Joint CTC-Attention based End-to-end Speech Recognition using Multi-task Learning

Suyoun Kim, Takaaki Hori, and Shinji Watanabe

Presenter: Desh Raj

Outline

- CTC and attention—the good and the bad
- The joint CTC-attention model
- Experimental results

End-to-end ASR

- Several issues with hybrid DNN-HMM models
- Several independent moving components—acoustic model, language model, lexicon, etc.
- Make conditional independence assumptions and approximations
- End-to-end models learn acoustic frames to character mapping

End-to-end ASR

- Two main approaches:
    1. Connectionist temporal classification (CTC)
    2. Attention-based encoder decoder

CTC

- Uses intermediate label representation—allows repetitions of labels and occurrence of a blank label

$$P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in \Phi(\boldsymbol{y}')} P(\boldsymbol{\pi}|\boldsymbol{x}),$$

Sum over all possible intermediate label representations

CTC

- Uses intermediate label representation—allows repetitions of labels and occurrence of a blank label

$$P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in \Phi(\boldsymbol{y'})} P(\boldsymbol{\pi}|\boldsymbol{x}),$$

Conditional independence of output labels

$$\boxed{P(\boldsymbol{\pi}|\boldsymbol{x}) \approx \prod_{t=1}^{T} P(\pi_t|\boldsymbol{x}) = \prod_{t=1}^{T} q_t(\pi_t)}$$

CTC

- Uses intermediate label representation—allows repetitions of labels and occurrence of a blank label

$$P(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi} \in \Phi(\boldsymbol{y'})} P(\boldsymbol{\pi}|\boldsymbol{x}),$$

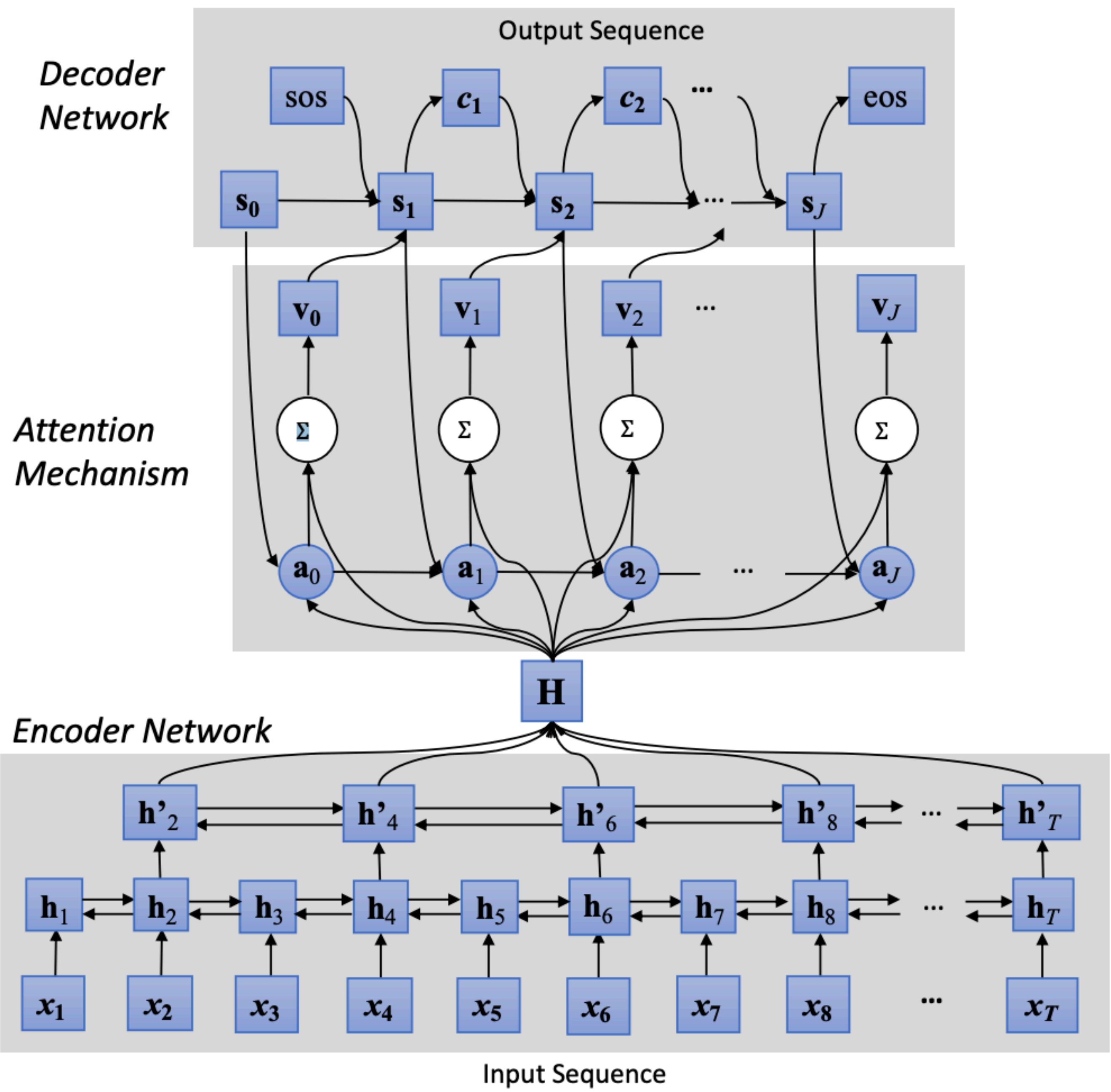
- Can just use forward-backward to compute

