# Minimum Latency Training Strategies for Streaming Sequence-to-Sequence ASR

**Hirofumi Inaguma**[1,2], Yashesh Gaur[2], Liang Lu[2], Jinyu Li[2], Yifan Gong[2]

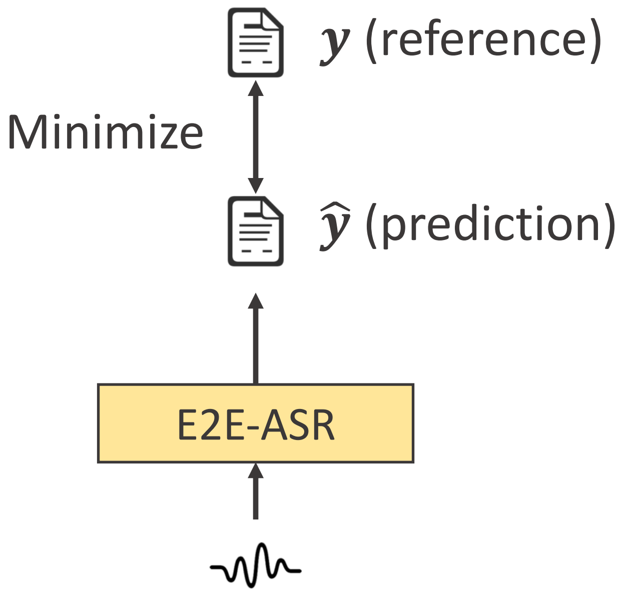
Kyoto Univiersity, Kyoto, Japan[1]
Microsoft Speech and Language Group, Redmond, WA, USA[2]

IEEE ICASSP 2020

Microsoft

# Background: End-to-end ASR

- Input sequence (speech): $\boldsymbol{x} = (x_1, \dots, x_T)$
- Output sequence (transcription): $\boldsymbol{y} = (y_1, \dots, y_L)$




**Time-synchronous** model ($|\boldsymbol{x}| = |\hat{\boldsymbol{y}}|$)

- Connectionist temporal classification (CTC) [Graves et al., 2006]
- RNN-Transducer (RNN-T) [Graves et al., 2013]
- Recurrent neural aligner (RNA) [Sak et al., 2017]

Low accuracy
Streaming: easy

**Label-synchronous** model ($|\boldsymbol{x}| \neq |\hat{\boldsymbol{y}}|$)

- Attention-based sequence-to-sequence (S2S) [Bahdanau et al., 2016]
- Transformer [Vaswani et al., 2017]

High accuracy
Streaming: difficult

# Streaming attention-based S2S ASR

Microsoft

## Neural Transducer [Jailty et al.,2015]

- Perform attention mechanism for a fixed size of block

## Hard monotonic attention [Raffel et al., 2017]

- Learn to detect token boundaries via stochastic binary decision
- Extension: Monotonic chunkwise attention (MoChA) [Chiu et al., 2018]

