

# **TURN-TO-DIARIZE: ONLINE SPEAKER DIARIZATION CONSTRAINED BY TRANSFORMER TRANSDUCER SPEAKER TURN DETECTION**

*Wei Xia\*, Han Lu\*, Quan Wang\*, Anshuman Tripathi, Yiling Huang, Ignacio Lopez Moreno, Hasim Sak*

Google LLC, USA

{ *ericwxia*, *luha*, *quanw*, *anshumant*, *yilinghuang*, *elnota*, *hasim* } @google.com

# Problem #1 - Data for Supervised Learning

Start: 0.0, end: 1.2, speaker: A, content: good morning

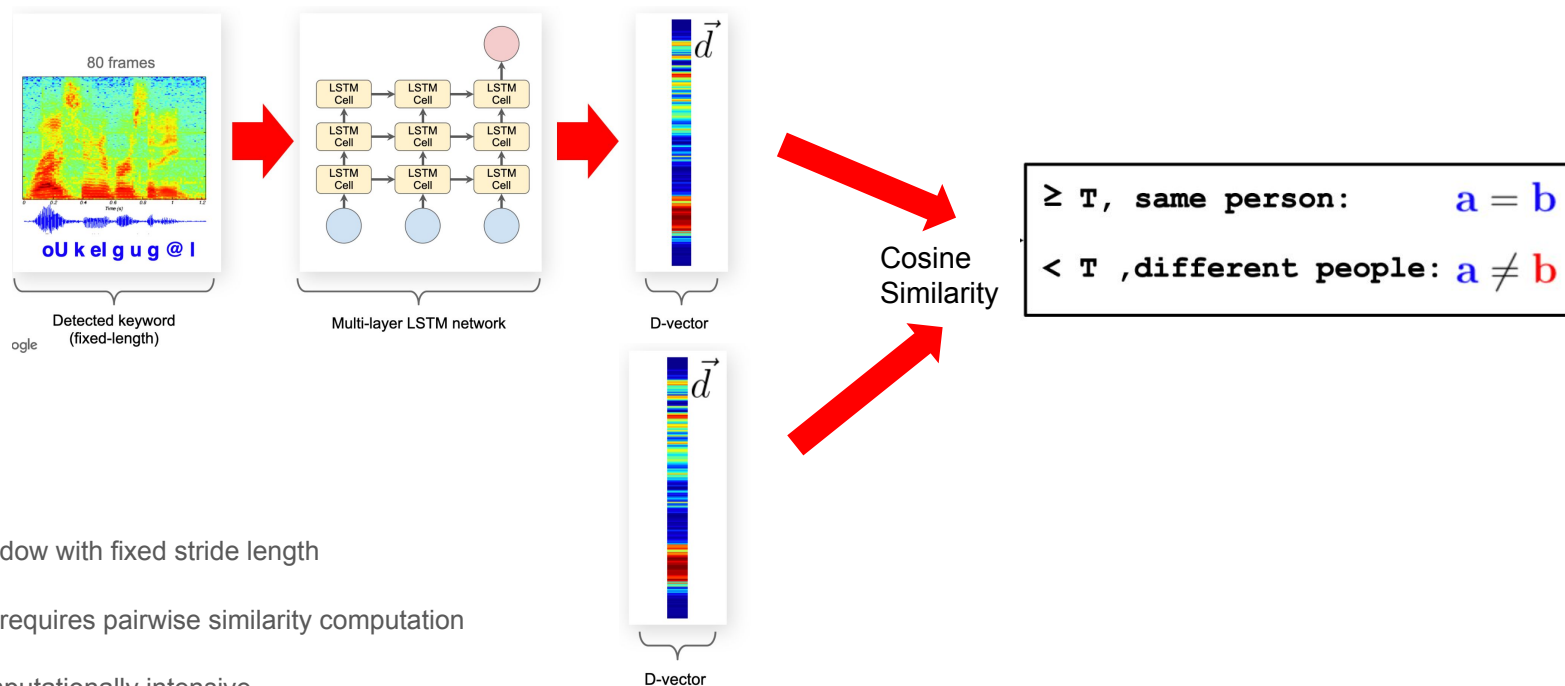
Start: 1.3, end: 4.4, speaker: B, content: morning, how are you

