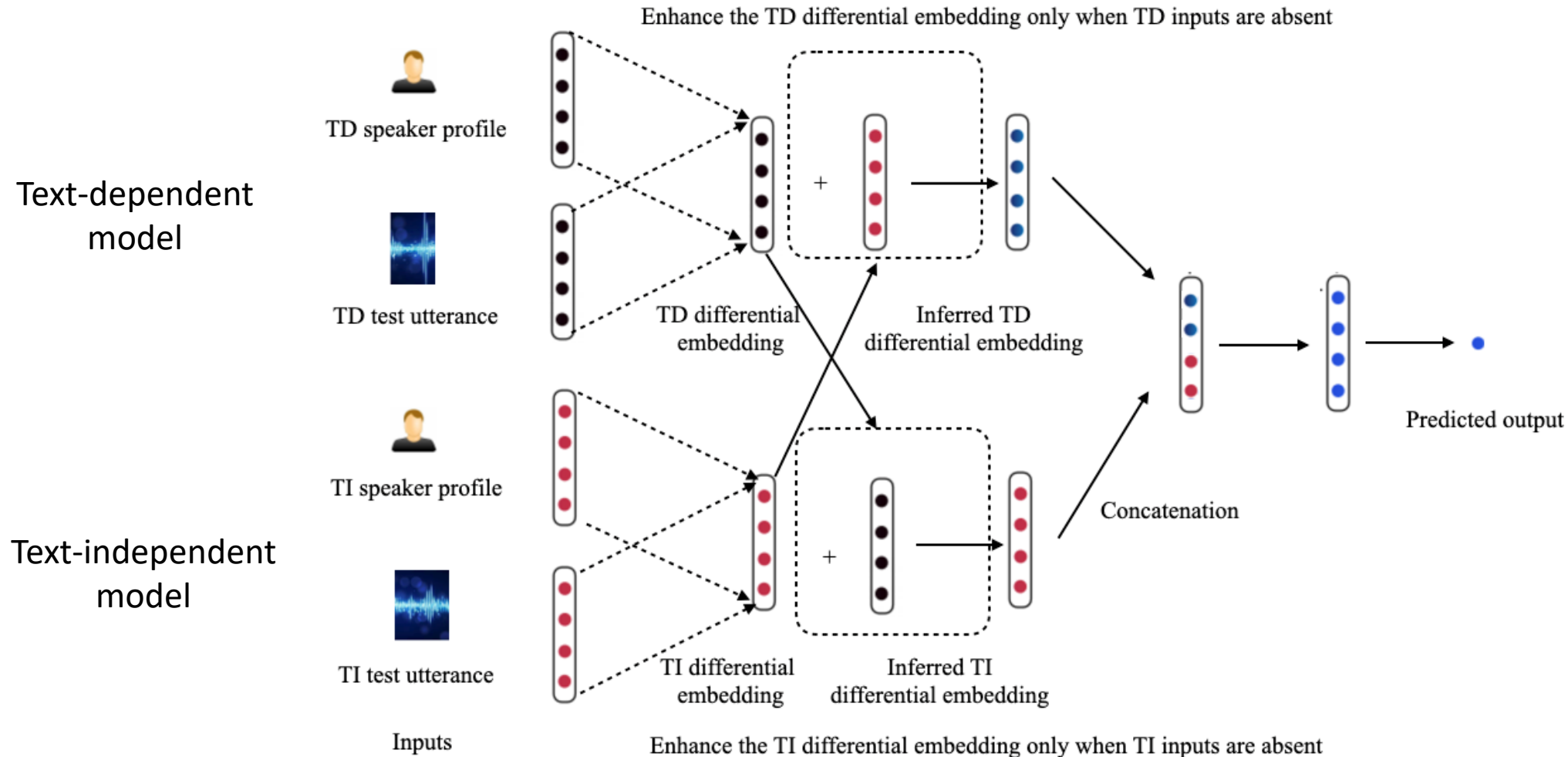
# Speaker ID for Households

# Robust Fusion of Embeddings

- Wakeword enabled text-dependent speaker ID
  - Text-dependent (TD): “Alexa, add eggs into my shopping list”
  - Text-independent (TI): “Alexa, add eggs into my shopping list”
- Fusing different models (TD + TI) is a great way to reduce error
- But what if one of the fused inputs is missing?
  - Wakeword missing / not recognized
  - Subsystem failure
- Approach: Estimate the missing speaker embeddings (TD or TI) from the available embedding, if needed

[R. Li, C. J.-T. Ju, Z. Chen, H. Mao, O. Elibol, A. Stolcke, “Fusion of Embeddings Networks for Robust Combination of Text Dependent and Independent Speaker Recognition”, Interspeech 2021](#)

# FOEnet: Fusion of Embeddings Network



# Data / Experiment

- Deidentified Alexa data
- LSTM embedding extractor, GE2E loss training
- FOEnext versus 4 baselines:
  - Single TI system (**GE2E**)
  - Average fusion (**AF**): takes the scores from TD and TI models as inputs and outputs the average of the two scores. Missing scores are replaced by a piecewise linear mapping
  - Score fusion (**SF**): takes the prediction scores from TD and TI as inputs and trains a neural network to make a joint prediction. -1 is used as input when inputs are missing
  - Enhanced score fusion (**E-SF**): same as SF except the missing input scores are inferred from the other system when needed



# Results

Relative reductions in false rejections at various false alarm rates (FAR)

Scenarios	Methods	Targeted FAR			
		0.8%	2.0%	5.0%	12.5%
TD&TI present	FOEnet vs GE2E	20.5	21.7	25.0	22.5
	FOEnet vs AF	8.3	12.1	13.1	14.6
	FOEnet vs SF	12.1	14.3	14.3	16.5
	FOEnet vs E-SF	10.5	13.3	13.7	17.4
TD absent	FOEnet vs GE2E	14.3	20.6	30.7	45.0
	FOEnet vs AF	15.2	20.6	30.7	46.0
	FOEnet vs SF	5.2	14.3	28.1	45.2
	FOEnet vs E-SF	4.0	13.0	26.6	43.8
TI absent	FOEnet vs GE2E	<i>not applicable</i>			
	FOEnet vs AF	34.9	40.7	49.9	53.3
	FOEnet vs SF	3.8	10.7	23.1	31.2
	FOEnet vs E-SF	10.4	15.7	25.8	32.8

# Conclusions

- Fusion of embeddings network allows robust combination of models (TD and TI) even when one input is missing
- False rejections reduced by 8.4% to 14.6% relative over score-level fusion by averaging
- Up to 53.3% reduced false rejections when one input is missing
- Inferring missing embeddings is better than inferring missing scores

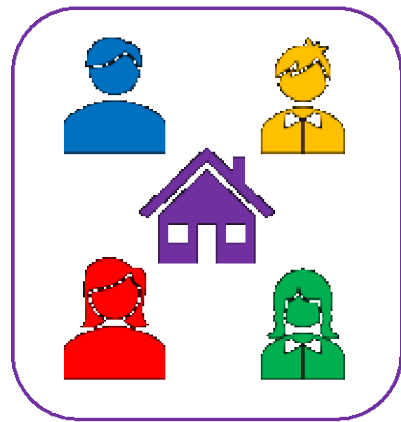
# Semi-supervised learning with graph-based inference

- We get only a few enrollment utterances but have a much larger set of unlabeled data from household speakers
- How can we use unsupervised learning? How can we do better than pseudo-labeling test utterances?
- Approach:
  - Label propagation (LP), a Graph-SSL method, propagates labels from labeled to unlabeled nodes over the graph
  - No retraining of the embedding model is required