## Triggered attention [Moritz et al., 2018]

- Perform global attention over encoder memories trun

- Good results
- Efficient training

## Adaptive computation steps (ACS) [Li et al., 2018]

- Learn how many tokens to generate with encoder outputs

## Continuous Integrate-and-Fire (CIF) [Dong et al,, 2019]

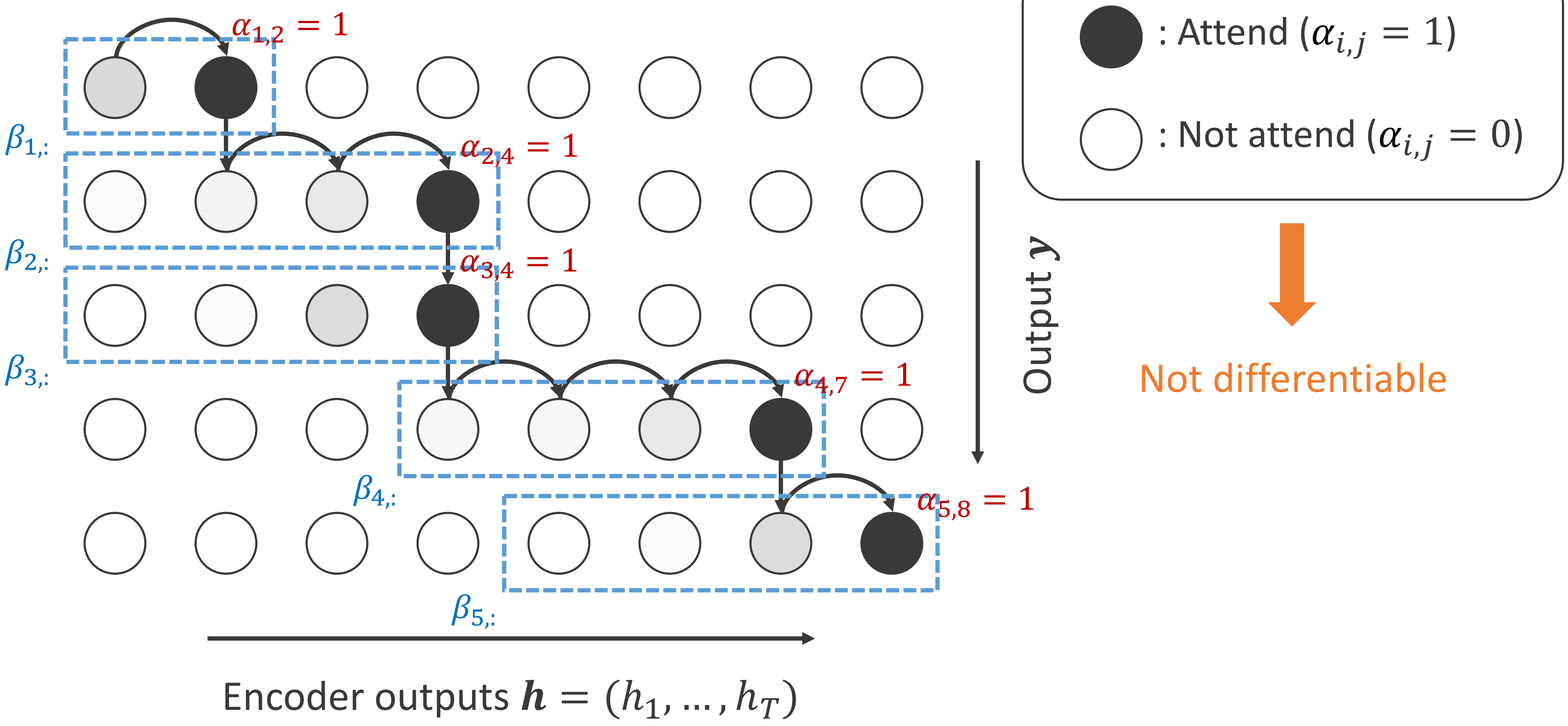
- Fine-grained version of ACS

And more…

- Windowing approaches
- Reinforcement learning

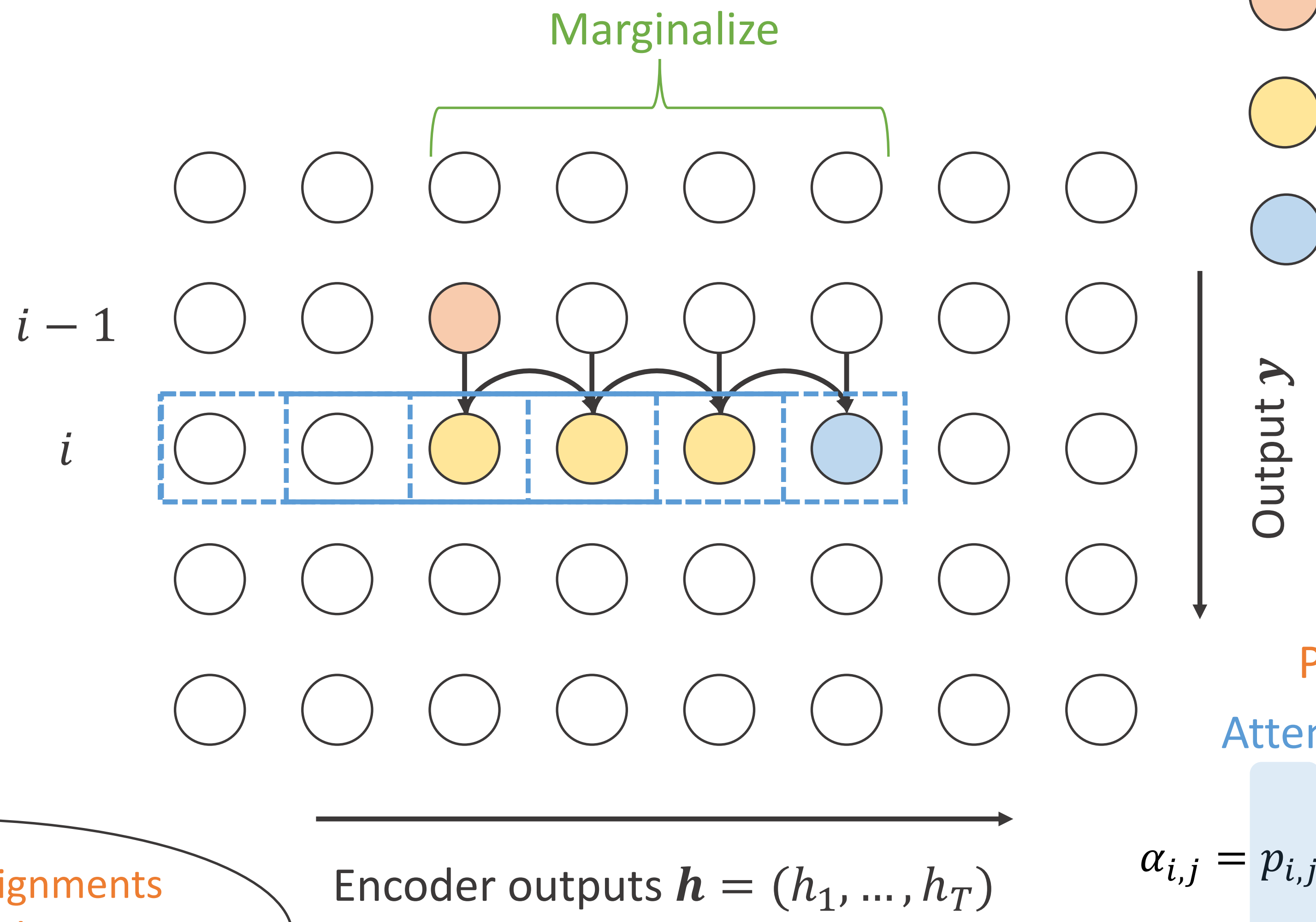
# MoChA (test time)

e.g., $w = 4$ (chunk size: 4)




1. ***Monotonic attention**: whether to attend or not*
2. ***Chunkwise attention**: soft attention over a small window*

# MoChA (training time)




: Attend at $(i-1)$-th step

: Not attend

: Attend at $i$-th step

Previous attention

Attend

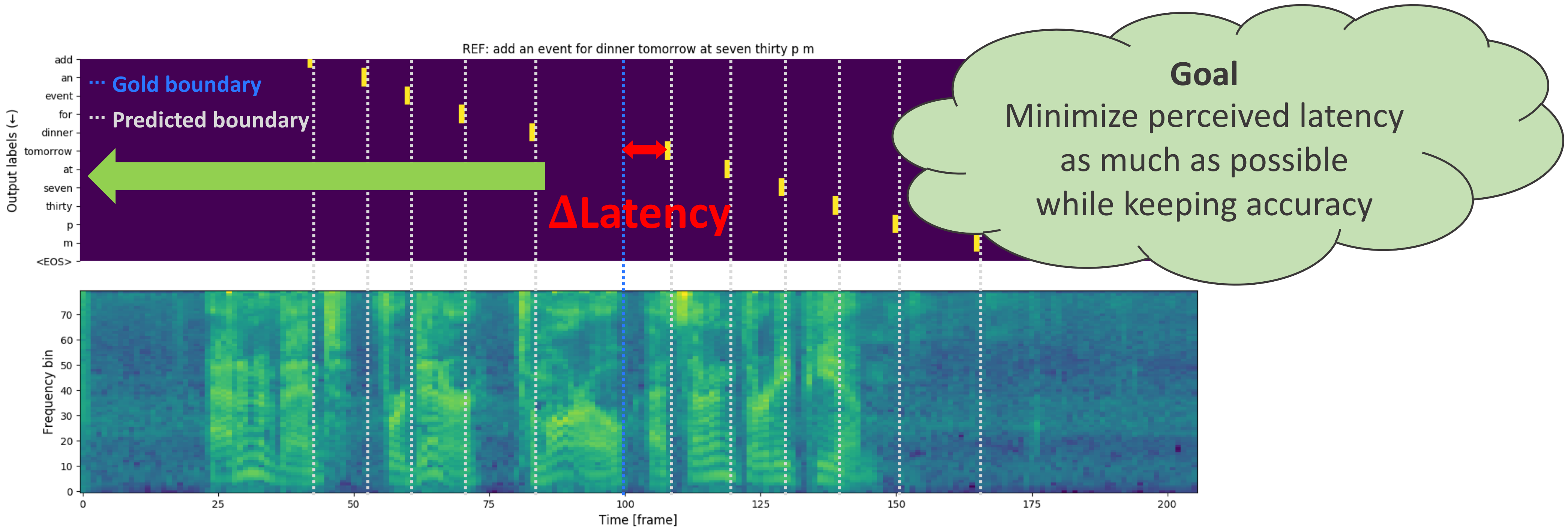
Not attend

$$\alpha_{i,j} = p_{i,j} \sum_{k-1}^{j} \left( \alpha_{i-1,k} \prod_{l=k}^{j-1} (1 - p_{i,l}) \right)$$

$$= (1 - p_{i,j-1}) \frac{\alpha_{i,j-1}}{p_{i,j-1}} + \alpha_{i-1,j}$$

Use expected alignments
during training
for backpropagation

# Delayed token generation problem




- Decision boundaries (yellow dots) are delayed from the actual acoustic boundary
  1. Unidirectional encoder (lacking the future information)
  2. Sequence-level criterion (utilizing as many future frames as possible to maximize the log-likelihood)
- This leads to increasing user perceived latency
  - ➢Similar behaviors have been reported in CTC [sak et al., 2015] and RNN-T [Li et al., 2019]

# Evaluation metric: latency

Microsoft

- Definition: difference between time-index of a predicted boundary and that of the gold boundary

***Corpus-level*** **latency** (averaged per token)

$$\Delta_{\text{corpus}} = \frac{1}{\sum_{k=1}^{N} |\boldsymbol{y}^k|} \sum_{k=1}^{N} \sum_{i=1}^{|\boldsymbol{y}^k|} (\widehat{b_i^k} - b_i^k)$$