Start: 6.7, end: 9.4, speaker: A, content: good what about you

Time consuming, error prone for human transcribers

-> Higher Cost

## Problem #2 - Sample and Cluster D vectors



Sliding window with fixed stride length

Clustering requires pairwise similarity computation

Highly computationally intensive

# Key Insight #1 - Convert To Speaker Turn

Start: 0.0, end: 1.2, speaker: A, content: good morning

Start: 1.3, end: 4.4, speaker: B, content: morning, how are you

Start: 6.7, end: 9.4, speaker: A, content: good what about you



good morning <st>

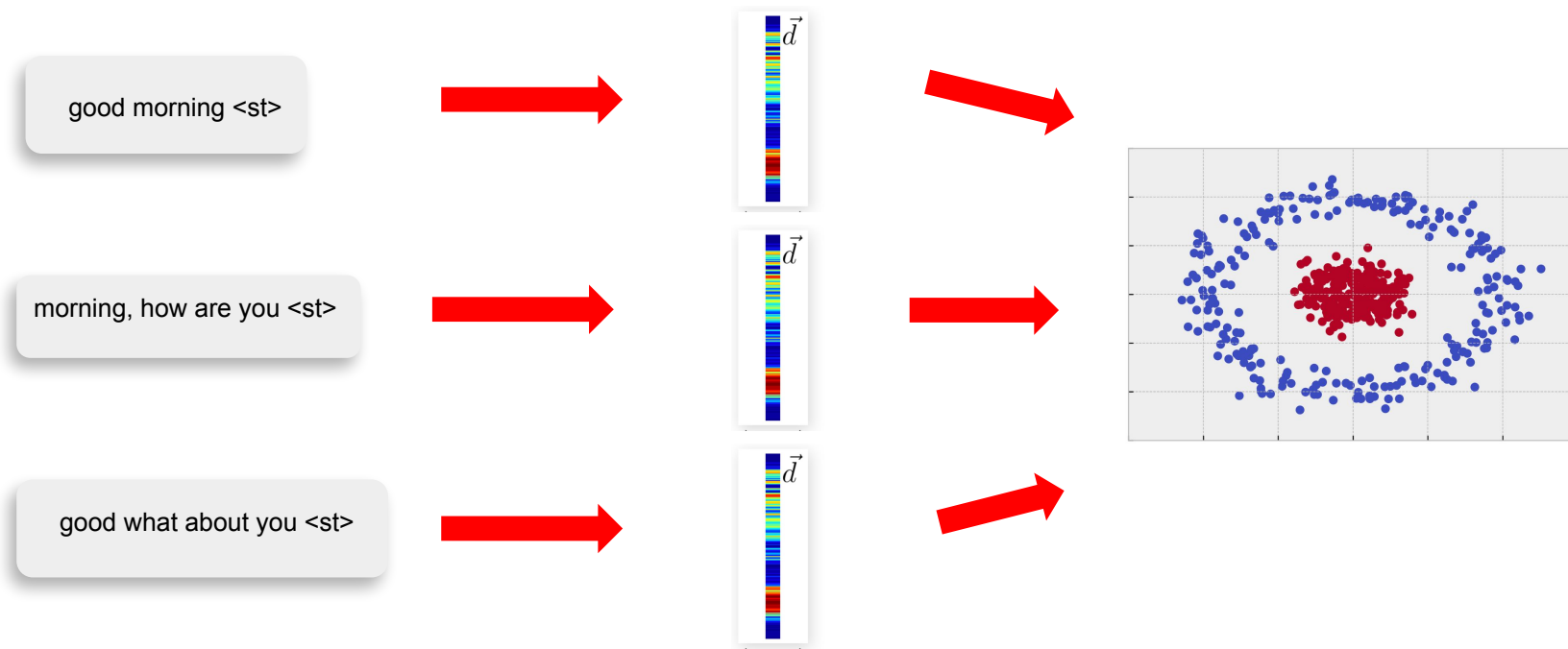
morning, how are you <st>

good what about you <st>

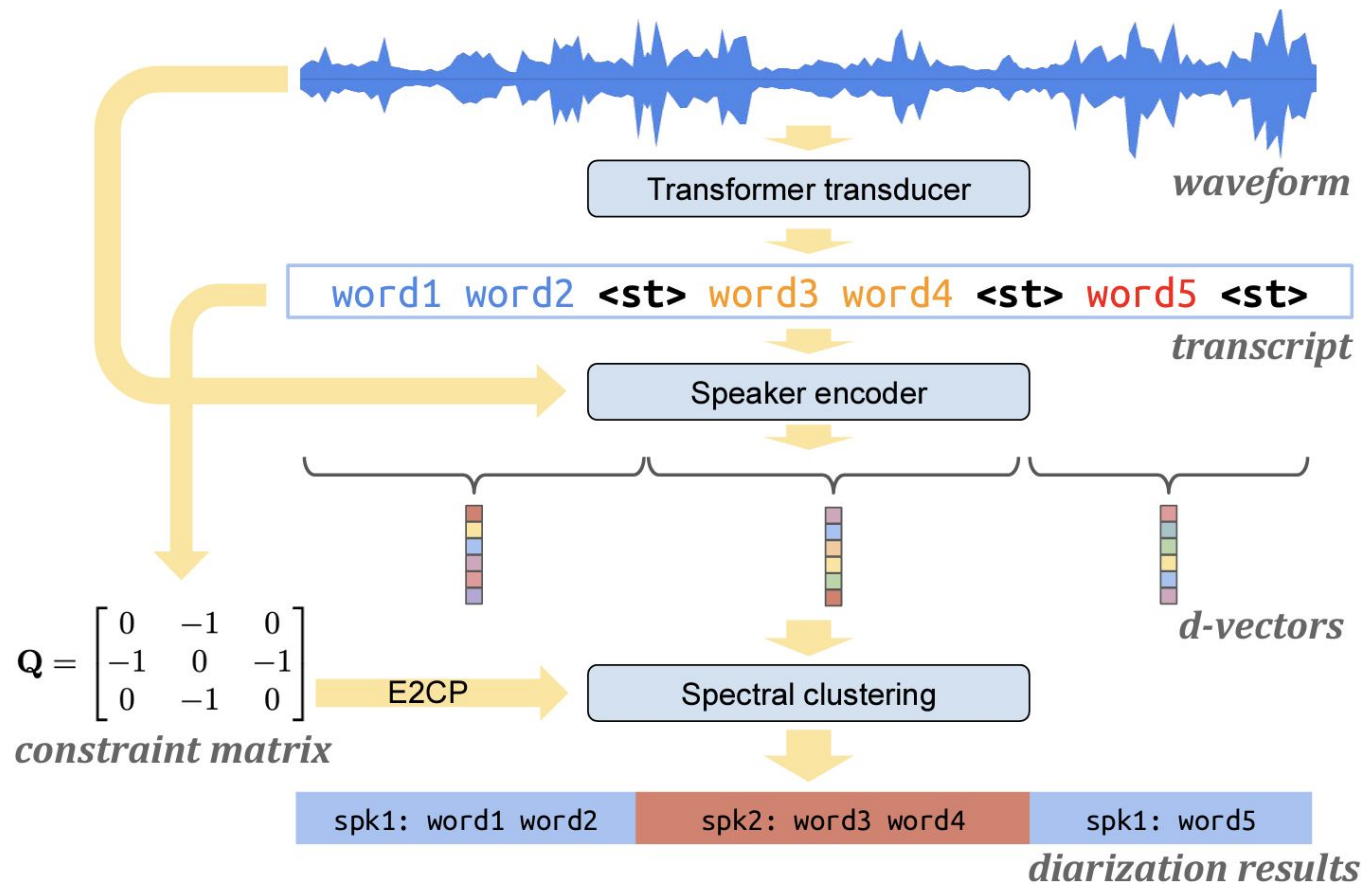
1. Less Error Prone
2. Easy to modify current transcription workflows
3. Less time, less cost