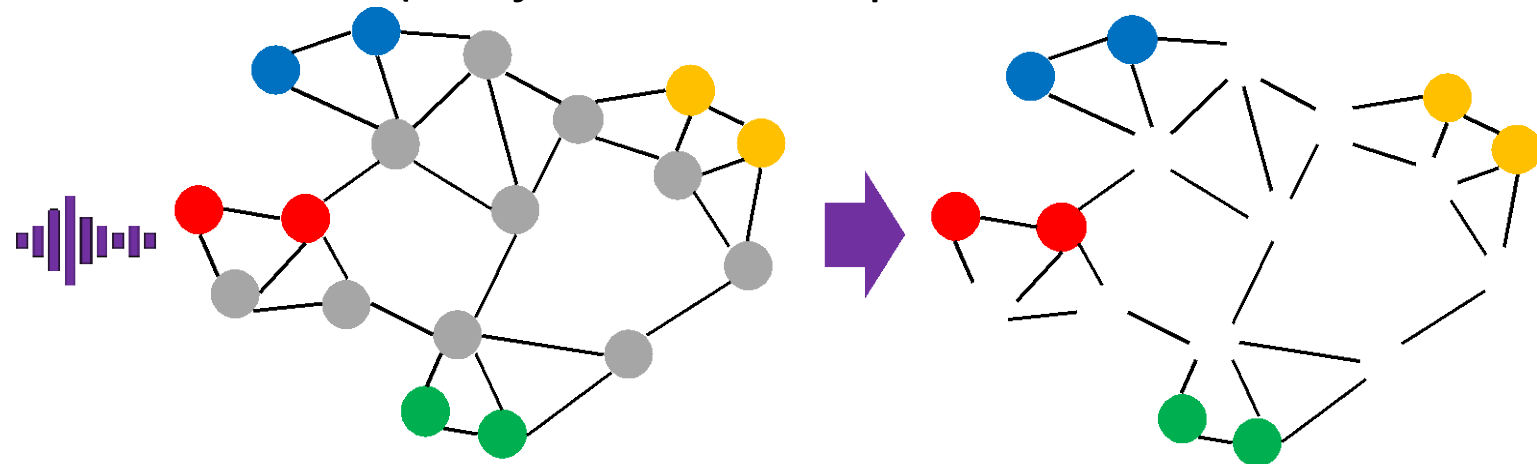
[L. Chen, V. Ravichandran, A. Stolcke, “Graph-based Label Propagation for Semi-Supervised Speaker Identification”, Interspeech 2021](#)

# Semi-supervised SID learning with Label Propagation

- Graphs are constructed per household:
  - utterances as graph nodes, for both labeled and unlabeled utterances
  - pairwise utterance similarity scores as edge weights
  - Prediction using Label Propagation: minimizes a global energy function that penalizes label differences on similar nodes
  - *Uses all pairwise similarities* (not just between profiles and test utterances)



Household



Utterance Graph

Prediction with Label Propagation

# Data / Experiments

- Datasets
  - VoxCeleb2 for embedding extractor training
  - VoxCeleb1 for evaluation
  - 112 4-speaker households as development
  - 200 4-speaker households as validation set
  - 10 utterances per speaker as the holdout dataset for evaluation
  - Remaining utterances used as labeled (enrollment) or unlabeled data for semi-supervised learning
- Training: GE2E, plain or with attention (GE2E-Att)
- Evaluation metric: SIER =  $1 - (\text{accuracy of top predicted speaker})$
- Unlabeled data:  $U=0,40,\dots,\text{all}$ ; labeled data  $L=1,2,\dots,\text{all}$

# Baselines / Unsupervised methods

## Baseline methods

- **CS**: cosine-score test utt with all labeled utts; pick highest
- **CSEA**: CS against the avg embedding of all labeled utts
- **2-CS**: Pseudo-label unlabeled utts with CS, then apply CS
- **2-CSEA**: Pseudo-label unlabeled utts with CS, then apply CSEA

## LP-based methods

- **LP**: LP over union of labeled, unlabeled and test utts
- **2-LP**: apply LP labeled and unlabeled utts only; then LP on test utts
- **2-LPEA**: apply LP labeled and unlabeled utts only; then use CSEA on test utts

# Results

Table 2: *SIER (%) on validation set with GE2E and GE2E-Att embeddings (L=2).*

Method	GE2E		GE2E-Att	
	<i>U</i> =40	<i>U</i> =All	<i>U</i> =40	<i>U</i> =All
CS	3.36	3.36	2.28	2.28
CSEA	3.06	3.06	2.08	2.08
2-CS	2.05	1.69	1.18	1.01
2-CSEA	1.93	1.39	1.15	0.87
LP	1.82	1.38	1.00	0.77
2-LP	1.73	<b>1.25</b>	0.94	<b>0.69</b>
2-LPEA	<b>1.49</b>	1.31	<b>0.88</b>	0.84

- LP-based methods outperform all baseline methods, for both GE2E and GE2E-Att embeddings
- Error reduced by 10%-23% relative
- Note: 2-LPEA requires no LP at runtime, only to update speaker profiles from unlabeled data

# Results: Varying amounts of data

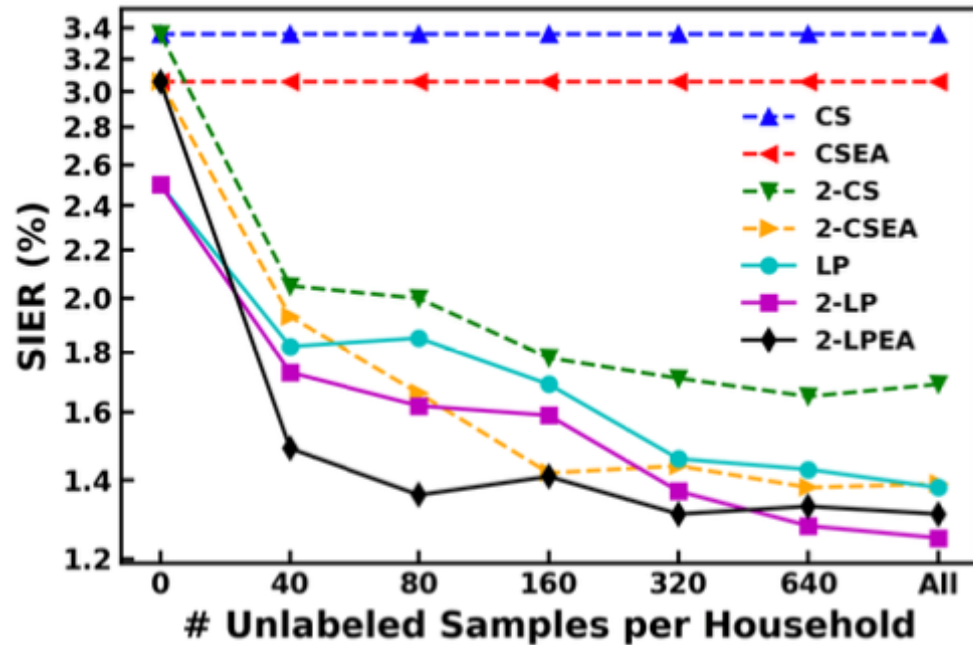


Figure 1: *SIER (%) in log scale as a function of the number of unlabeled samples per household*

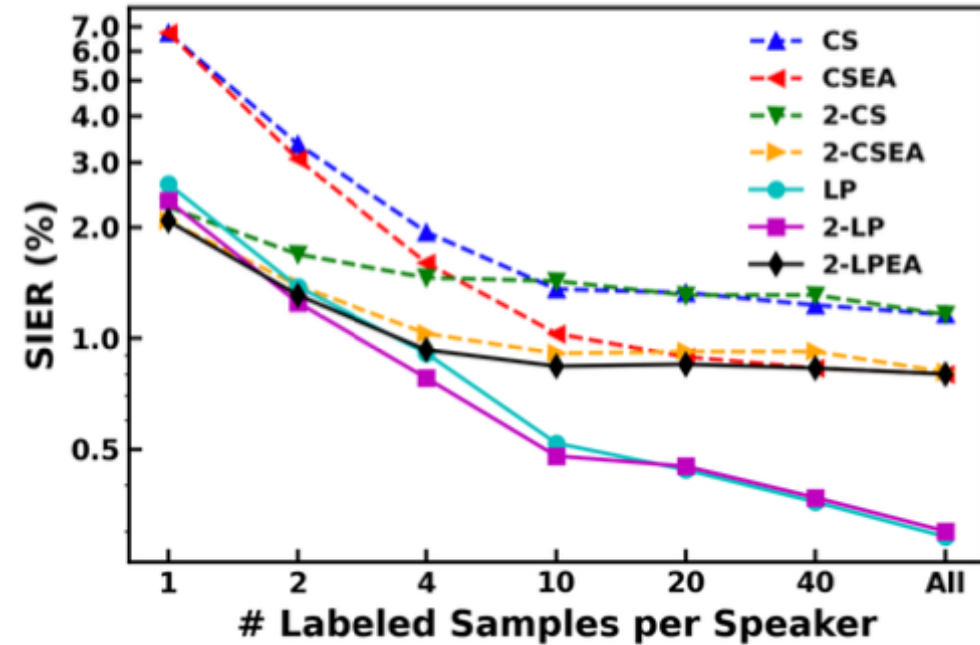


Figure 2: *SIER (%) in log scale as a function of the number of labeled samples per speaker*