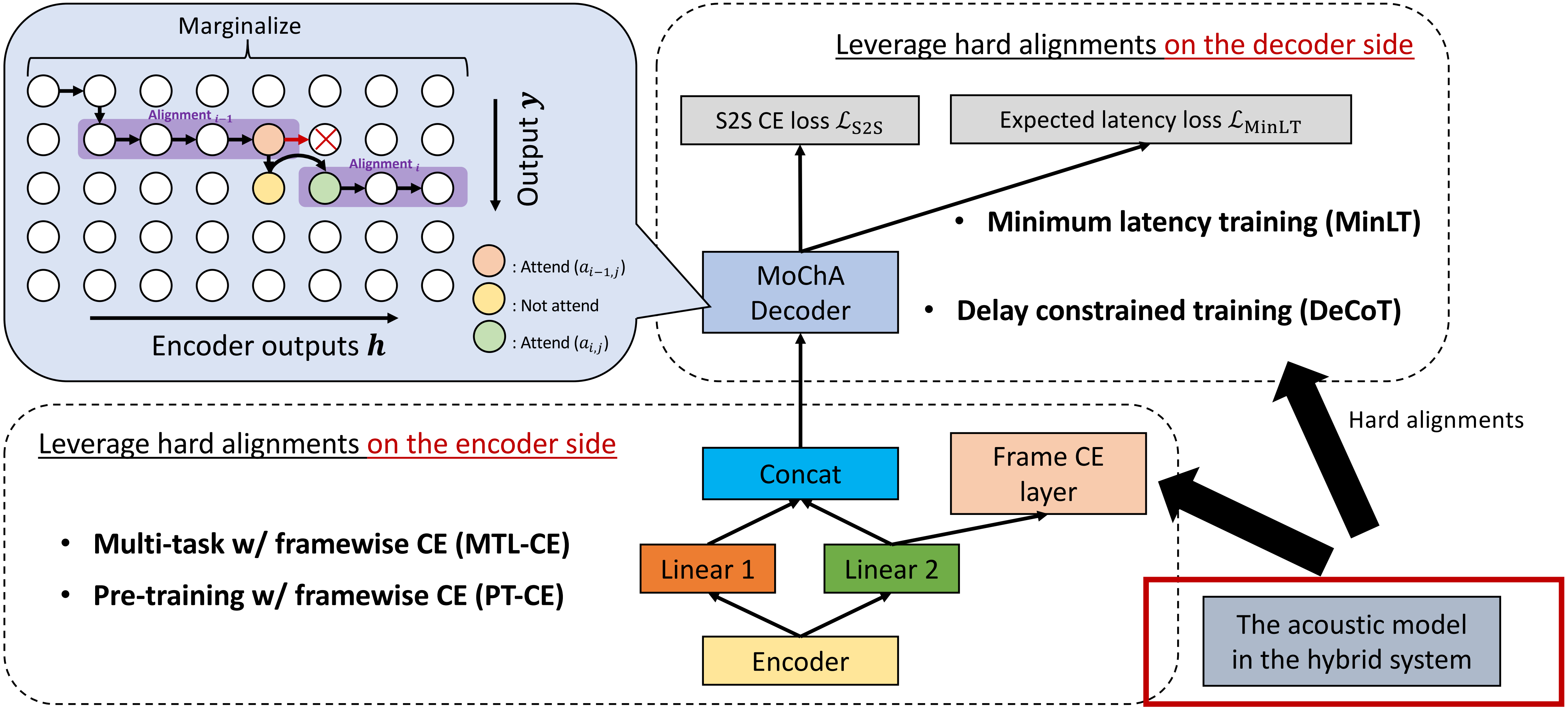
***Utterance-level*** **latency** (averaged per utterance)

$$\Delta_{\text{utterance}} = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{|\boldsymbol{y}^k|} \sum_{i=1}^{|\boldsymbol{y}^k|} (\widehat{b_i^k} - b_i^k)$$

- Report (1) average, (2) median, (3) 90-th, and (4) 99-th percentile
- Teacher-forcing when calculating latency to match the sequence lengths

# Proposed methods

**Where should we apply alignment information in the model?**




Leveraging hard alignments extracted from the hybrid system

# 1. Multi-task learning w/ framewise CE (MTL-CE)

Microsoft

Train both branches
from scratch

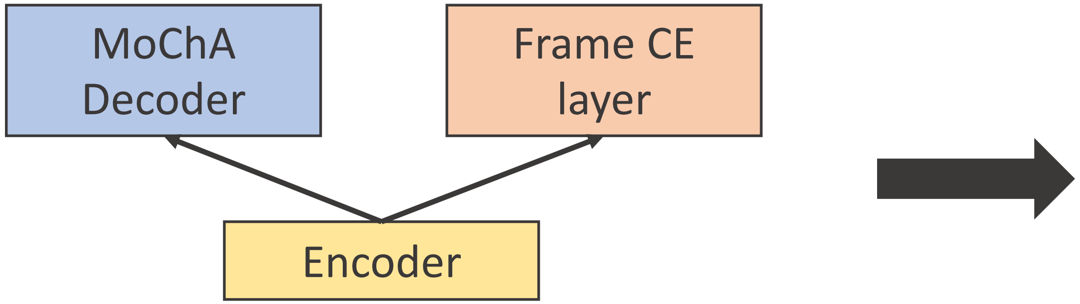
## ◆Objective function

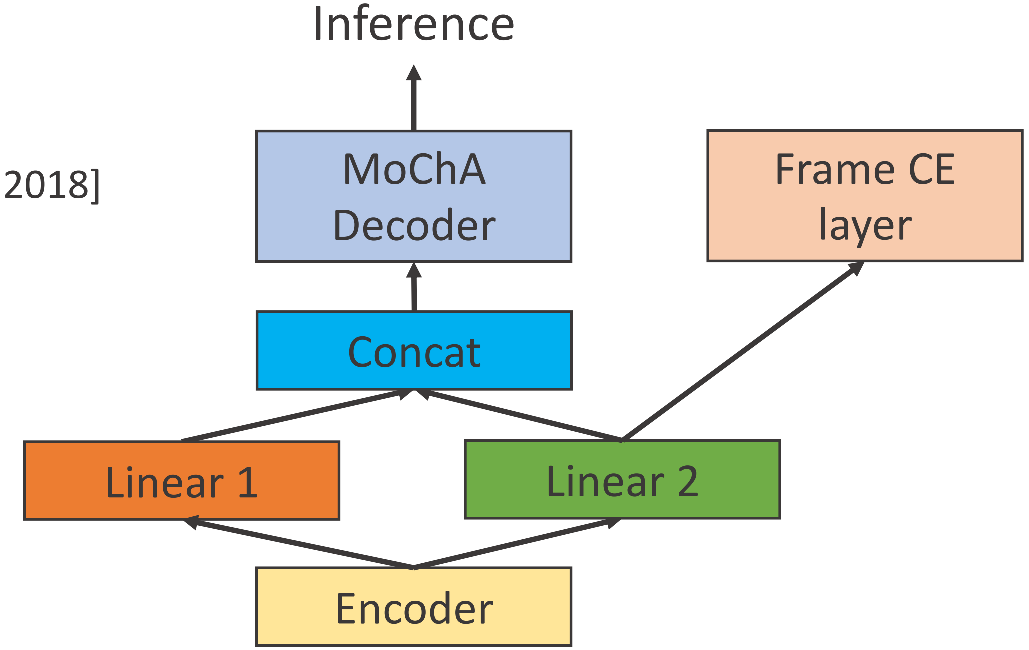
$$\mathcal{L}_{\text{total}} = (1-\lambda_{\text{CE}})\underbrace{\mathcal{L}_{\text{S2S}}(\boldsymbol{y}|\boldsymbol{x})}_{\text{MoChA}} + \lambda_{\text{CE}}\underbrace{\mathcal{L}_{\text{CE}}(\boldsymbol{A}|\boldsymbol{x})}_{\text{Frame CE}} \quad (0 \leq \lambda_{\text{CE}} \leq 1)$$