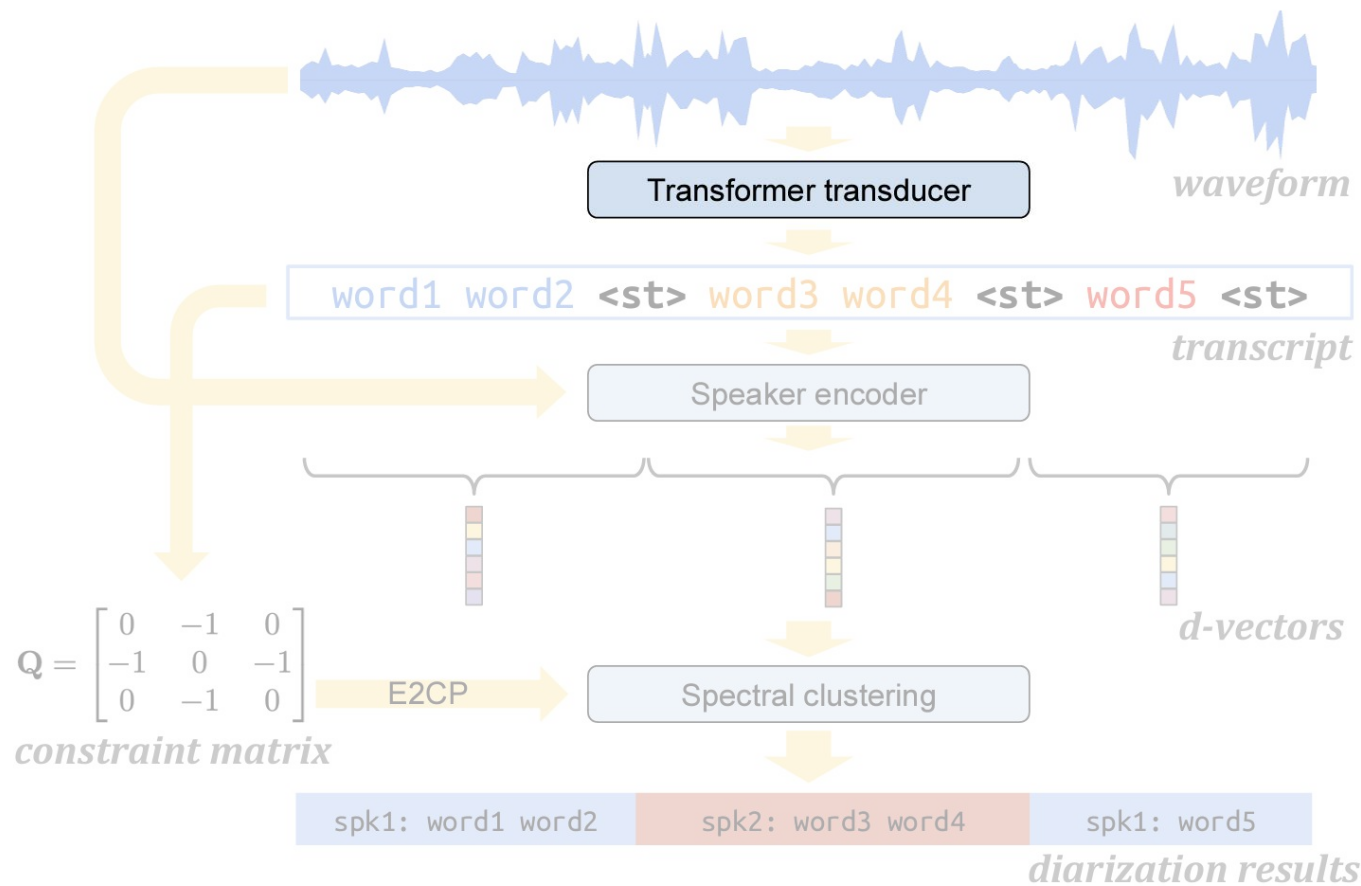
# Key Insight #2 - Use Speaker Turn information

1. Calculate D vector for each Speaker Turn **instead** of sliding window (or max 6 secs)
2. Utilize Speaker Turn information to **constrain** the clustering