- LP-based benefit from more unlabeled utts; always better than CS-based
- All methods benefit from more labeled data. However, relative error reduction with LP methods does not diminish

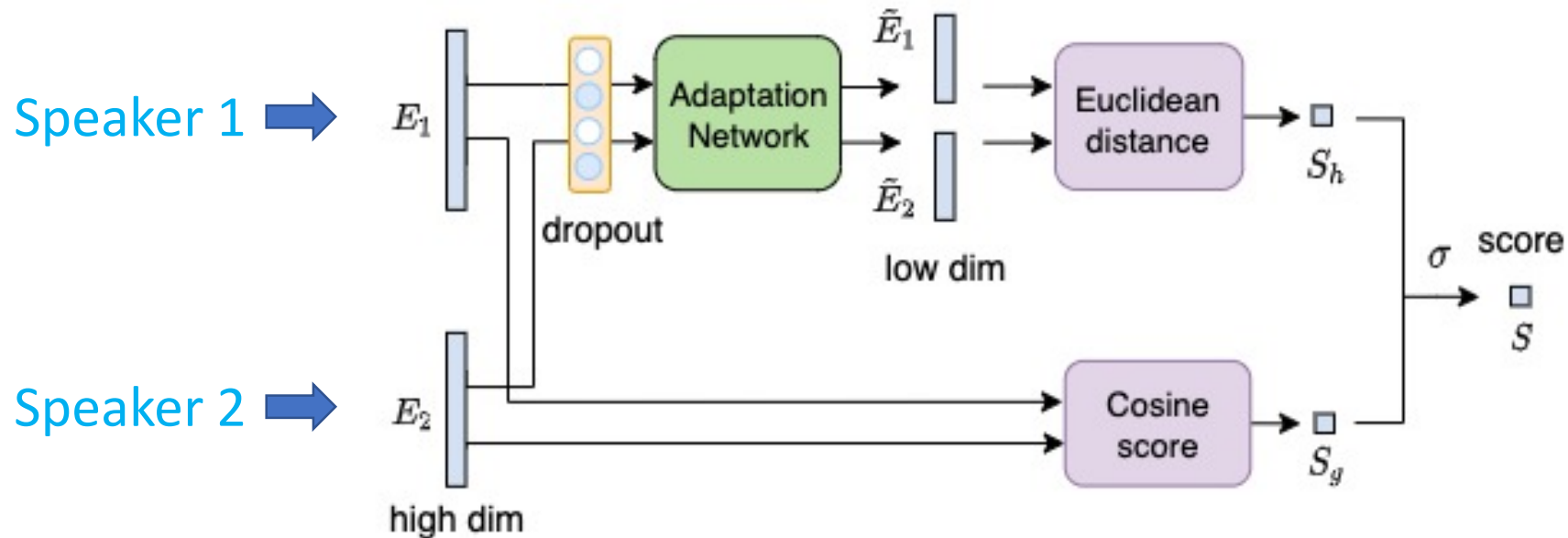


# Unsupervised Adaptation of Embeddings

- SID models are trained to discriminate among a large set of speakers
- But in a household scenario we only care about
  - a few speakers, often with a specific linguistic background
  - in a shared acoustic environment (device, room acoustics, background noise)
- Approach:
  - Learn an adapted speaker embedding specific to the household
  - Train it with self-supervision

Z. Tan, Y. Yang, E. Han, A. Stolcke, “Improving Speaker Identification for Shared Devices by Adapting Embeddings to Speaker Subsets”, to appear in ASRU 2021, <https://arxiv.org/abs/2109.02576>

# Embedding adaptation



- $\hat{E}_1, \hat{E}_2$  are adapted, low-dimensional embeddings
- $S_g$  is the *global* speaker similarity score, to be fused with
- $S_h$  the *household* similarity score
- Dropout on adaptation network inputs (synchronized) improves robustness

# Simulated Households: Experiments

- 2-7 speakers per HH drawn from VoxCeleb1, 1000x per HH size
- 4 enrollment utterances, up to 50 for adaptation, 10 for evaluation
- 250 guest speakers from outside HH
- Two sampling methods for HH speakers: *random* and *hard*
  - *Hard* means all speaker pairs within a HH have cos-similarity in the 98<sup>th</sup> percentile
- Global embedding model trained on VoxCeleb2, 5994 speakers
  - Half Resnet34, embedding dimension = 256
- Adaptation network: single layer, fully connected, ReLU
  - Output dimension = 32
- Score fusion network: single layer (logistic regression)

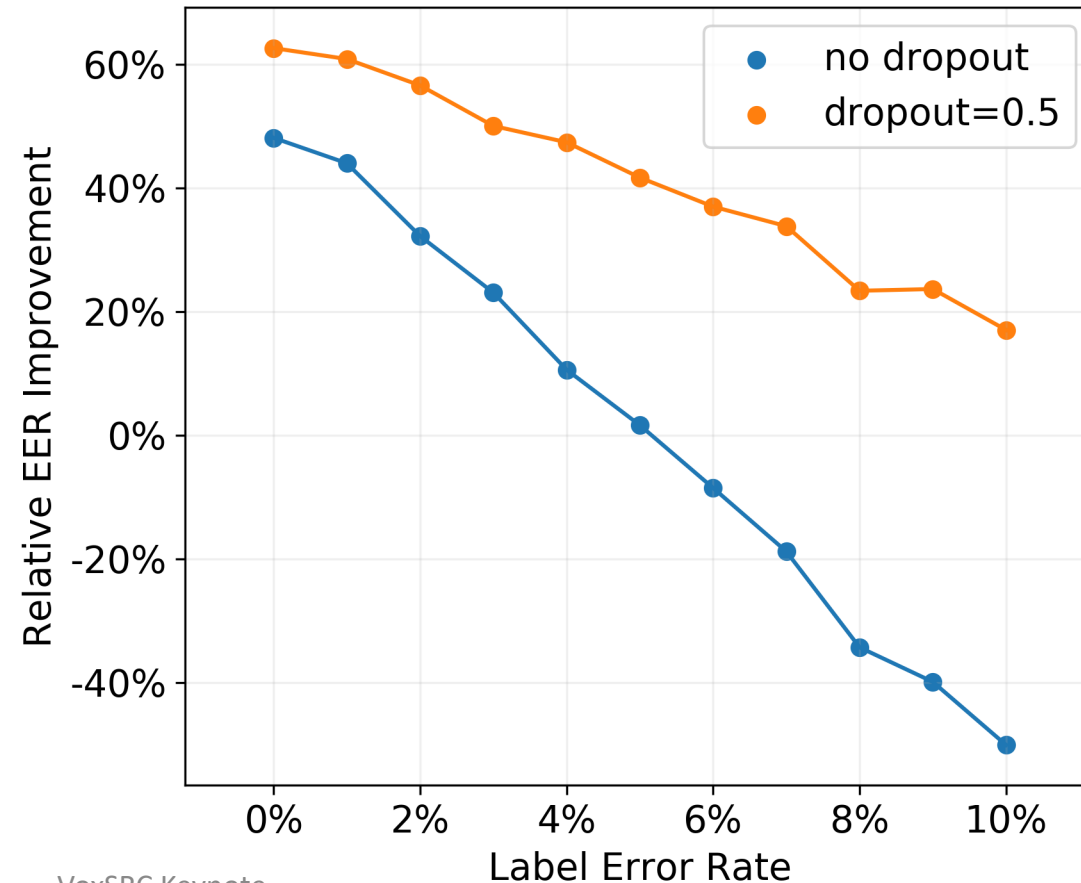
# Simulated households: Results (EER)