- Motivation: align encoder outputs to the true acoustic location

## ◆Insert linear bottleneck layers

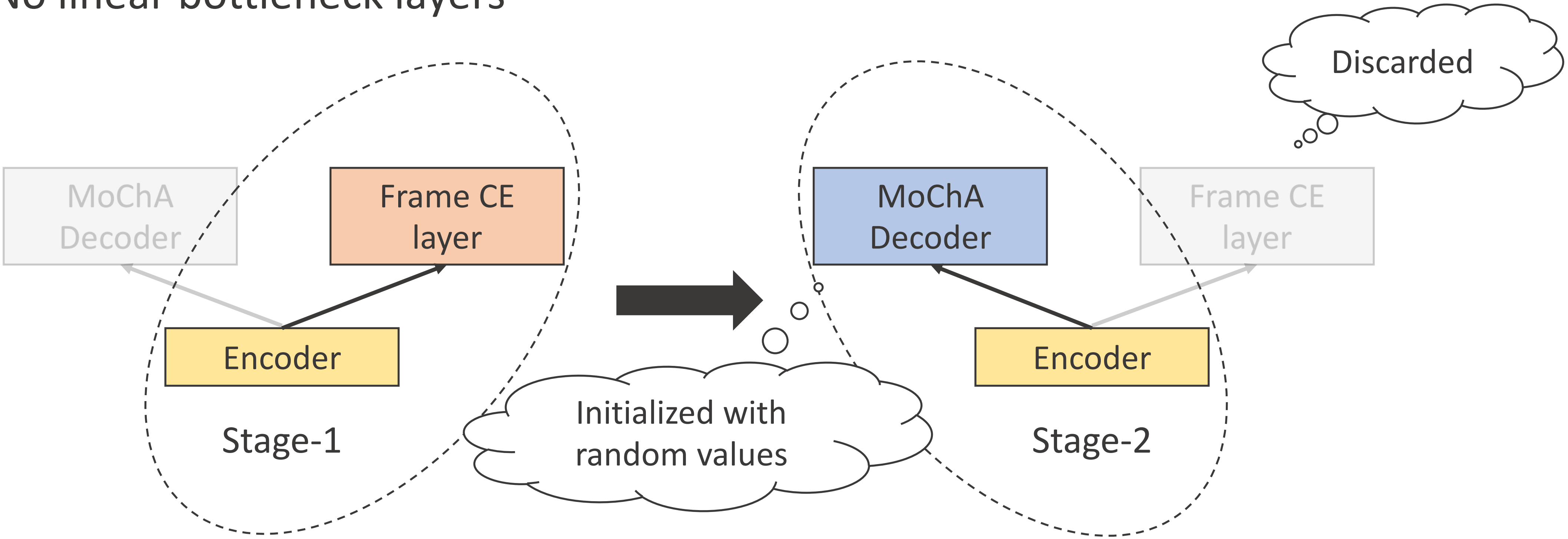
- Inspired by the CTC acoustic model [Yu et al., 2018]