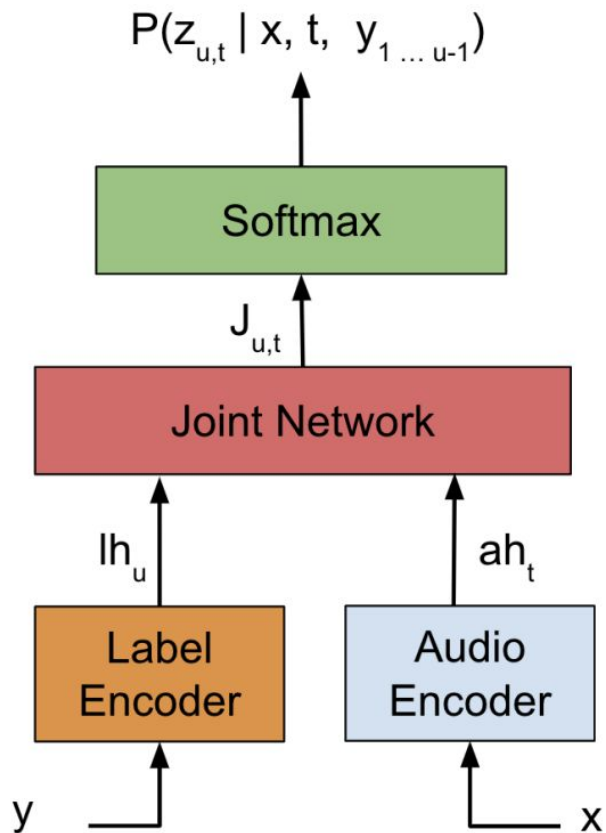


# Model

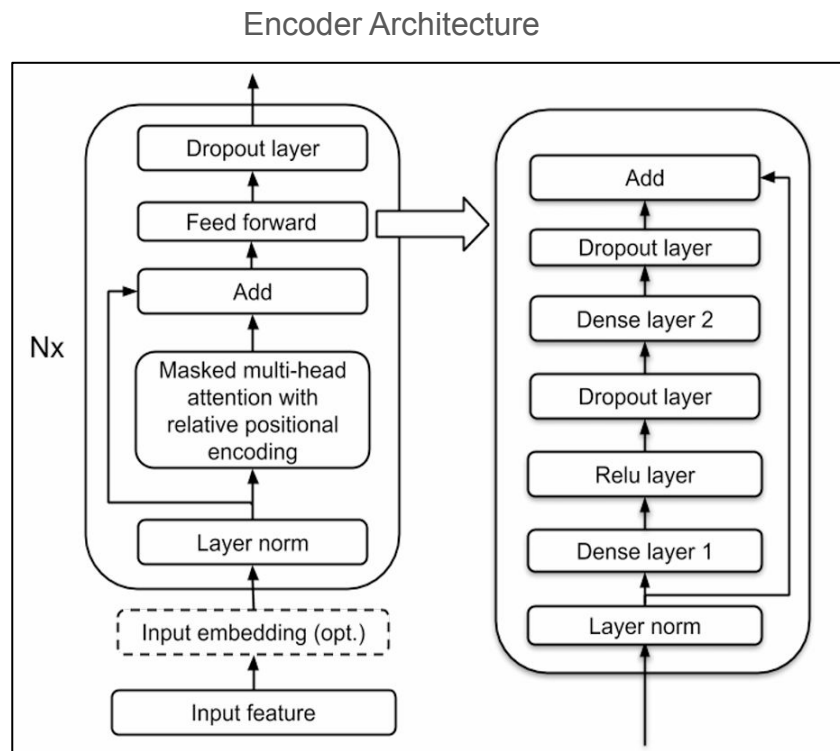
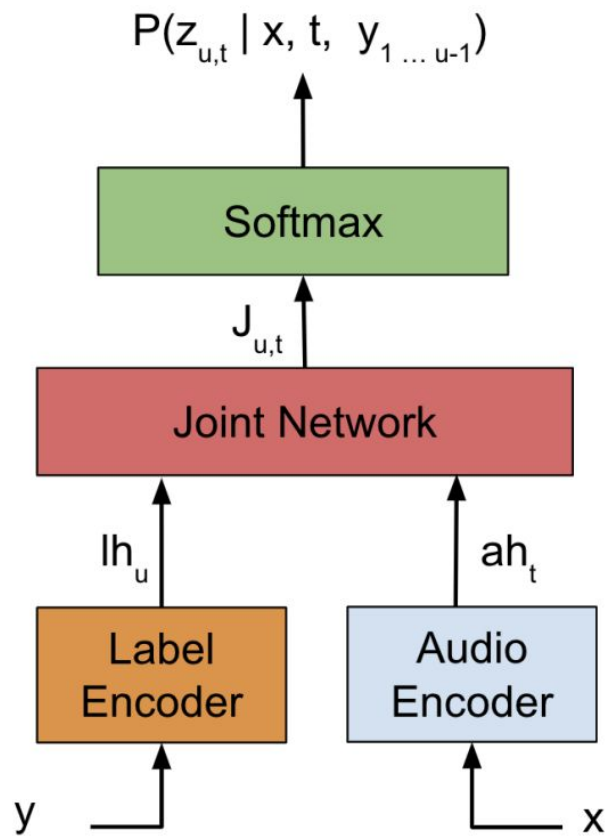