$N$	Random households				Hard households			
	Baseline	$S_h$ only	$S_h, S_g$ both	$S_h, S_g$ both (dropout=0.5)	Baseline	$S_h$ only	$S_h, S_g$ both	$S_h, S_g$ both (dropout=0.5)
2	2.66	1.87 (29.7%)	1.63 (38.7%)	<b>1.60 (39.8%)</b>	3.32	2.35 (29.2%)	1.98 (40.4%)	<b>1.82 (45.2%)</b>
3	3.40	2.59 (23.8%)	2.14 (37.1%)	<b>2.06 (39.4%)</b>	3.97	2.43 (38.8%)	2.06 (48.1%)	<b>1.70 (57.2%)</b>
4	3.85	3.14 (18.4%)	2.57 (33.2%)	<b>2.31 (40.0%)</b>	4.41	2.73 (38.1%)	2.29 (48.1%)	<b>1.65 (62.6%)</b>
5	3.98	3.81 (4.3%)	2.83 (28.9%)	<b>2.54 (36.2%)</b>	5.29	3.23 (38.9%)	2.56 (51.6%)	<b>1.54 (70.9%)</b>
6	4.58	4.68 (-2.2%)	3.41 (25.5%)	<b>2.83 (38.2%)</b>	6.74	4.48 (33.5%)	3.93 (41.7%)	<b>2.78 (58.8%)</b>
7	4.86	5.58 (-14.8%)	3.79 (22.0%)	<b>2.97 (38.9%)</b>	7.87	5.67 (28.0%)	4.34 (44.9%)	<b>2.89 (62.3%)</b>

- Household embeddings generally better than global
- Fusion of scores helps; dropout helps
- Random HHs are easier for baseline, but see less gain with adaptation
- Hard HHs: EER reduced 45% to 70% relative, depending on HH size  $N$

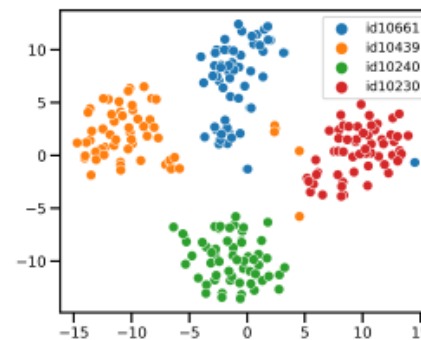
# Robustness to label noise

- In real life, adaptation labels will be errorful (from self-supervision)
- Dropout can help
- Simulated different levels of labeling error on *hard* households

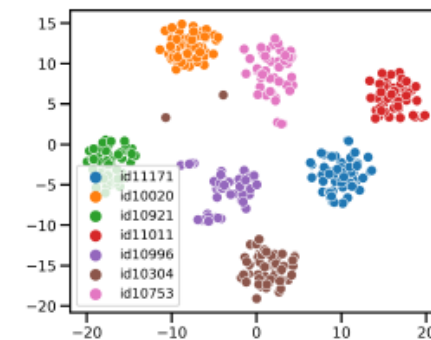


# Embedding space visualization

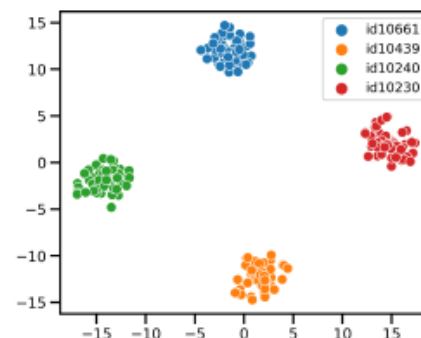
- Adaptation makes speaker clusters more compact
- Inter-cluster distances are more uniform



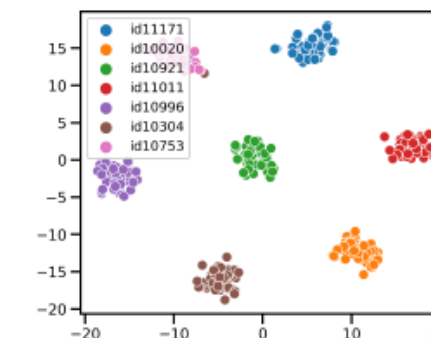
(a) Original ( $N=4$ )



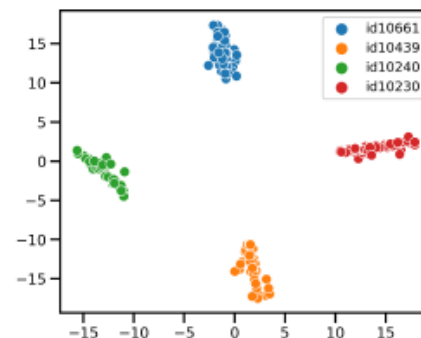
(b) Original ( $N=7$ )



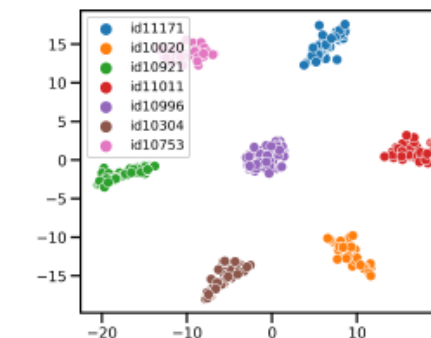
(c) No dropout ( $N=4$ )



(d) No dropout ( $N=7$ )



(e) Dropout=0.5 ( $N=4$ )



(f) Dropout=0.5 ( $N=7$ )

# Real households: Results

- Deidentified Alexa data
- Global embeddings from 3-layer LSTM, dimension = 512

	<b>EER reduction</b>
No dropout	42.5 %
With dropout	49.2 %

- Results are consistent with those seen on simulated data

# End-to-end Neural Diarization (EEND)



# Streaming EEND for Variable Number of Speakers

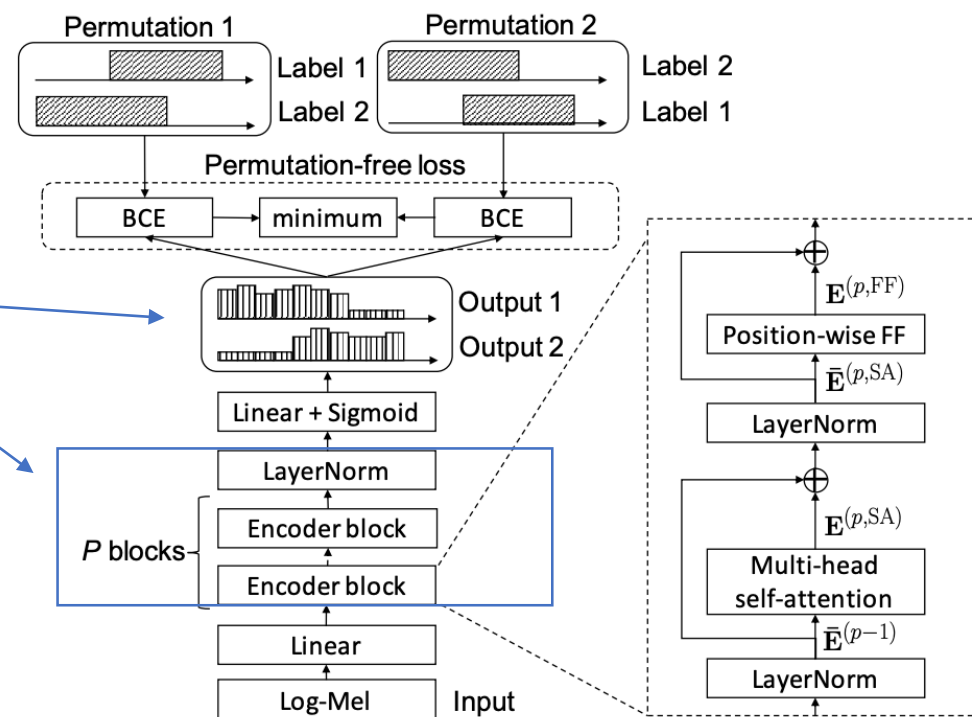
- End-to-end neural models have been widely adopted in ASR, SID and now diarization
- Advantages: compact, unified design, end-to-end optimization, handling of overlapping speakers
- Disadvantages: difficult to deal with *unknown number of speakers*, requires batch processing
- Recent EEND approach for variable no. of speakers: encoder-decoder-based attractor (EDA) networks by Horiguchi et al. (Interspeech 2020)
- Here: turn this into a *streaming (online) model: blockwise EDA-EEND*

[E. Han, C. Lee, A. Stolcke, “BW-EDA-EEND: Streaming End-to-End Neural Speaker Diarization for a Variable Number of Speakers”, ICASSP 2021](#)

# EEND with Self-Attention (SA-EEND)

- In SA-EEND [1], a **Transformer encoder** computes embeddings
- Only works with a fixed number of speakers