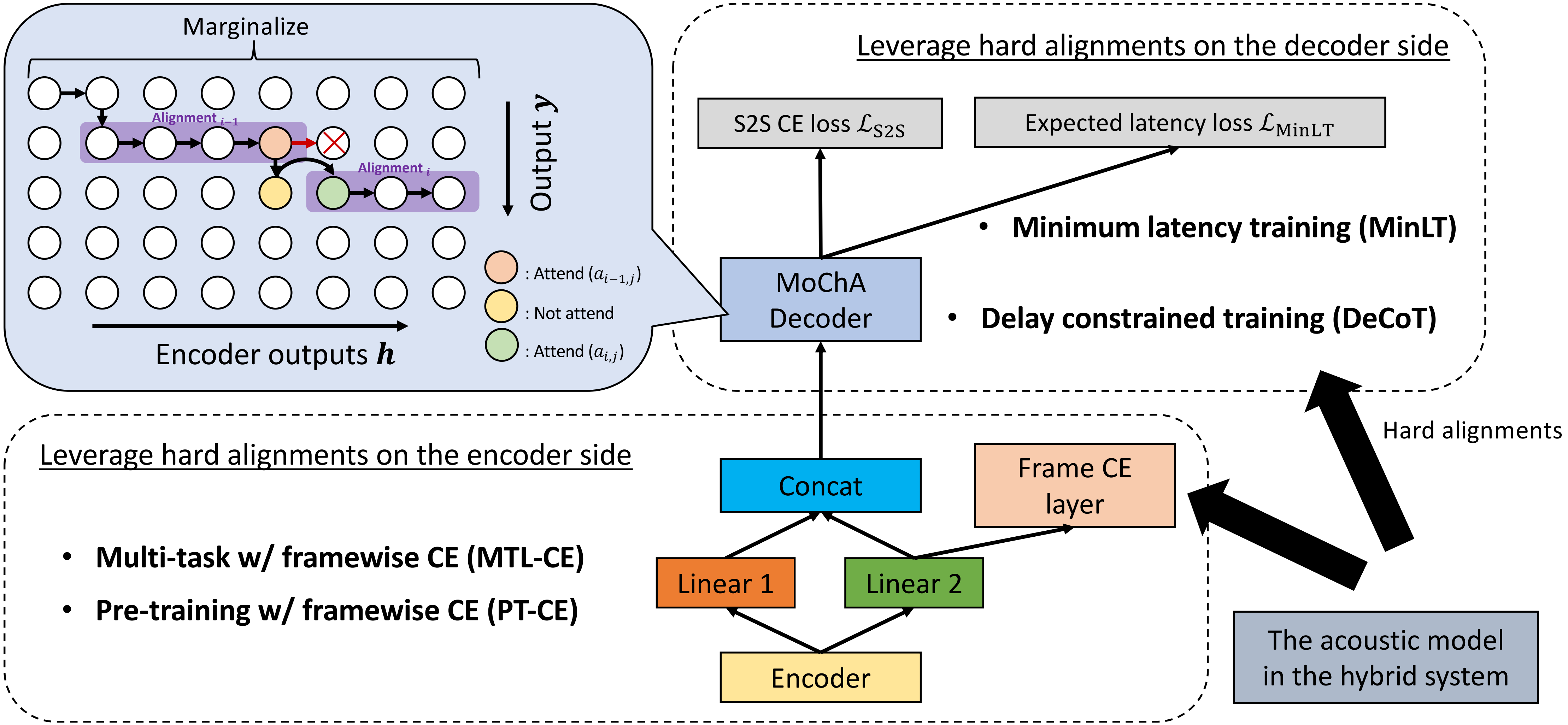



# 2. Pre-training with framewise CE (PT-CE)

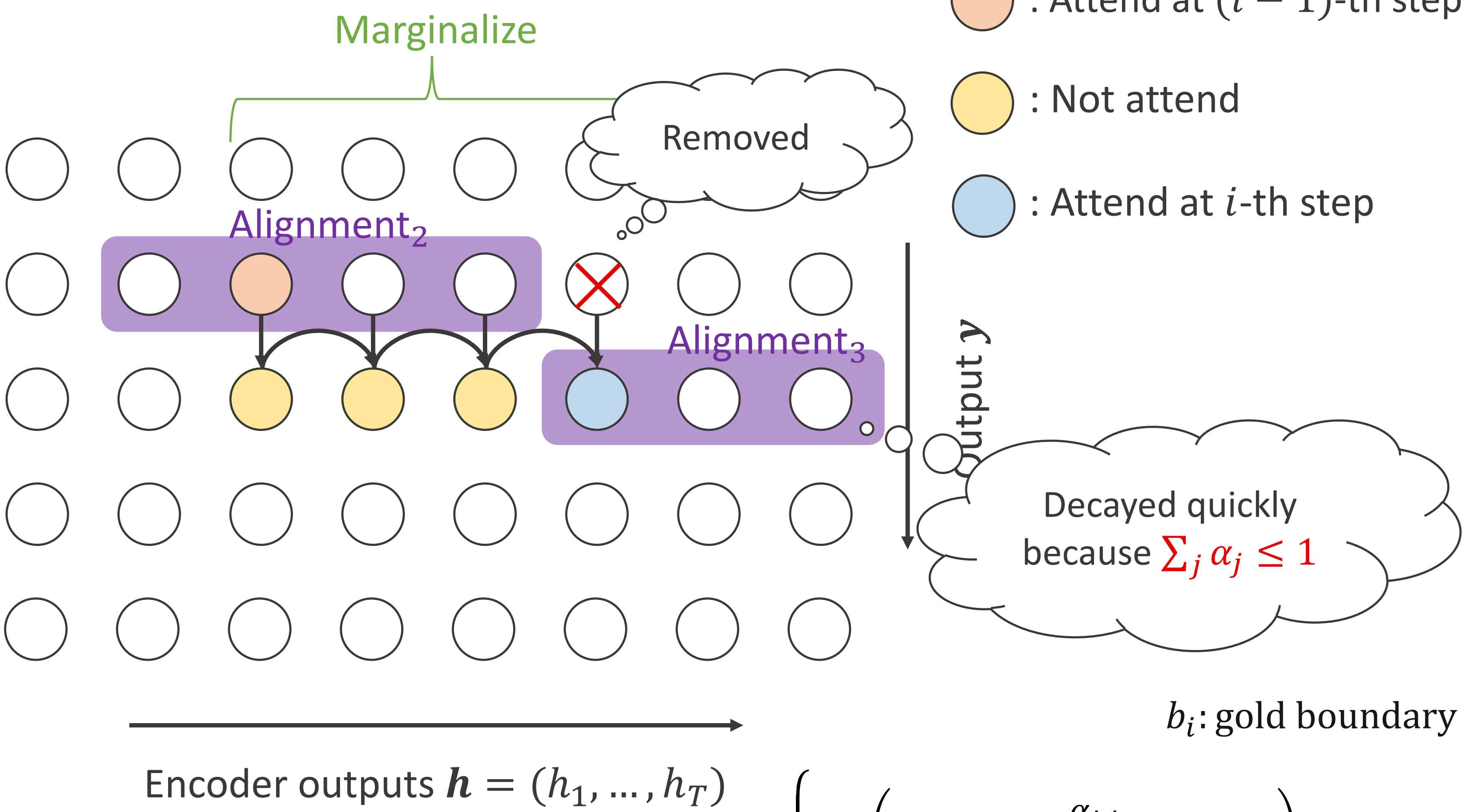
- **2-staged training**
  - Motivation
    - Start training from well-aligned encoder representations
    - Do not have to tune the CE weight $\lambda_{\mathrm{CE}}$
  - No linear bottleneck layers



# Proposed methods



# 3. Delay constrained training (DeCoT)




$b_i$: gold boundary

$$\alpha_{i,j} = \begin{cases} p_{i,j}\left((1-p_{i,j-1})\dfrac{\alpha_{i,j-1}}{p_{i,j-1}} + \alpha_{i-1,j}\right) & (j \leq b_i + \delta) \\ 0 & (\text{otherwise}) \end{cases}$$

Remove inappropriate paths whose boundaries surpass the actual acoustic boundary more than fixed acceptable latency $\delta$ [frames]

# 3. Delay constrained training (DeCoT)

Microsoft

## Regularization with quantity loss

- Add a regularization term to keep $\sum_j \alpha_j = 1$
- Originally proposed in CIF [Dong et al., 2019] with a different motivation

$L$: the number of tokens in the reference

$$\mathcal{L}_{\mathrm{QUA}} = |L - \sum_{i=1}^{L} \sum_{j=1}^{T} \alpha_{i,j}|$$

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{S2S}} + \lambda_{\mathrm{QUA}} \mathcal{L}_{\mathrm{QUA}} \quad (\lambda_{\mathrm{QUA}} \geq 0)$$

# 4. Minimum latency training (MinLT)

Microsoft

## ◆Objective function

- Directly minimize the expected latency $\mathcal{L}_{\mathrm{MinLT}}$ by utilizing hard alignments $\boldsymbol{A}$

Expected boundary

$$\mathcal{L}_{\mathrm{MinLT}} = \frac{1}{|y|} \sum_{i=1}^{L} \left| \sum_{j=1}^{T} j\alpha_{i,j} - b_i \right| \quad (b_i\text{: refernce boundary for } i\text{-th token})$$

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{S2S}} + \lambda_{\mathrm{MinLT}} \mathcal{L}_{\mathrm{MinLT}} \quad (\lambda_{\mathrm{MinLT}} \geq 0)$$

- Motivation: reduce latency flexibly
  - ➢DeCoT assumes <u>the fixed latency for each token</u>

## ◆Related work

- Latency loss has been investigated in simultaneous NMT [Arivazhagan et al., 2019]
- Non-silence frames are <u>not distributed uniformly</u> over the input speech in ASR