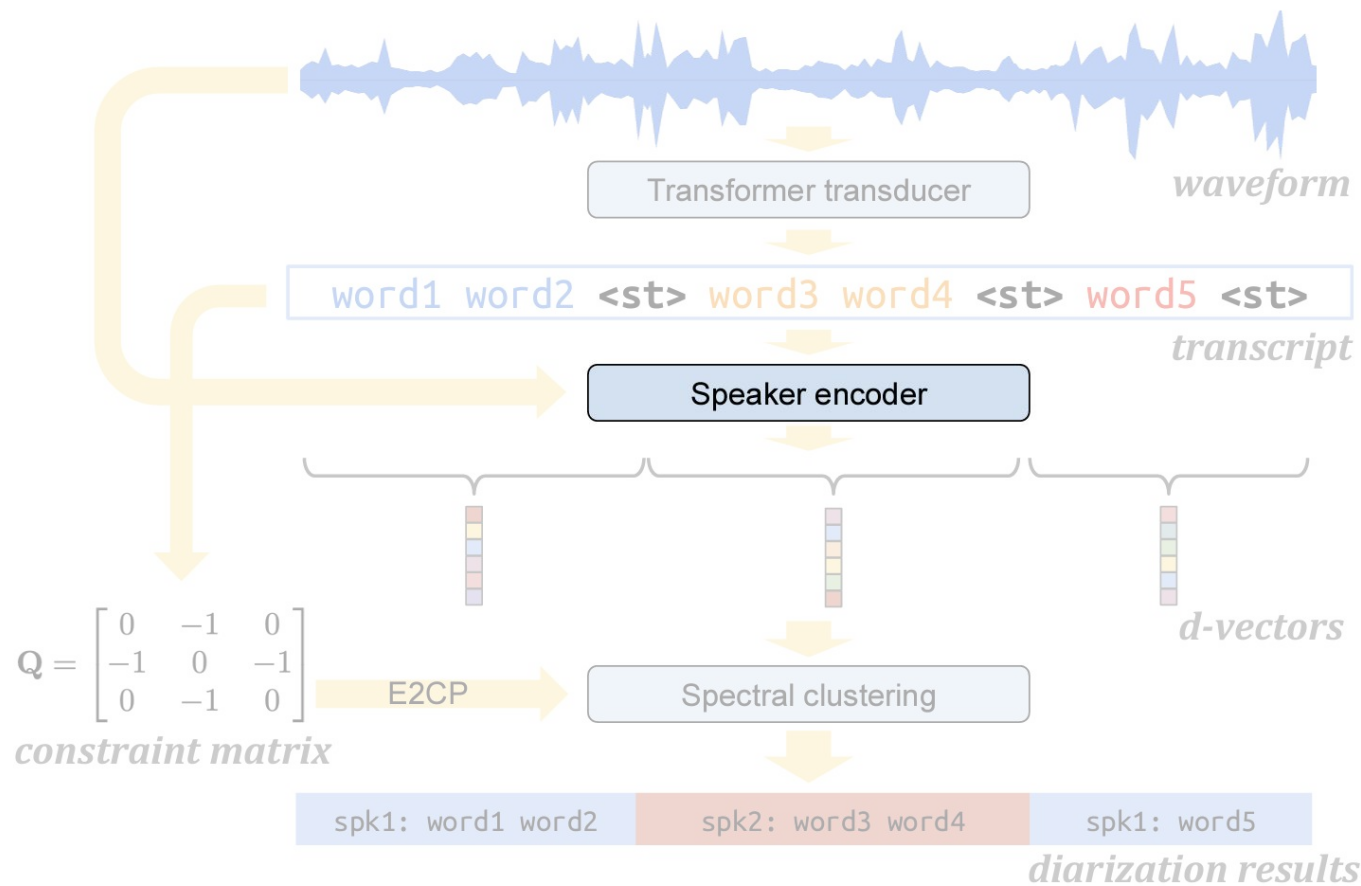




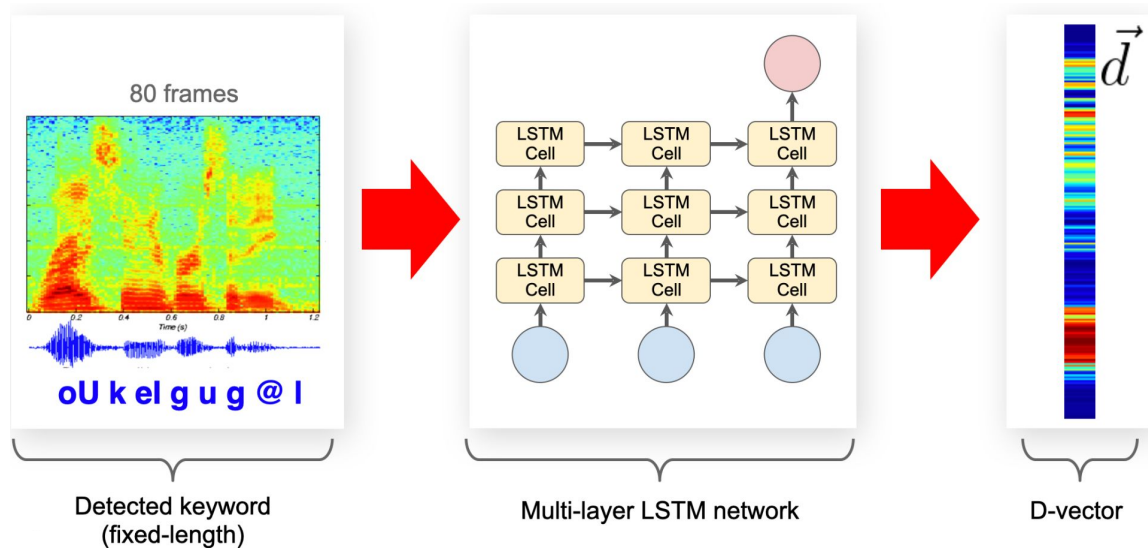


- The audio encoder has **15 layers** of Transformer blocks. Each block has 32 left context and **no right context**.
- The label encoder has **2 encoder layers**. Label Encoder states do not attend to Audio Encoder states, the **attention mechanism** here only operates within AudioEncoder or LabelEncoder, contrary to the standard practice for Transformer-based systems.
- For the joint network, we have a projection layer that projects the audio encoder output to 256-d.
- At the output of the joint network, it produces a distribution over **75 possible graphemes** with a softmax layer.