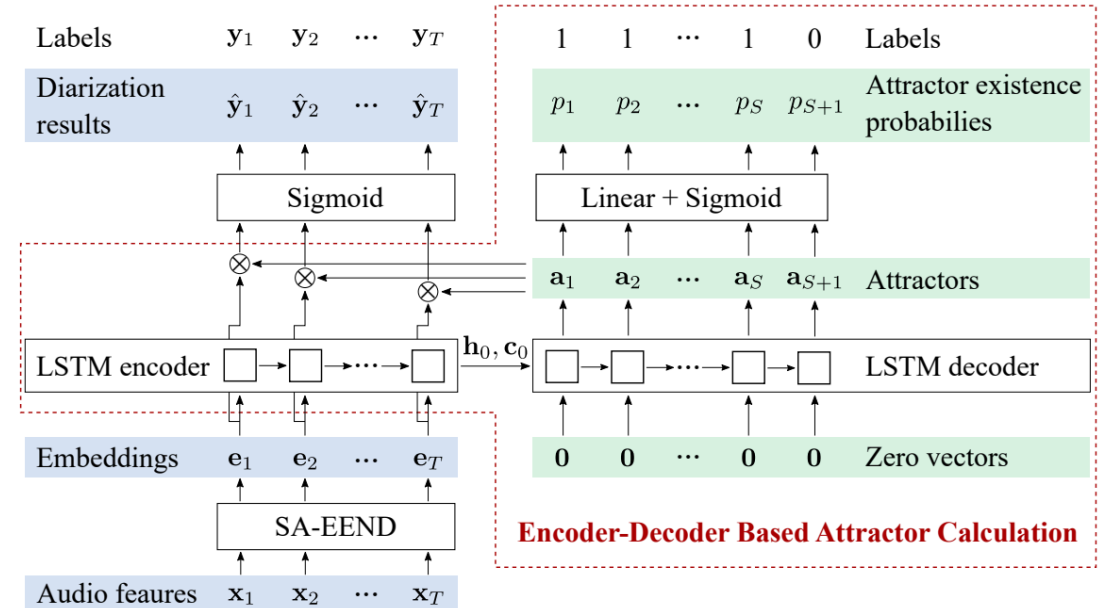
[1] Y. Fujita et al., “End-to-end neural speaker diarization with self-attention”, ASRU 2019



# EEND with Encoder-Decoder Attractor (EDA-EEND)

- In EDA-EEND [2], an LSTM encoder/decoder on top of SA-EEND is used to extract attractors for variable number of speakers
- Embeddings are shuffled to teach order-invariance
- Attractors are matched against embeddings to label speakers

[2] S. Horiguchi et al., “End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors”, Interspeech 2020



$$E = \text{TransformerEncoder}(X),$$

$$(h_0, c_0) = \text{LSTMEncoder}(E),$$

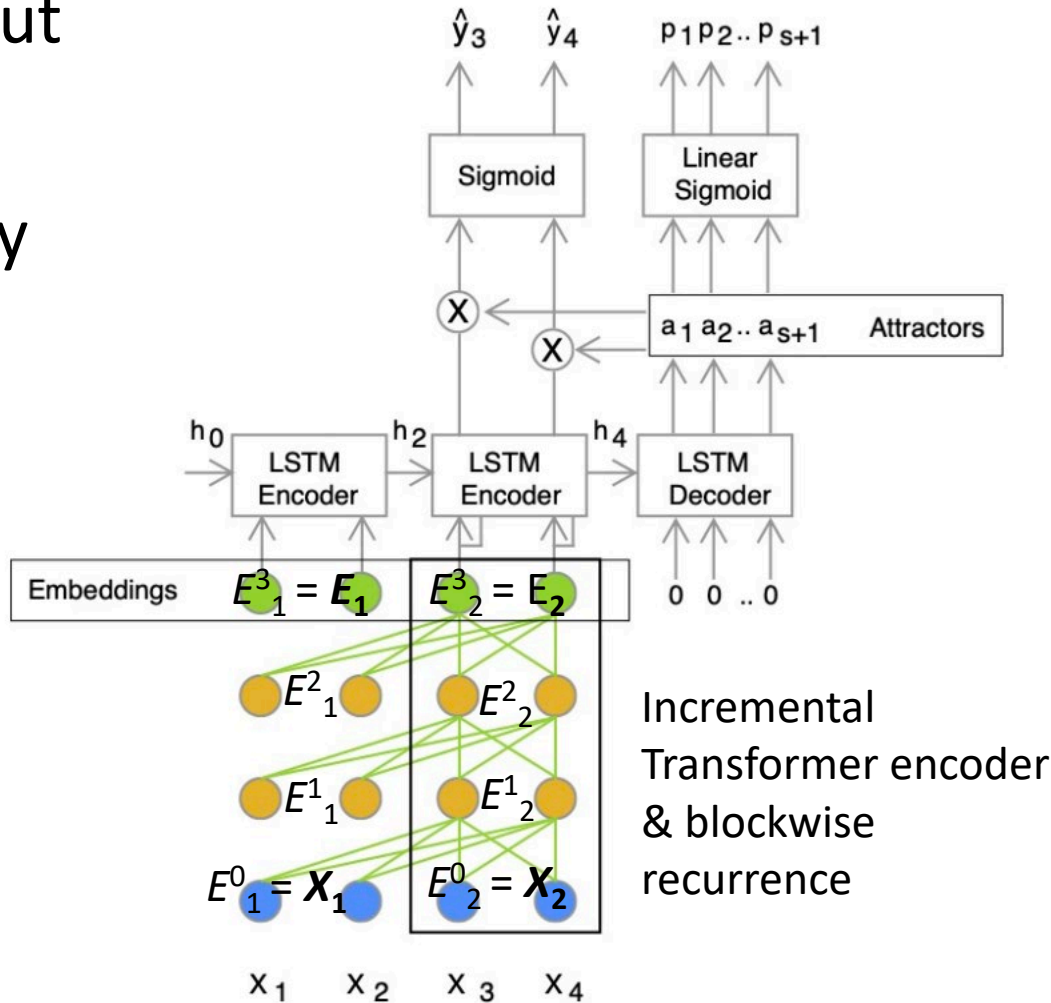
$$a_s, (h_s, c_s) = \text{LSTMDecoder}(0, (h_{s-1}, c_{s-1})) \text{ for } 1 \leq s \leq S + 1,$$