# Experimental condition

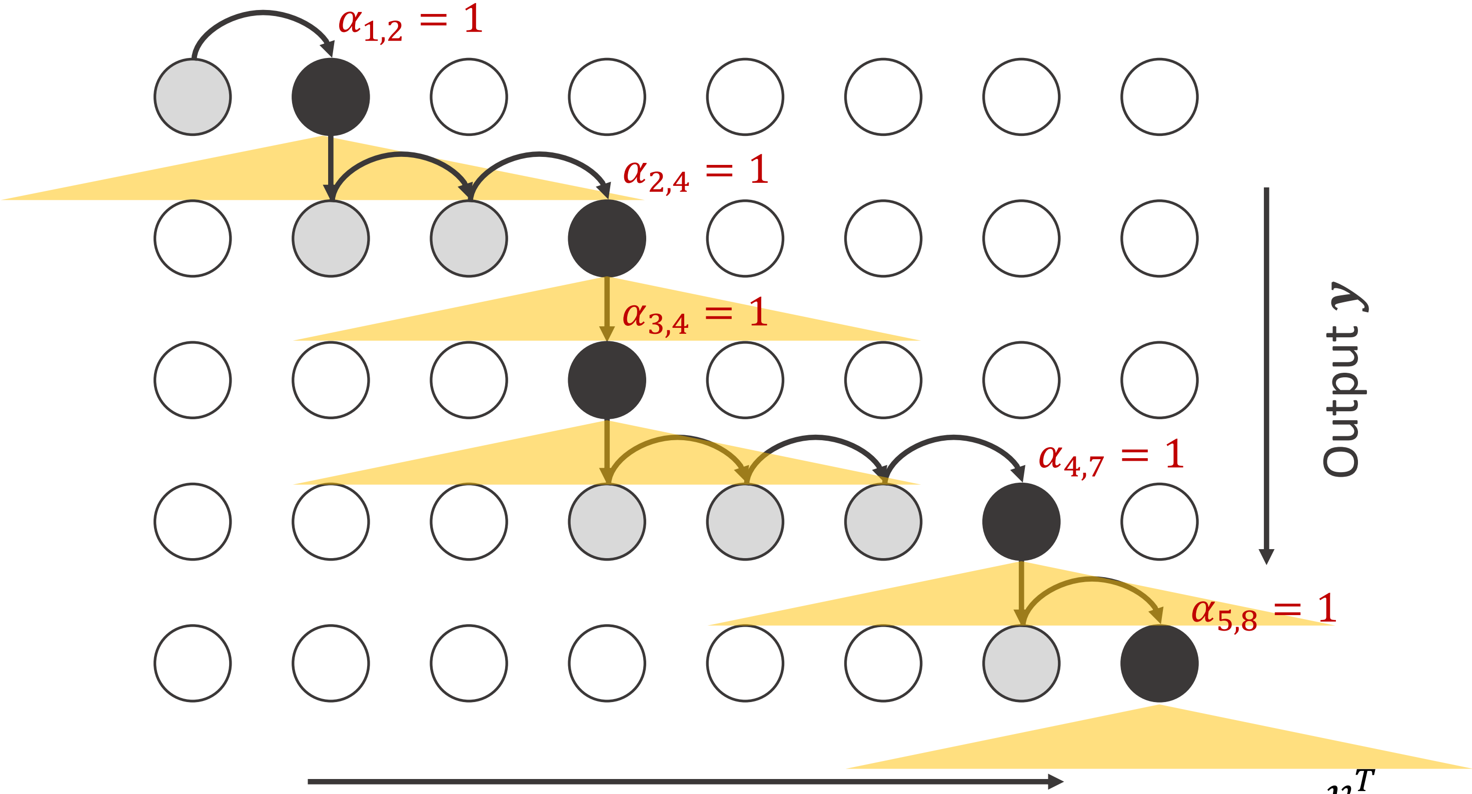
Microsoft

| | |
|---|---|
| Data | Train: Cortana voice assistant (3.4k hours)<br>Validation: Sampled disjoint 4k utterances form the training set<br>Test: 5.6k utterances |
| Feature | 80-dim log-mel fbank (3 frame stacked, 30ms per frame) |
| Output unit | Mixed units (34k vocabulary) |
| Architecture | Offline: 512-dim (per direction) 6-layer BiGRU encoder<br>Streaming: 1024-dim 6-layer GRU encoder<br>Decoder: 512-dim 2-layer GRU |
| Optimization | Adam |
| Decoding | Beam width: 8, no LM |

- Word-level alignments: $A = (a_1, \dots, a_T)$ ($\{a_j\}_{j=1,\dots,T}$: one-hot vector)
  - ➢Divide duration based on the ratio of the character length of each subword
- Start DeCoT and MinLT from the baseline MoChA (warm start training)

# Enhance monotonic attention with 1D convolution

e.g., $k = 5$ (lookahead: 2, 60ms)




Encoder outputs $\boldsymbol{h} = (h_1, \dots, h_T)$

$$e_{i,j}^{\mathrm{mono}} = g \frac{v^T}{\| v \|} \mathrm{ReLU}\left(W_{\mathrm{h}} h'_j + W_{\mathrm{s}} s_i + b\right) + r$$

$$h'_j = W_{\mathrm{c}} * h_j \ (W_{\mathrm{c}} \in \mathbb{R}^{d \times d \times k})$$

$k$: kernel size, $d$: unit size

## ◆Motivation

- Leveraging the surrounding frames for robust binary decision

# Results: Baseline

Microsoft

| Model | | WER [%] |
|---|---|---|
| Offline | BiGRU global attention | 7.01 |
| | UniGRU global attention | 8.44 |
| | BiGRU MoChA (chunk: 4) | 8.09 |
| Streaming | UniGRU MoChA (chunk:4) | 10.37 |
| | + 1D-convolution (baseline) | **9.93** |

4.24% WERR

- Huge gaps between (1) bidirectional <-> unidirectional

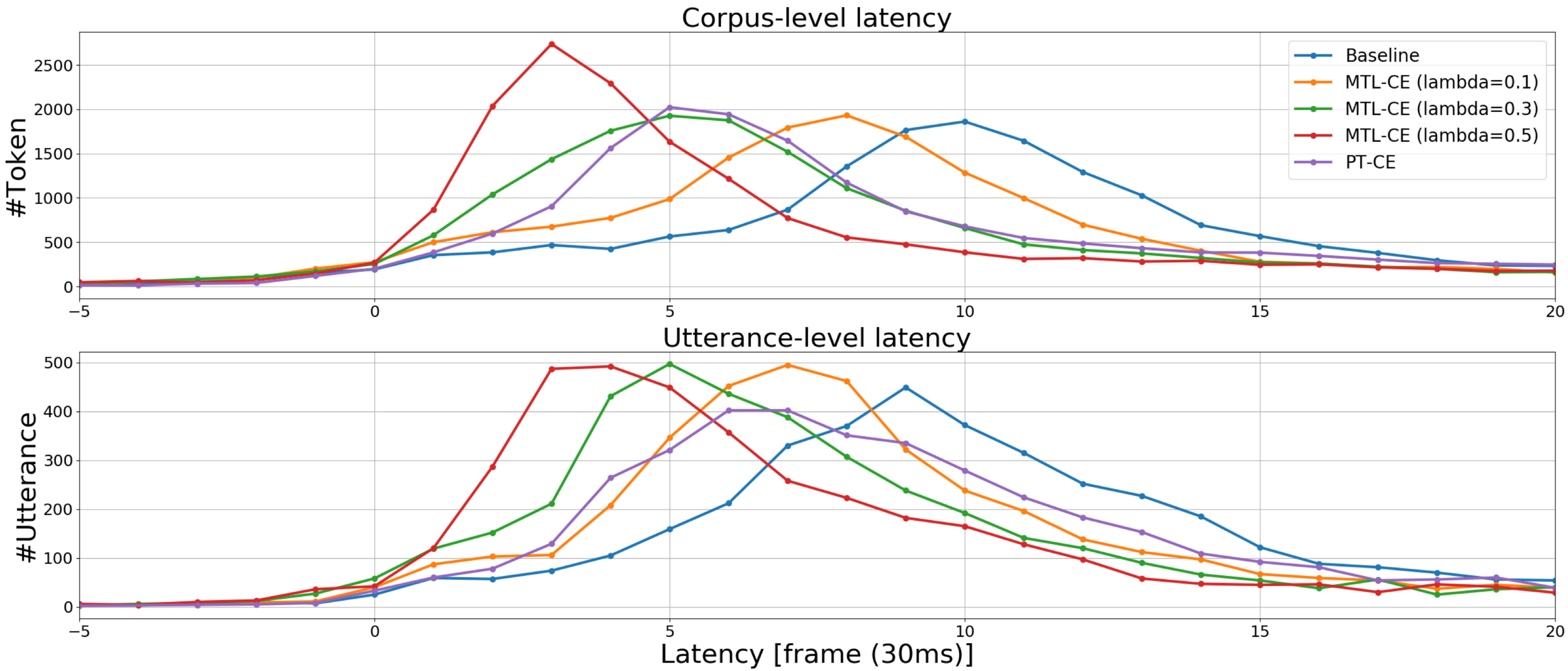
  (2) offline <-> streaming S2S
- 1D-convolution layer improved the streaming MoChA by **4.24%** relatively