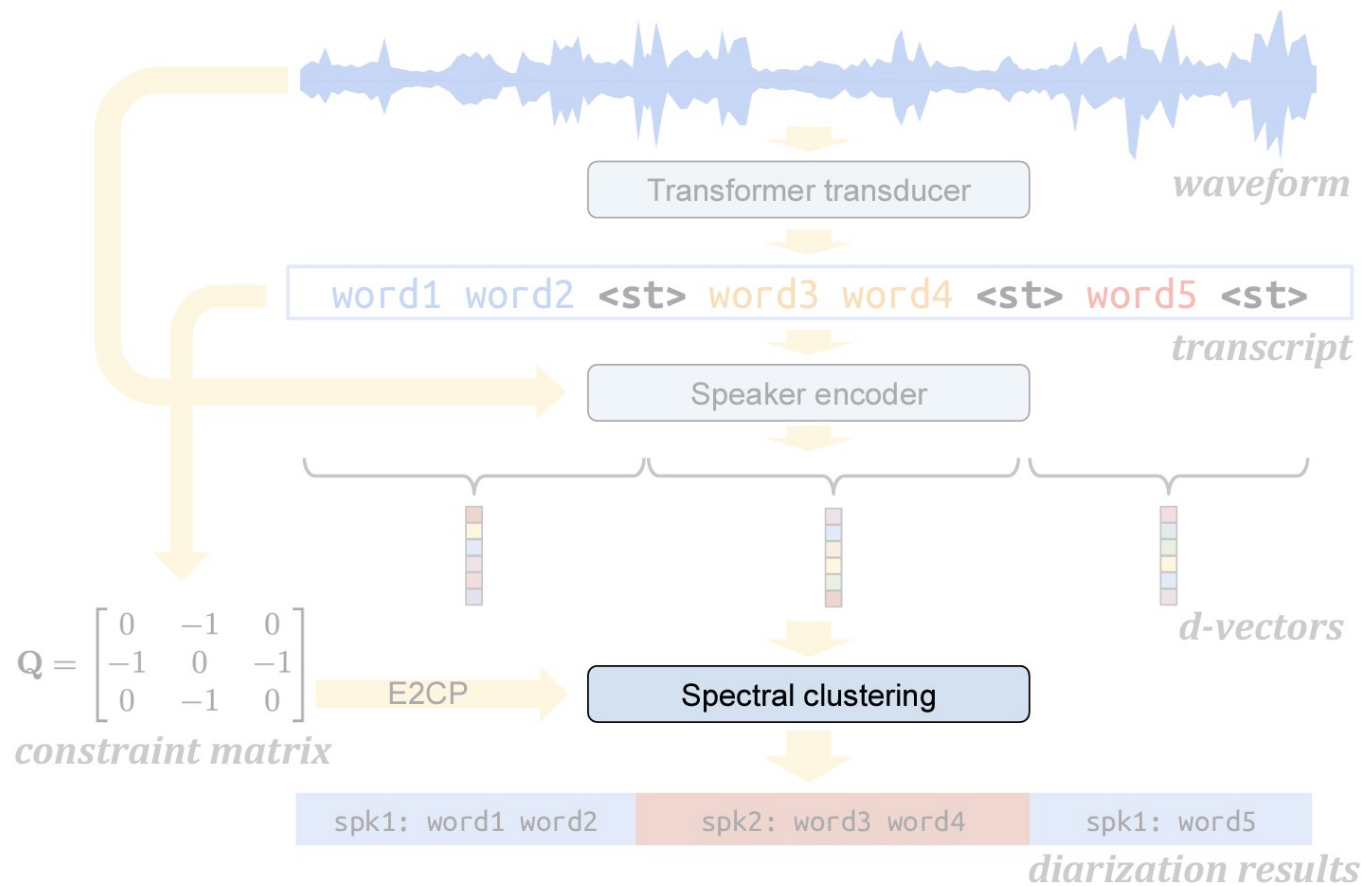




# Speaker Encoder - LSTM



- The speaker encoder model has 3 LSTM layers each with 768 nodes and a projection size of 256. The output of the last LSTM layer is then linearly transformed to the final 256-dimension d-vector.
- At inference time, we use the detected speaker turns as signals to reset the LSTM states of the speaker encoder, such that it does not carry information across different turns.
- 75% of this turn to represent this speaker turn, segment longer than 6 seconds.



# Spectral Clustering

Affinity Matrix

$$\mathbf{A} \in \mathbb{R}^{N \times N} \mid a_{ij} = \frac{1}{2} (1 + \cos(\mathbf{x}_i, \mathbf{x}_j))$$

$$\bar{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$$

# Spectral Clustering

## Constraint Matrix

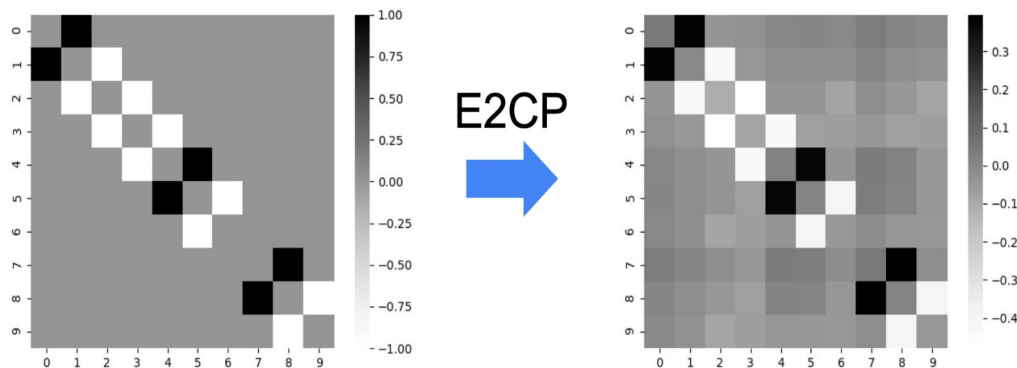
$$Q_{ij} = \begin{cases} -1, & \text{If } (i, j) \in \text{CL and } c(<\text{st}>) > \sigma; \\ +1, & \text{If } (i, j) \in \text{ML and } \text{len}(\text{st}) > 6 \text{ sec}; \\ 0, & \text{Otherwise.} \end{cases}$$

- If there is a speaker turn between segment  $i$  and  $i + 1$ , and the confidence of the  $<\text{st}>$  token  $c(<\text{st}>)$  is larger than a threshold  $\sigma$ , we define this pair as a “Cannot-link” (CL)
- If there is no speaker turn between two segments, we define it as a “Must-Link” (ML).
- $Q_{ij} = 0$  if  $i, j$  are not neighboring segments.

# Spectral Clustering

## E2CP

$$\mathbf{Q}^* = (1 - \alpha)^2 (\mathbf{I} - \alpha \bar{\mathbf{A}})^{-1} \mathbf{Z} (\mathbf{I} - \alpha \bar{\mathbf{A}})^{-1}.$$



Calculate constraint matrix  $\mathbf{Z}$  using the <st> tokens

Propagate constraints to non neighbouring segments

Speaker A = Speaker B,  
 $A \neq C \Rightarrow B \neq C$



# Results

# Results

Diarization eval sets:

- “Outbound” telephone speech: 450 conversations, each with 2 speakers
- “Inbound” telephone speech : 250 conversations, each with 2~10 speakers
- Callhome American English (eval subset)

**Table 2.** Confusion (%), total DER (%) and GFLOP/s on three datasets for different embeddings and methods.

System	Method	Inbound		Outbound		Callhome Eval		GFLOP/s at 10min	GFLOP/s at 1h
		Conf.	DER	Conf.	DER	Conf.	DER		
Dense d-vector	Dense	17.98	22.13	10.66	15.97	5.39	7.76	0.85	36.54
	Dense + Auto-tune	14.09	18.24	9.56	14.88	5.42	7.79	4.76	361.37
Turn-to-diarize	Turn	17.87	19.43	8.41	10.34	8.23	10.08	1.00	1.18
	Turn + E2CP	17.21	18.77	7.94	9.86	3.56	5.41	1.00	1.18
	Turn + Auto-tune	13.83	15.39	7.01	8.93	5.11	6.95	1.02	2.81
	Turn + E2CP + Auto-tune	13.66	15.22	6.86	8.78	3.49	5.33	1.02	2.81

# Summary - Key Insights

- Easier to annotate Data
  - Easy to modify current transcription workflows
  - Less time, less cost
- Clustering becomes very cheap
  - We only compute D vectors for each speaker turn
  - Apply constraints using E2CP before clustering
- Easy to determine when to diarize
  - If no <st> detected, no need for the diarization system
  - Else, calculate D vectors and cluster

# Limitations

1. When a new speaker comes into the conversation, the D vector from the speaker turn is compared against all D vectors from all the previous speaker turns
  - a. This is redundant since Spectral Clustering is the bottleneck
  - b. Instead maintain a set of all identified speakers by averaging the D vectors for each speaker
2. Need multiple language models for speakers in different languages
  - a. Need to detect language switch
3. Training Transducers is computationally intensive
  - a. Most research done by Google, Meta
  - b. No issues during inference