$$p_s = \sigma(\text{Linear}(a_s)) \text{ for } 1 \leq s \leq S + 1,$$

$$\hat{Y} = \sigma(EA^T), A = [a_1, \dots, a_S]$$

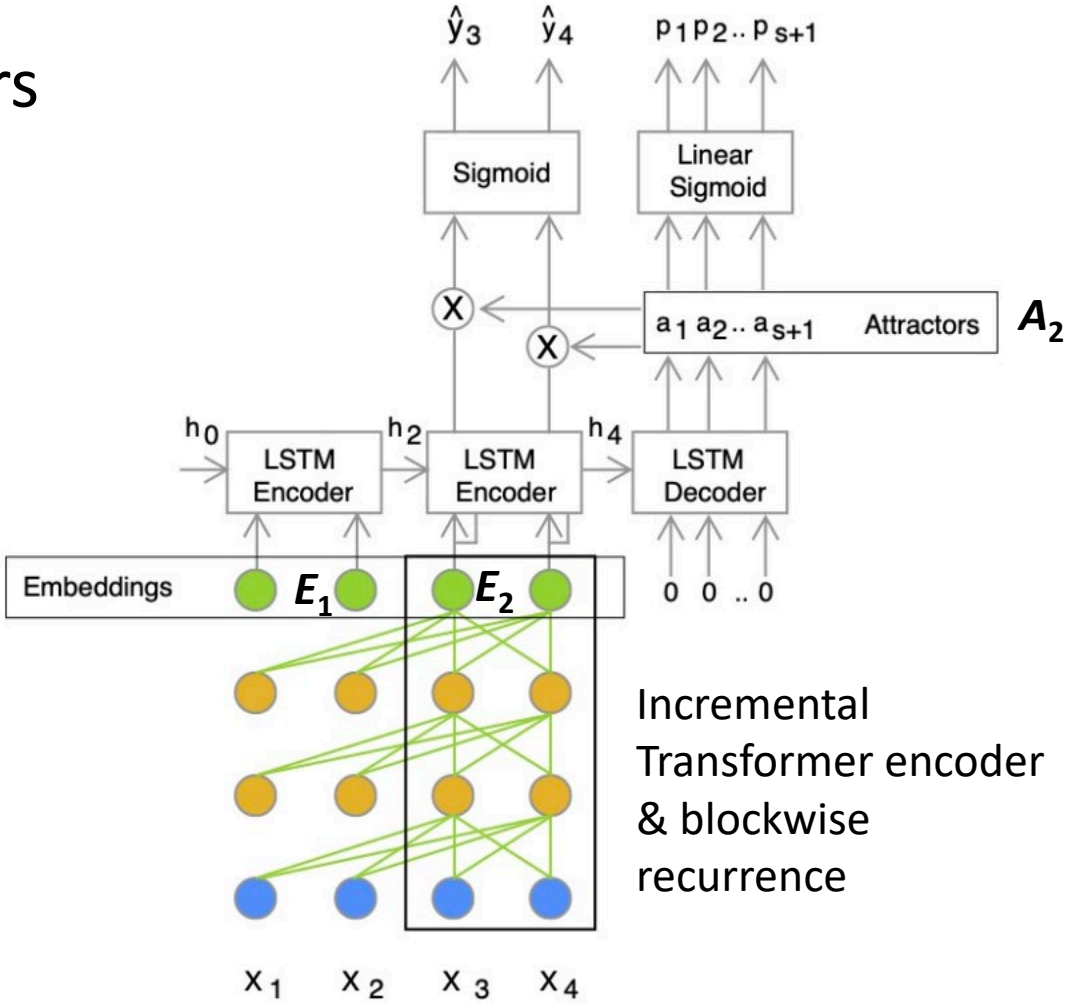
# Blockwise EDA-EEND: Encoder

- Structure computation over blocks of input
- Limit attention to left context
- Ensure computation linear in input size by
  - **incremental Transformer encoder**
  - **blockwise recurrence** in the hidden states, caching results of previous blocks
  - Similar to Transformer XL
- Two parameters
  - $W$  block width
  - $L$  number of prior blocks to attend to



# Blockwise EDA-EEND: Attractor decoder

- Block-dependent  $D$ -dimensional attractors ( $A_b$ ) are decoded using a blockwise unidirectional LSTM
- Attractor decoding and matching is independent of input length
- Special measures to ensure consistent attractor/output labels across blocks
  - Permute to minimize cosine distances to previous block
  - Average attractors across blocks
  - Shuffle embeddings across blocks



# Two variants of BW-EDA-EEND

- Two variants of BW-EDA-EEND that differ in how the attractors are computed
- **Limited-latency BW-EDA-EEND**
  - Computes attractors **for each block** and produces outputs with limited latency
  - Suitable for generating outputs online
- **Unlimited-latency BW-EDA-EEND**
  - Computes attractors **at the end of the inputs**
  - By limiting context size in the encoder, embedding computation is still linear in input length

# Results

Model	Attractor computation	Embedding shuffling	Number of speakers						
			1	2	3	4	2	3	4
			Simulated				CALLHOME		
Offline x-vector			37.42	7.74	11.46	22.45	15.45	18.01	22.68
Offline EDA-EEND	by utterance	within utterance	0.27	4.18	9.66	14.21	9.02	13.78	20.69
Online $W = 10, L = \infty$	by utterance	within block	<b>0.28</b>	<b>4.22</b>	<b>11.17</b>	<b>21.04</b>	<b>10.05</b>	<b>16.59</b>	25.50
Online $W = 10, L = 1$	by utterance	within block	<b>0.30</b>	<b>4.42</b>	13.35	22.71	<b>11.73</b>	19.87	28.03
Online $W = 10, L = \infty$	by block	within block	<b>1.72</b>	8.46	20.60	37.17	<b>12.91</b>	22.04	30.62
Online $W = 10, L = 1$	by block	within block	<b>1.85</b>	10.99	22.23	36.72	16.84	26.01	28.91
Online $W = 10, L = \infty$	by block	across blocks	<b>1.03</b>	<b>6.10</b>	12.58	<b>19.17</b>	<b>11.82</b>	18.30	25.93
Online $W = 10, L = 1$	by block	across blocks	<b>2.49</b>	<b>7.53</b>	16.65	24.50	16.18	19.35	27.52

- *Unlimited-latency* BW-EDA-EEND shows only moderate degradation of accuracy for up to two speakers, compared to offline EDA-EEND.
- *Limited-latency* BW-EDA-EEND has accuracy comparable to an offline, clustering-based system when frame-level embeddings are shuffled across blocks.

# Improved Transformer- and Conformer-based EEND

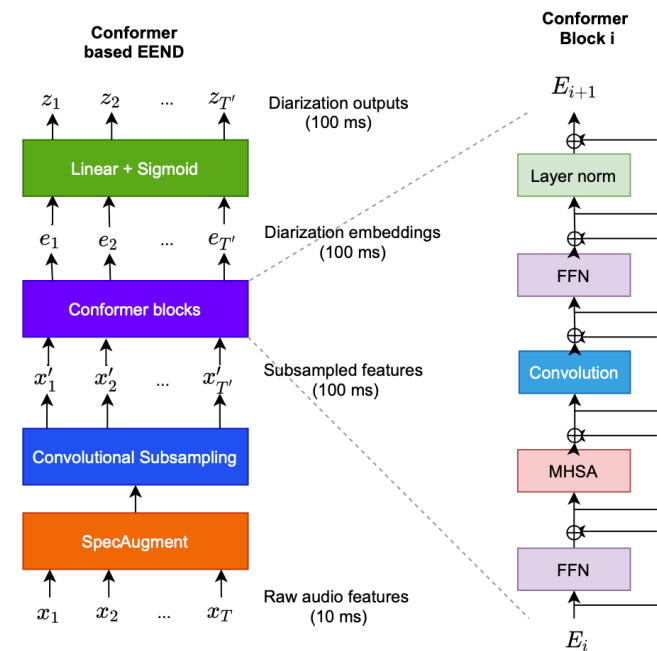
- Improve the basic SA-EEND approach by
  - applying data augmentation and convolutional subsampling
  - introducing Conformer layers
- Investigate the effect of train/test mismatch on EEND
- Quantify mismatch in turn-taking patterns (overlap and non-speech durations)
- Combine simulated with real data in training to overcome mismatch

[Y. C. Liu, E. Han, C. Lee, A. Stolcke, “End-to-end Neural Diarization: From Transformer to Conformer”, Interspeech 2021](#)



# Conformer-based EEND (CB-EEND)

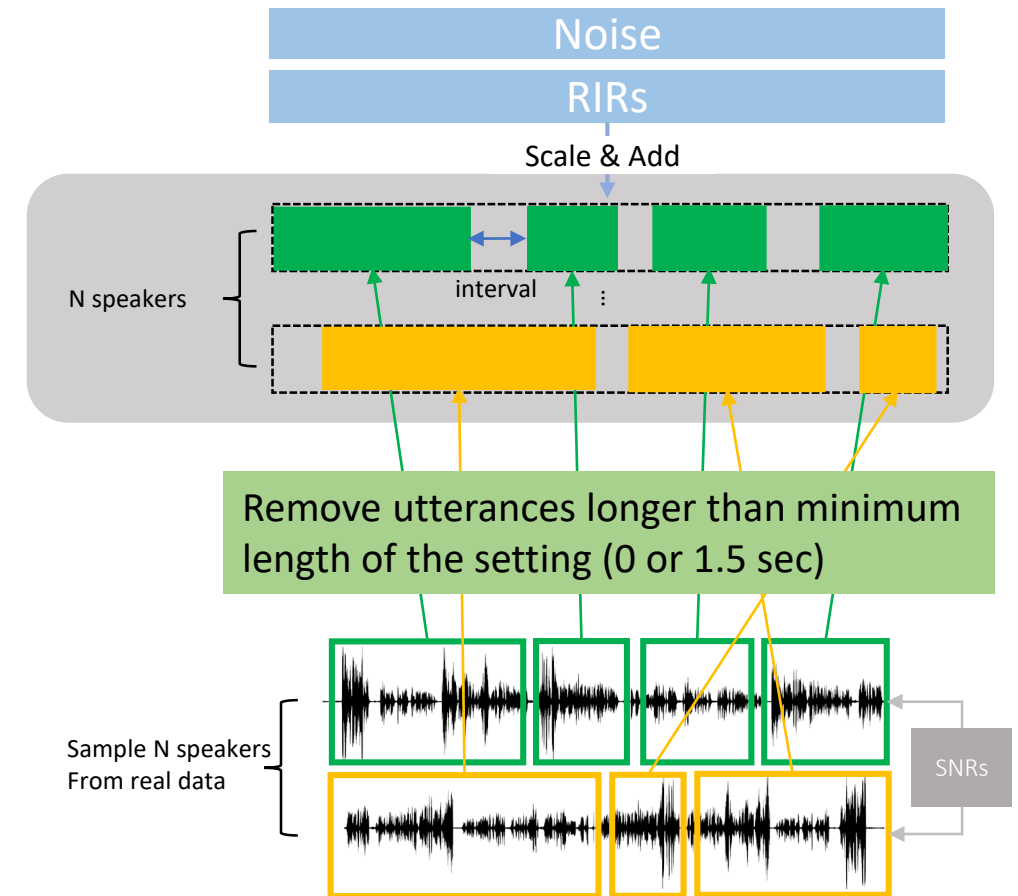
- Conformer combines convolution and self-attention.
- Diarization relies on both
  - Local cues at speaker turn-taking (convolution)
  - long-range comparisons of speaker characteristics (self-attention)
- CB-EEND replaces the Transformer encoder in TB-EEND with a Conformer encoder
- Each Conformer block composed of:
  - 1st feed-forward network (FFN) module
  - Multi-head self-attention module
  - Convolution module
  - 2nd feed-forward network (FFN) module