# Results: Alignments on the encoder side

| Model | WER [%] | Corpus-level [frame (30ms)] | | | |
|---|---|---|---|---|---|
| | | Ave. | Med. | 90th | 99th |
| Baseline MoChA | 9.93 | 11.65 | 10.00 | 21.39 | **44.29** |
| MTL-CE ($\lambda_{\mathrm{CE}} = 0.1$) | 10.21 5.6% | 9.84 | 8.00 40% | **19.42** | 46.54 |
| MTL-CE ($\lambda_{\mathrm{CE}} = 0.3$) | 10.48 | 8.78 | 6.00 | 19.69 | 47.96 |
| MTL-CE ($\lambda_{\mathrm{CE}} = 0.5$) | 11.11 | **8.36** | **5.00** | 21.21 | 49.86 |
| PT-CE | 12.74 | 10.49 | 7.00 | 22.90 | 48.65 |

- MTL-CE reduced latency in proportion to $\lambda_{\mathrm{CE}}$ while degrading WER slightly
- PT-CE also reduced latency but degraded WER too much
- Contrastive results to previous works using CTC + framewise CE objective
  - ➢MoChA is a label-synchronous model

# Visualization of latency distribution (encoder)

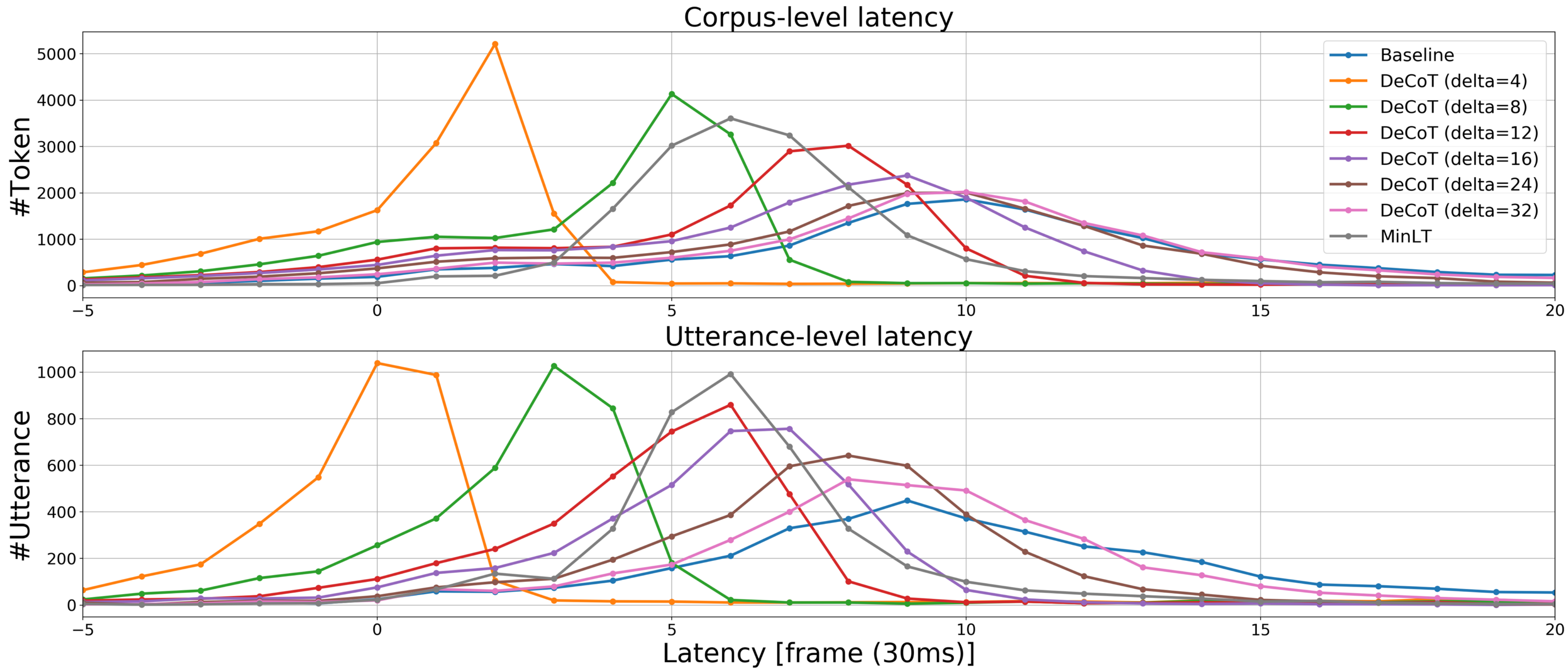


# Results: Alignments on the decoder side

| Model | WER [%] | Corpus-level [frame (30ms)] | | | |
|---|---|---|---|---|---|
| | | Ave. | Med. | 90th | 99th |
| UniGRU global attention (offline) | 8.44 | N/A | N/A | N/A | N/A |
| Baseline MoChA | 9.93 | 11.65 | 10.00 | 21.39 | 44.29 |
| DeCoT ($\delta = 4$) | 20.25 | 3.66 | 1.00 | 9.56 | 62.27 |
| DeCoT ($\delta = 8$) | 14.35 | 4.60 | 5.00 | 7.00 | 47.04 |
| DeCoT ($\delta = 12$) | 11.40 | 6.02 | 7.00 | 9.92 | 35.58 |
| DeCoT ($\delta = 16$) | 9.13 | **6.63** | 8.00 | 11.71 | **16.43** |
| DeCoT ($\delta = 24$) | **8.87** | 8.37 | 9.00 | 14.45 | 21.07 |
| DeCoT ($\delta = 32$) | 9.17 | 9.79 | 10.00 | 16.54 | 27.01 |
| MinLT | 9.70 | 7.06 | **6.00** | **10.63** | 26.76 |

- DeCoT: large WER improvement and moderate latency reduction (tail part)
- MinLT: small WER improvement and large latency reduction (median)

# Visualization of latency distribution (decoder)



# Ablation study: Decoder side

- **Combination of DeCoT and MinLT reduced the latency, but degraded WER too much**

| Model | WER [%] | Corpus-level [frame (30ms)] | | | |
|---|---|---|---|---|---|
| | | Ave. | Med. | 90th | 99th |
| DeCoT ($\delta = 16$) | 9.13 | 6.63 | 8.00 | 11.71 | 16.43 |
| + MinLT | 12.75 | **4.05** | **4.00** | **7.96** | **15.92** |
| MinLT | 9.70 | 7.06 | 6.00 | 10.63 | 26.76 |

# Ablation study: Decoder side

Microsoft

- **Quantity loss was essential for DeCoT but not necessary for the baseline and MinLT**