# References

1. Xia, W. et al. (2022). Turn To Diarize  
<https://arxiv.org/pdf/2109.11641.pdf>
2. Zhang, Q. et al. (2020). Transformer Transducer  
<https://arxiv.org/pdf/2002.02562.pdf>
3. Wan, L. et al. (2020). Generalized End To End Loss For Speaker Verification  
<https://arxiv.org/pdf/1710.10467.pdf>
4. Wang, Q. (2022). [ICASSP 2022] Turn-to-Diarize  
<https://www.youtube.com/watch?v=U79Aw1ky7ag>
5. Fleshman, L. (2019). Spectral Clustering  
<https://towardsdatascience.com/spectral-clustering-aba2640c0d5b>
6. Wang, Q. (2022). Spectral Cluster  
<https://github.com/wq2012/SpectralCluster>
7. Lu, Z. (2011). Exhaustive and Efficient Constraint Propagation  
<https://arxiv.org/pdf/1109.4684.pdf>
8. Lugosch, L. (2020). Sequence-to-sequence learning with Transducers  
<https://lorenlugosch.github.io/posts/2020/11/transducer/>
9. Graves, A. (2012). Sequence Transduction with Recurrent Neural Networks  
<https://arxiv.org/pdf/1211.3711.pdf>
10. Jain, M. (2021). RNN-T Based ASR Systems  
[https://www.cc.gatech.edu/classes/AY2021/cs7643\\_spring/assets/L24\\_rnn\\_t\\_asr\\_tutorial\\_gt.pdf](https://www.cc.gatech.edu/classes/AY2021/cs7643_spring/assets/L24_rnn_t_asr_tutorial_gt.pdf)

# Appendix

[Slide 26: Log Mel Filterbank Energy Features](#)

[Slide 27: Transformer](#)

[Slide 28: Attention](#)

[Slide 29: T-T Training: Dataset & Loss](#)

[Slide 30: Speech Encoder Training: Loss, Optimizer, Dataset](#)

[Slide 31: Spectral Clustering](#)

[Slide 32: Diarization Error Rate](#)

# Log Mel Filterbank Energy Features

- Mel Scale

The **mel scale** (after the word *melody*)<sup>[1]</sup> is a perceptual scale of *pitches* judged by listeners to be equal in distance from one another.

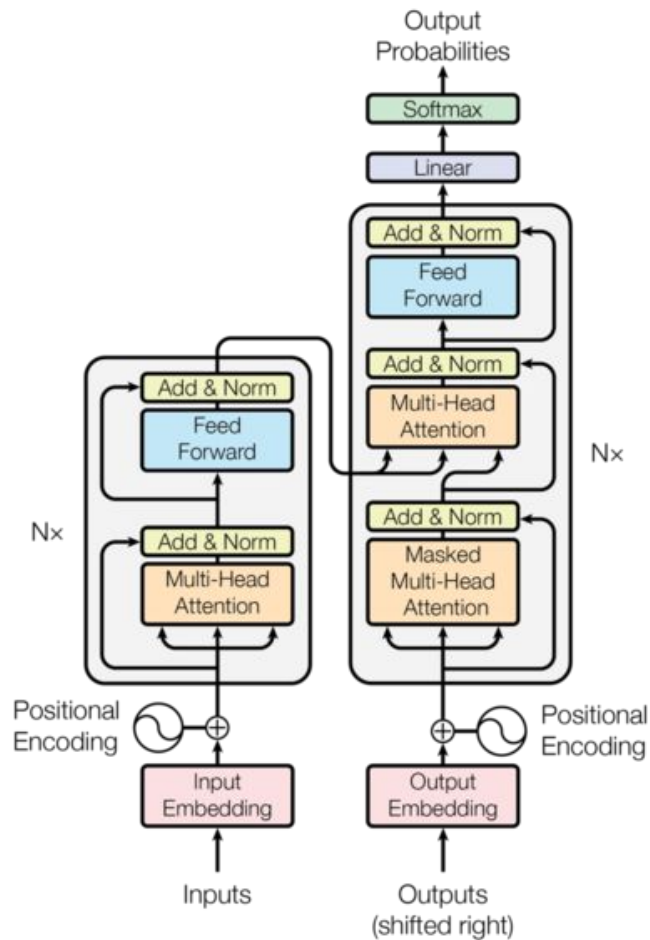
The Mel-scale aims to mimic the non-linear human ear perception of sound, by being more discriminative at lower frequencies and less discriminative at higher frequencies

- Fourier Transform

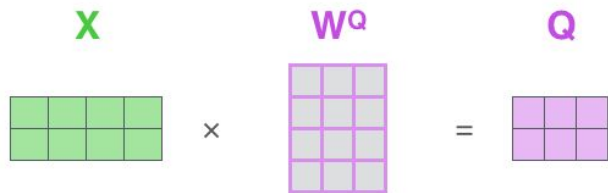
A **Fourier transform (FT)** is a *mathematical transform* that decomposes *functions* depending on *space* or *time* into functions depending on *spatial frequency* or *temporal frequency*. An example application would be decomposing the *waveform* of a musical *chord* into terms of the *intensity* of its constituent *pitches*.