# Data Simulation and Selection

- Multiple sources and speaking styles
  - SWBD: conversations between strangers
  - SRE: NIST speaker rec. eval data
  - LS: LibriSpeech (audio books)
  - CH: CallHome – conversations among family (2-speaker conversations only)
- Simulated and real turn taking

Data Style	Source corpora	Min. Length	Avg. duration	Overlap ratio	Total duration
S0	SWBD+SRE+LS	1.5s	166.1s	48.4	9000h
S1	SWBD + SRE	1.5s	88.3s	34.5	2452h
S2	SWBD + SRE	0s	71.5s	26.7	2482h
R1	SWBD + SRE	1.5s	306.1s	6.5	2230h
R2	SWBD + SRE	0s	306.5s	18.3	2231h
CH	CALLHOME	0s	74.0s	14.0	3h



# Experiments

- Input features
  - 23- or 80-dimensional log-Mel-filterbanks, 25-ms frame length, 10-ms frame shift.
  - Subsampled by the factor of 10 and each frame represent 100 ms
  - **SA-EEND**: sub-sampling on stacked frames
  - **TB-EEND** and **CB-EEND**: convolutional sub-sampling
- Conformer parameters chosen to keep the total number of parameters unchanged or less than Transformer
  - 4 encoder blocks (P=4), 4 attention heads (H=4), Attention units = 256 (D=256)
  - Internal units of FFN: **SA-EEND / TB-EEND**: 1024, **CB-EEND**: 256
- Initial training with simulated training data
- Fine-tuning with CALLHOME training portion

# Results on simulated test data

- SA-EEND: baseline system (transformer-based)
- TB-EEND: enhanced transformer-based (data aug + conv. subsampl.)
- CB-EEND: conformer-based system

Training data style	Test data style	SA-EEND	TB-EEND	CB-EEND
S0	S0	5.09	3.44	<b>2.73</b>
S1	S1	6.50	3.54	<b>2.85</b>
S2	S2	8.76	5.94	<b>4.60</b>
R2	S2	29.65	32.61	<b>23.12</b>
S1+R2	S1	10.35	8.60	<b>3.28</b>

- Baseline DER for an x-vector diarization system is 28.8%
- CB-EEND consistently beats transformer-based models

# Results on Callhome data

- Similarity between training and test data is quantified by applying Earth Movers Distance to distribution of silence and overlap durations

Training data style	Test data style	Overlap similarity	Silence similarity	SA-EEND	TB-EEND	CB-EEND	Relative improvement with CB-EEND compared to	
							SA-EEND	TB-EEND
S0	CH	0.31	0.26	10.60	<b>7.63</b>	9.35	11.8%	-22.5%
S1	CH	0.50	0.59	10.52	<b>8.12</b>	9.70	7.8%	-19.5%
S2	CH	0.72	0.58	9.31	<b>8.10</b>	8.61	7.5%	-6.3%
R2	CH	0.89	0.96	10.11	9.61	<b>7.48</b>	26.0%	22.2%
S1+R2	CH	-	-	9.01	8.56	<b>6.82</b>	24.3%	20.3%

- Results correlate with train/test similarity
- CB-EEND trained on simulated does not generalize well to real data
- Pooling of simulated and real data overcomes mismatch

# Conclusion

- Data augmentation and convolutional subsampling layers enhance the SA-EEND
- We observe that Conformer is sensitive to mismatch between simulated training data and real conversational test data, which we quantify by similarity metrics based on overlap and silence region durations.
- Our proposed Conformer-based EEND is highly effective when trained on a mixture of simulated and real conversation data, which is not the case for a corresponding Transformer-based system.
- Overall, on two-speaker CALLHOME conversations, we achieve a relative error reduction of 24.3% over the best baseline SA-EEND training setup, and of 10.6% over the best augmented Transformer-based system.

[6] H. Li, P. Chaudhari, H. Yang, M. Lam, A. Ravichandran, R. Bhotika, and S. Soatto, "Rethinking the hyperparameters for fine-tuning," in *8th International Conference on Learning Representations (ICLR)*. Addis Ababa: OpenReview.net, Apr. 2020.

# Wrap-up

# Highlights

- Effective, robust combination of TI and TD speaker ID using a fusion of embedding network that estimates missing input data
- Graph-based LP is effective to better learn from unlabeled runtime utterances
- Learning of household-specific, low-dimensional embeddings is effective for semi-supervised adaptation of speaker ID
- EDA-EEND can be modified for online processing, using blockwise and incremental
- Improved end-to-end diarization using convolutional downsampling, conformer blocks (24.3% over the SA-EEND)
- “Conversational similarity” important for EEND generalization -- pooling simulated and real training data helps



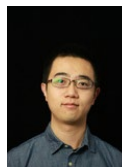
# Challenges

- Generalization to new locales & languages
- Fairness in speaker ID: recognize everyone equally well
  
- Generalization in end-to-end diarization, *or*
- How to simulate more realistic multi-party conversations?
- More parsimonious/elegant online EEND
  
- Integration of streaming multi-speaker ASR and diarization
- High-level, long-term modeling in speaker recognition and diarization

# Credits



Long Chen



Zeya Chen



Oguz Elibol



Eunjung (Christine) Han



Chelsea Ju



Chul Lee



Ruirui Li



Yi Chieh Liu



Hongda Mao



Venkatesh Ravichandran



Zhenning (Terry) Tan

More at [amazon.science](https://amazon.science)

Thank you!

Questions?

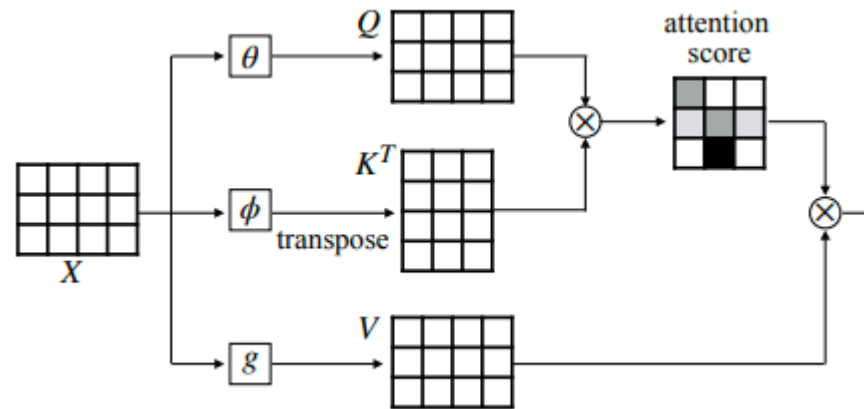
# Backup slides

# Self-attention and adversarial training for household speaker ID

- Fundamental issue #1: Modeling long-term correlations in speech
  - CNNs are good at capturing local patterns
  - RNNs/LSTMs are good at short-span sequential patterns
  - Approach: apply self-attention to input utterance
- Fundamental issue #2: Robustness to test/train mismatch
  - E.g., we may never observe certain kinds of noise in the training/enrollment data
  - Approach: augment training data with adversarial samples designed to break the classifier