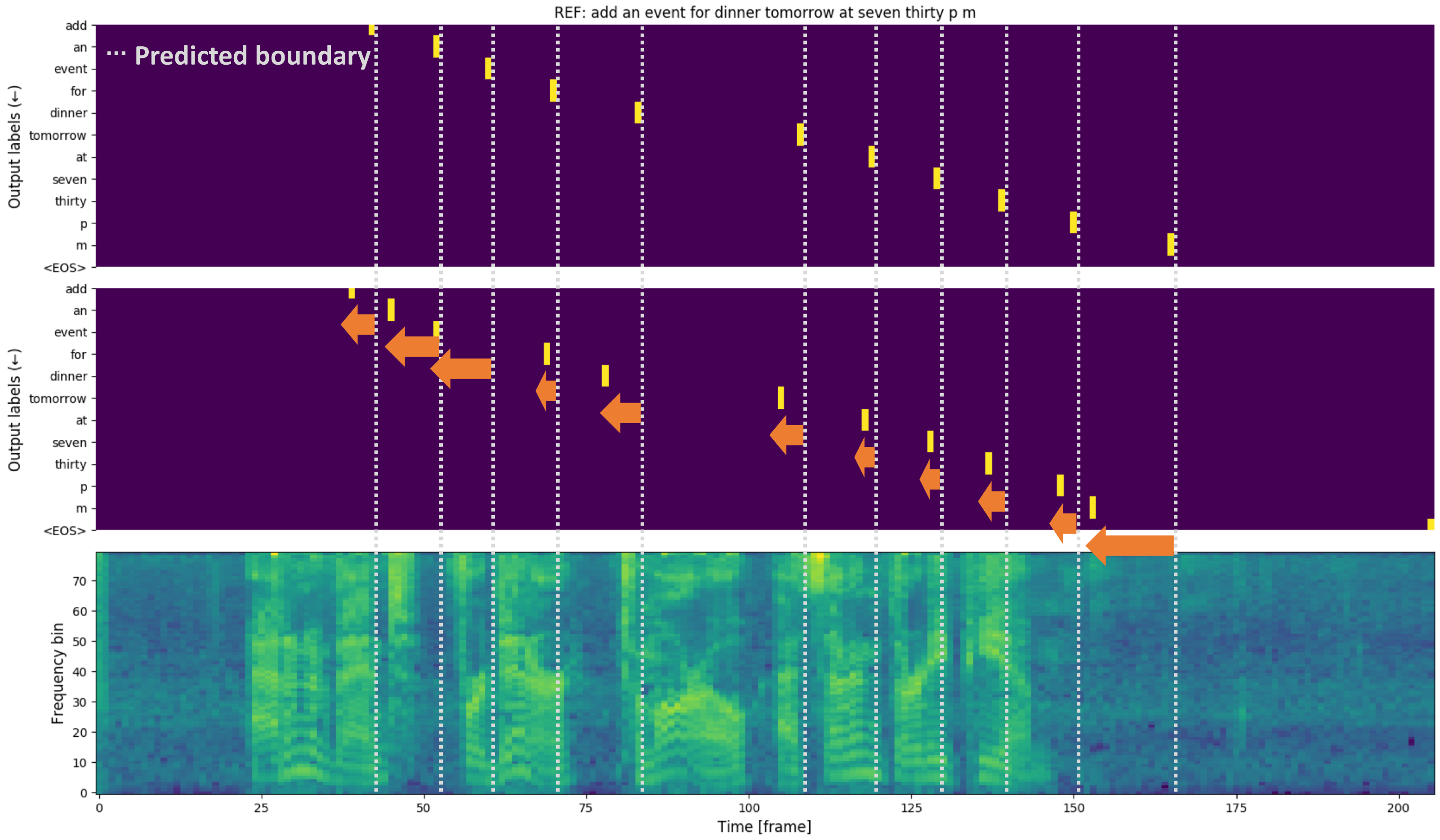
| Model | WER [%] | Corpus-level [frame (30ms)] | | | |
|---|---|---|---|---|---|
| | | Ave. | Med. | 90th | 99th |
| Baseline MoChA | 9.93 | 11.65 | 10.00 | 21.39 | 44.29 |
| + Quantity loss | 10.30 | 11.24 | 10.00 | 20.39 | 36.01 |
| DeCoT ($\delta = 16$) | 9.13 | 6.63 | 8.00 | 11.71 | 16.43 |
| - Quantity loss | 14.28 | 3.93 | 3.00 | 7.20 | 27.39 |
| MinLT | 9.70 | 7.06 | 6.00 | 10.63 | 26.76 |
| + Quantity loss | 13.66 | 6.82 | 6.00 | 10.45 | 25.57 |

# Ablation study: Decoder side

- **Warm start training from the baseline was effective for DeCoT and MinLT**

| Model | WER [%] | Corpus-level [frame (30ms)] | | | |
|---|---|---|---|---|---|
| | | Ave. | Med. | 90th | 99th |
| Baseline MoChA | 9.93 | 11.65 | 10.00 | 21.39 | 44.29 |
| + Warm start training | 9.21 | 12.27 | 11.00 | 22.23 | 43.16 |
| DeCoT ($\delta = 16$) | 9.13 | 6.63 | 8.00 | 11.71 | 16.43 |
| - Warm start training | 10.72 | 6.28 | 7.00 | 11.12 | 36.03 |
| MinLT | 9.70 | 7.06 | 6.00 | 10.63 | 26.76 |
| - Warm start training | 13.63 | 11.83 | 10.00 | 21.41 | 45.06 |

# Alignment visualization




Baseline MoChA

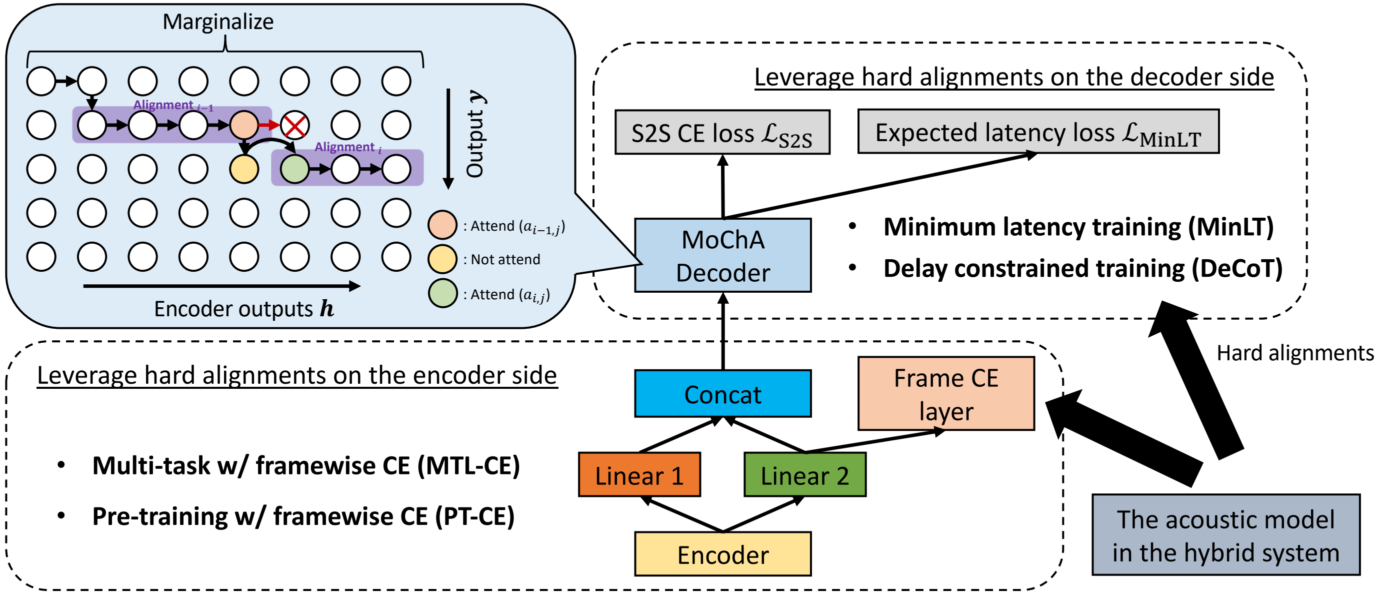
DeCoT ($\delta = 16$)

# Conclusion

Microsoft

- Explored to leverage frame-level hard alignments extracted from the hybrid system to reduce user perceived latency

- Alignments were effective for latency reduction on both sides, and also improved ASR performance when applying on the decoder side



Question?

E-mail: inaguma [at] sap.ist.i.kyoto-u.ac.jp