# Transformer

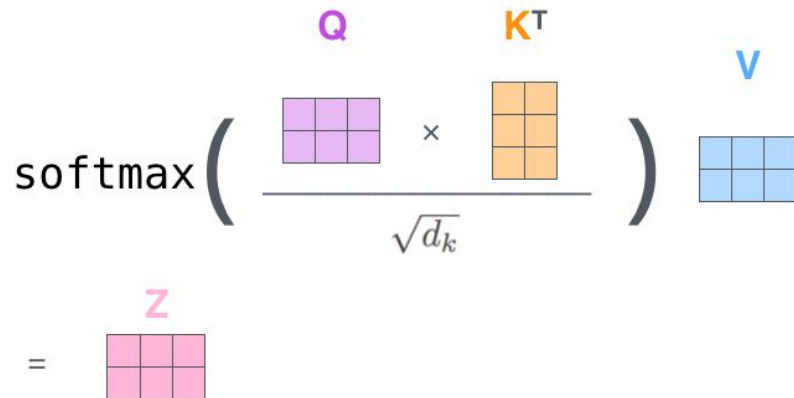


# Attention

$$\mathbf{X} \times \mathbf{W}^Q = \mathbf{Q}$$


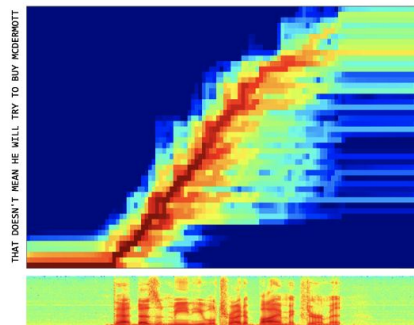
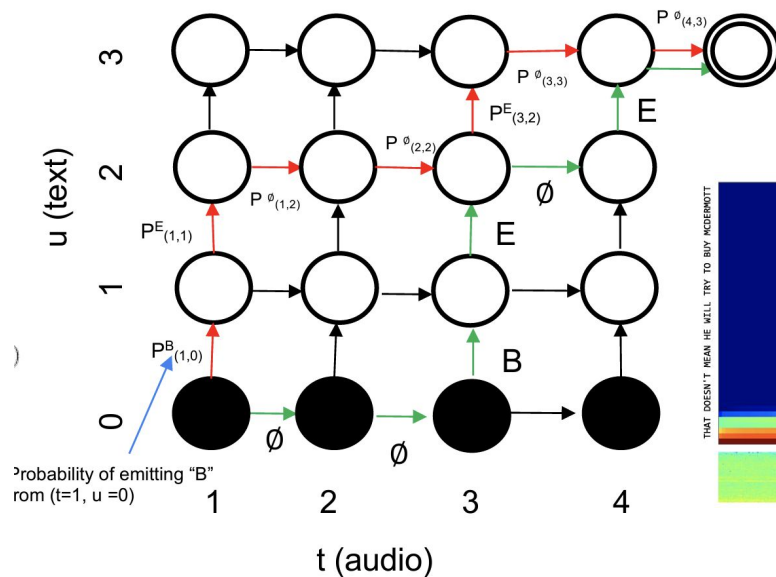
$$\mathbf{X} \times \mathbf{W}^K = \mathbf{K}$$


$$\mathbf{X} \times \mathbf{W}^V = \mathbf{V}$$


$$\text{softmax}\left(\frac{\mathbf{Q} \times \mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}$$
$$= \mathbf{Z}$$


# T-T Training: Dataset & Loss

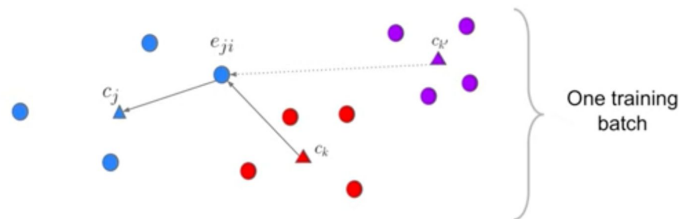
- Fisher, Callhome American English (training subset), ~7500 hours of YouTube videos (internal)
- RNN-T loss
  - Probability of alignment is multiplication of probabilities assigned along the path of alignment
  - Lattice contains all valid alignment paths(traversals). During training, we change (optimize) neural network parameters to maximize sum of probabilities of all alignment paths



# Speech Encoder Training: Loss, Optimizer, Dataset

- Vendor collected speech queries (37 locales), LibriVox, CN-Celeb, TIMIT, VCTK
- Generalized end-to-end (GE2E) loss

## Loss function - Generalized end-to-end loss



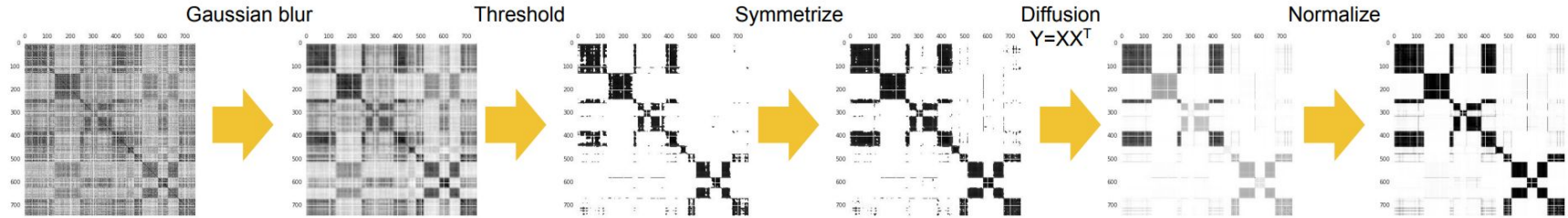
For each embedding  $e_{ji}$ , we only look at its distance to:

- True speaker centroid  $c_j$
- Closest false speaker centroid  $c_k \Rightarrow \text{max-margin principle, "support speaker"}$

# Spectral Clustering

- Order:
  - Affinity Matrix
  - Build constraint Matrix - same shape as A matrix
  - Propagate constraints using E2CP
  - Adjust Affinity matrix -  $\hat{A}$
  - Refine
    - Gaussian Blur
    - Row wise soft thresholding
    - Symmetrization
  - Laplacian
  - Eigen Decomp
  - Eigengap / spectral gap
    - The first nonzero eigenvalue is called the spectral gap. The spectral gap gives us some notion of the density of the graph. If this graph was densely connected (all pairs of the 10 nodes had an edge), then the spectral gap would be 10.
  - K means on renormalized spectral embeddings

# Spectral Clustering



- Calculate Laplacian
- Eigen Decomp
- Eigengap / Spectral Gap
  - The first nonzero eigenvalue, gives us some notion of the density of the graph
- K means on renormalized spectral embeddings

# Diarization Error Rate

Speaker error: percentage of scored time that a speaker ID is assigned to the wrong speaker.

False alarm speech: percentage of scored time that a hypothesized speaker is labelled as a non-speech in the reference.

Missed speech: percentage of scored time that a hypothesized non-speech segment corresponds to a reference speaker segment.

Overlap speaker: percentage of scored time that some of the multiple speakers in a segment do not get assigned to any speaker.

$$DER = E_{spkr} + E_{MISS} + E_{FA} + E_{ovl}$$