[R. Li, J. Jiang, X. Wu, C. Hsieh, A. Stolcke, “Speaker Identification for Household Scenarios with Self-attention and Adversarial Training”, Interspeech 2020](#)

# Self-attention Mechanism

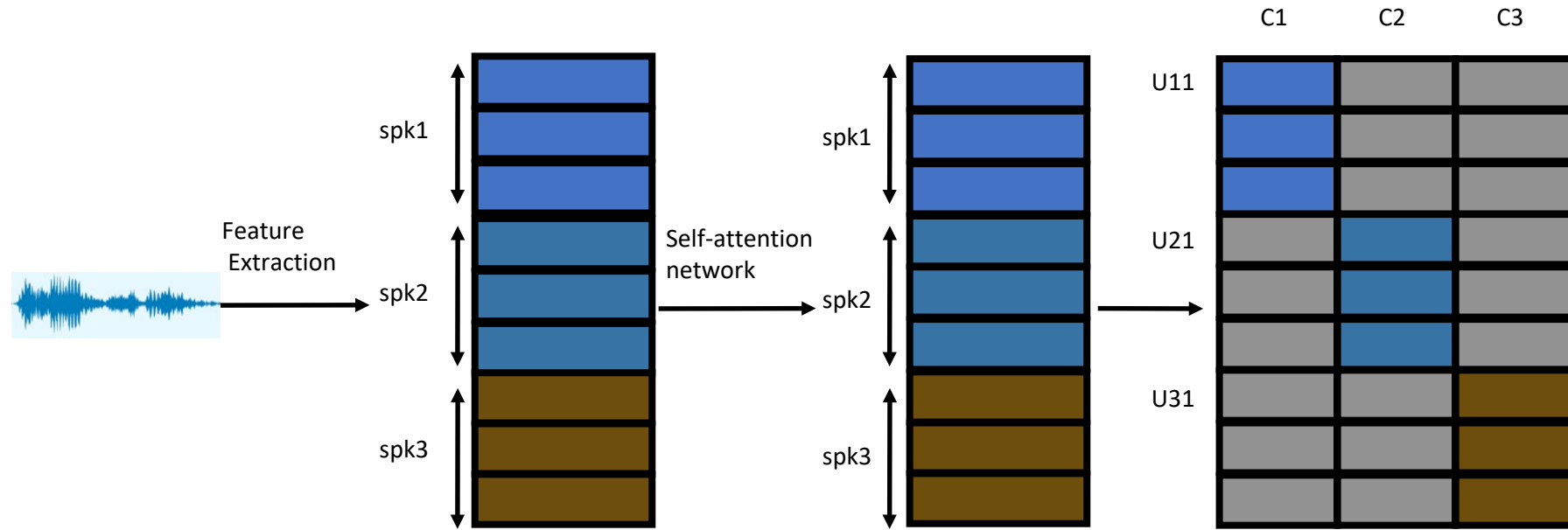


$$\text{Self-Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_Q}}\right) \mathbf{V}$$

$$\mathbf{SP}'_u = \mathbf{SP}_u + \mathbf{E}_p$$

$$\tilde{\mathbf{E}}_u = \text{Self-Att}(\mathbf{SP}'_u \cdot \mathbf{W}^Q, \mathbf{SP}'_u \cdot \mathbf{W}^K, \mathbf{SP}'_u \cdot \mathbf{W}^V)$$

# GE2G Loss with Self-attention



$$L(\mathbf{E}_{ji}|\Theta) = -\mathbf{S}_{ji,j}^{\Theta} + \log \sum_{k=1}^N \exp(\mathbf{S}_{ji,k}^{\Theta})$$

$$L(\mathbf{S}|\Theta) = \sum_{j,i} L(\mathbf{E}_{ji}|\Theta)$$

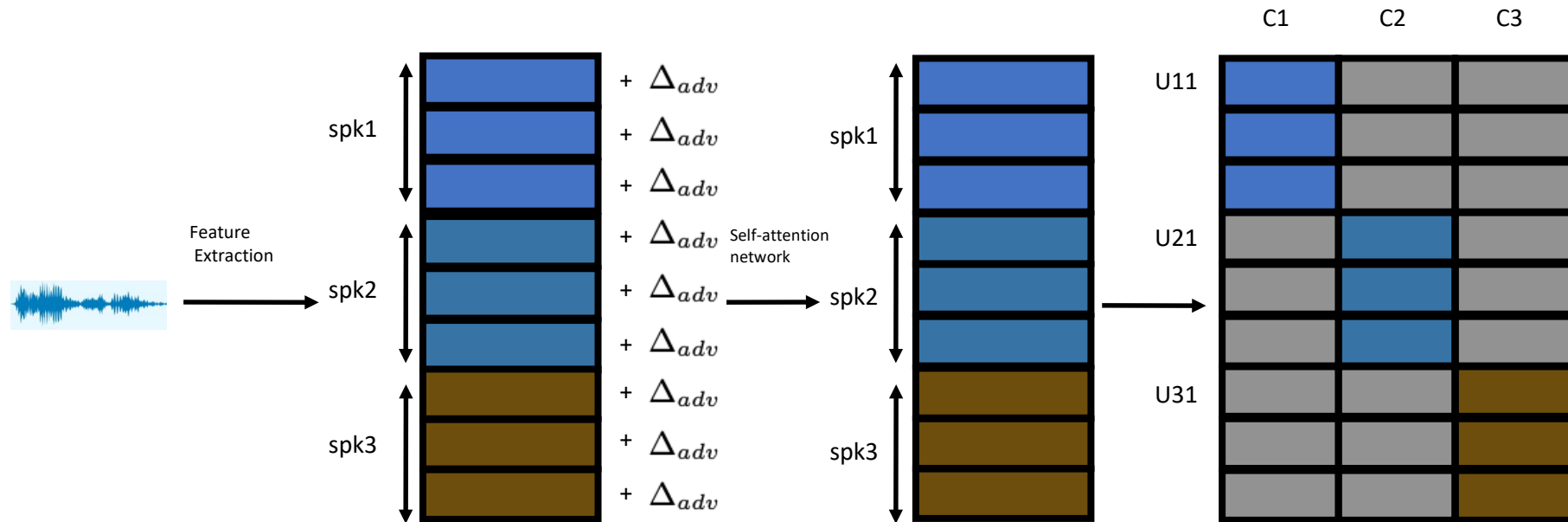
Pos Label




Neg Label



# Adversarial Training



 +  $\Delta_{adv}$  serves as purebred adversarial training examples

$$L_{adv}(\mathbf{S}|\Theta) = L(\mathbf{S}|\Theta) + \lambda L(\mathbf{S}_{\Delta_{adv}}|\Theta)$$

$$\text{where } \Delta_{adv} = \arg \max_{\Delta, \|\Delta\| \leq \epsilon} L(\mathbf{S}_{\Delta}|\hat{\Theta})$$



# Data / Experiments

- VCTK dataset
- 61 female, 47 male speakers
- Ages 10-40, 6 English-speaking locales
- 80% of speakers used as known users, 20% as unknown
- Evaluate on known-known and known-unknown speaker pairings
- Compare 4 models:
  - GE2E optimizes the speaker identification system by maximizing the similarity among utterances from the same speaker
  - SNL extends GE2E by adding an attention layer on top of LSTM to extract informative acoustic features
  - GE2E<sub>adv</sub> extends GE2E by conducting training in an adversarial manner
  - SNL<sub>adv</sub> conducts adversarial training on SNL

# Results

Table 3: *H-EER performance on known users*

Utt Length	Embed Size	GE2E	GE2E <sub>adv</sub>	SNL	SNL <sub>adv</sub>	SAASI
1.5s	64	6.95%	5.76%	4.22%	4.13%	3.67%
1.5s	128	6.49%	5.66%	4.03%	3.85%	3.39%

Table 4: *H-EER performance on new users*

Utt Length	Embed Size	GE2E	GE2E <sub>adv</sub>	SNL	SNL <sub>adv</sub>	SAASI
1.5s	64	13.84%	13.58%	10.86%	9.31%	6.56%
1.5s	128	13.11%	12.73%	10.30%	9.11%	6.39%