Attention

- No conditional independence assumptions

$$
\begin{aligned}
P(\boldsymbol{y}|\boldsymbol{x}) &= \prod_u P(y_u|\boldsymbol{x}, y_{1:u-1}) \\
\boldsymbol{h} &= \mathrm{Encoder}(\boldsymbol{x}) \\
y_u &\sim \mathrm{AttentionDecoder}(\boldsymbol{h}, y_{1:u-1}).
\end{aligned}
$$

Attention



Output Sequence
Decoder Network
sos
c_1
c_2
...
eos
$\mathbf{s}_0$
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_J$
$\mathbf{v}_0$
$\mathbf{v}_1$
$\mathbf{v}_2$
$\mathbf{v}_J$
Attention Mechanism
Σ
$\mathbf{a}_0$
$\mathbf{a}_1$
$\mathbf{a}_2$
$\mathbf{a}_J$
H
Encoder Network
$\mathbf{h'}_2$
$\mathbf{h'}_4$
$\mathbf{h'}_6$
$\mathbf{h'}_8$
$\mathbf{h'}_T$
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_3$
$\mathbf{h}_4$
$\mathbf{h}_5$
$\mathbf{h}_6$
$\mathbf{h}_7$
$\mathbf{h}_8$
$\mathbf{h}_T$
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_T
Input Sequence

Attention

- No conditional independence assumptions

$$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_u P(y_u|\boldsymbol{x}, y_{1:u-1})$$

$$\boldsymbol{h} = \text{Encoder}(\boldsymbol{x})$$

$$y_u \sim \text{AttentionDecoder}(\boldsymbol{h}, y_{1:u-1}).$$

Can be content-based or location-based

Attention

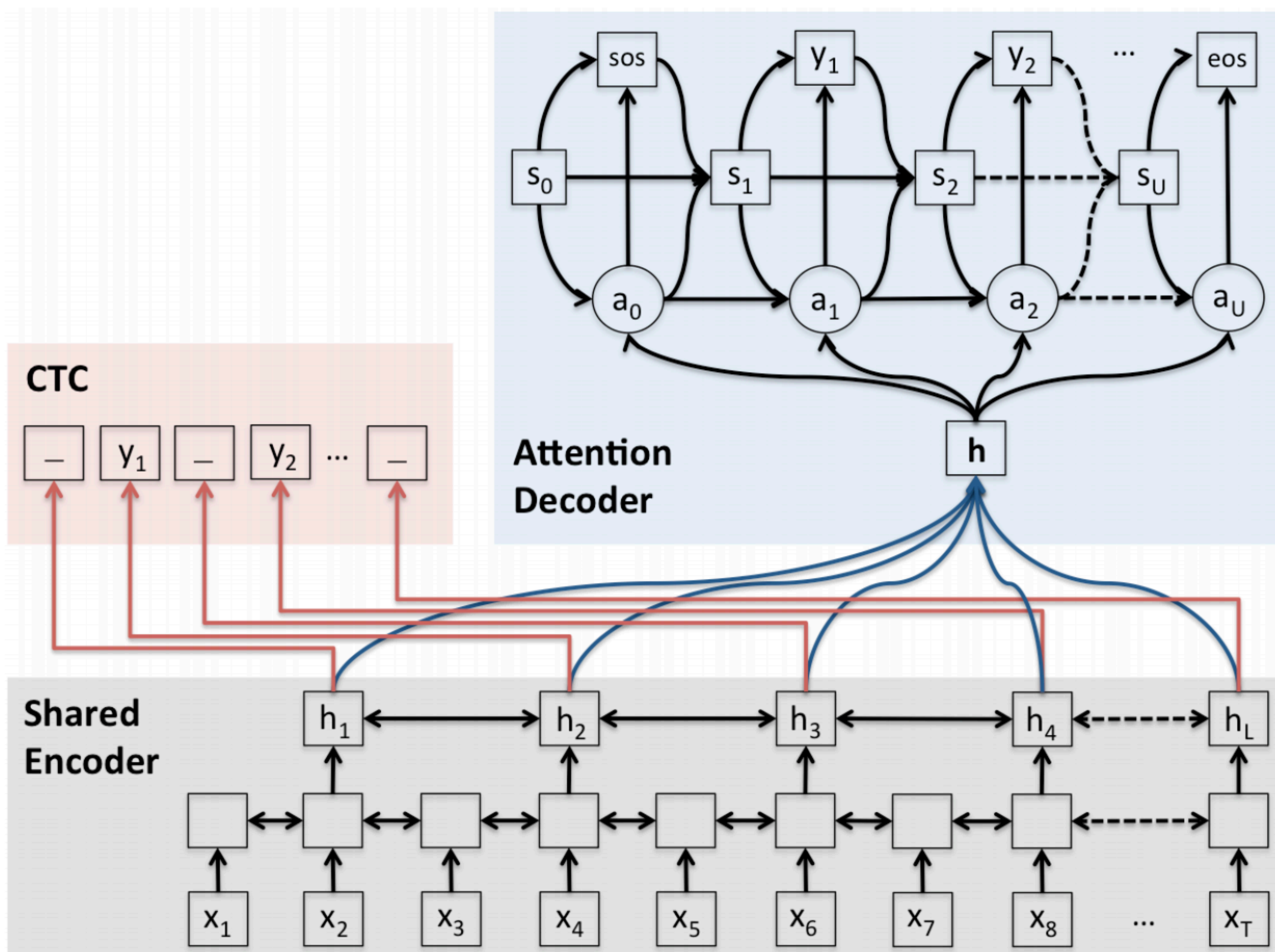
- So what’s the problem?
- Too much flexibility—easily affected by noise.
- Also hard to train from scratch on long input sequences.

Attention

- So what's the problem?
- Too much flexibility—easily affected by noise.
- Also hard to train from scratch on long input sequences.

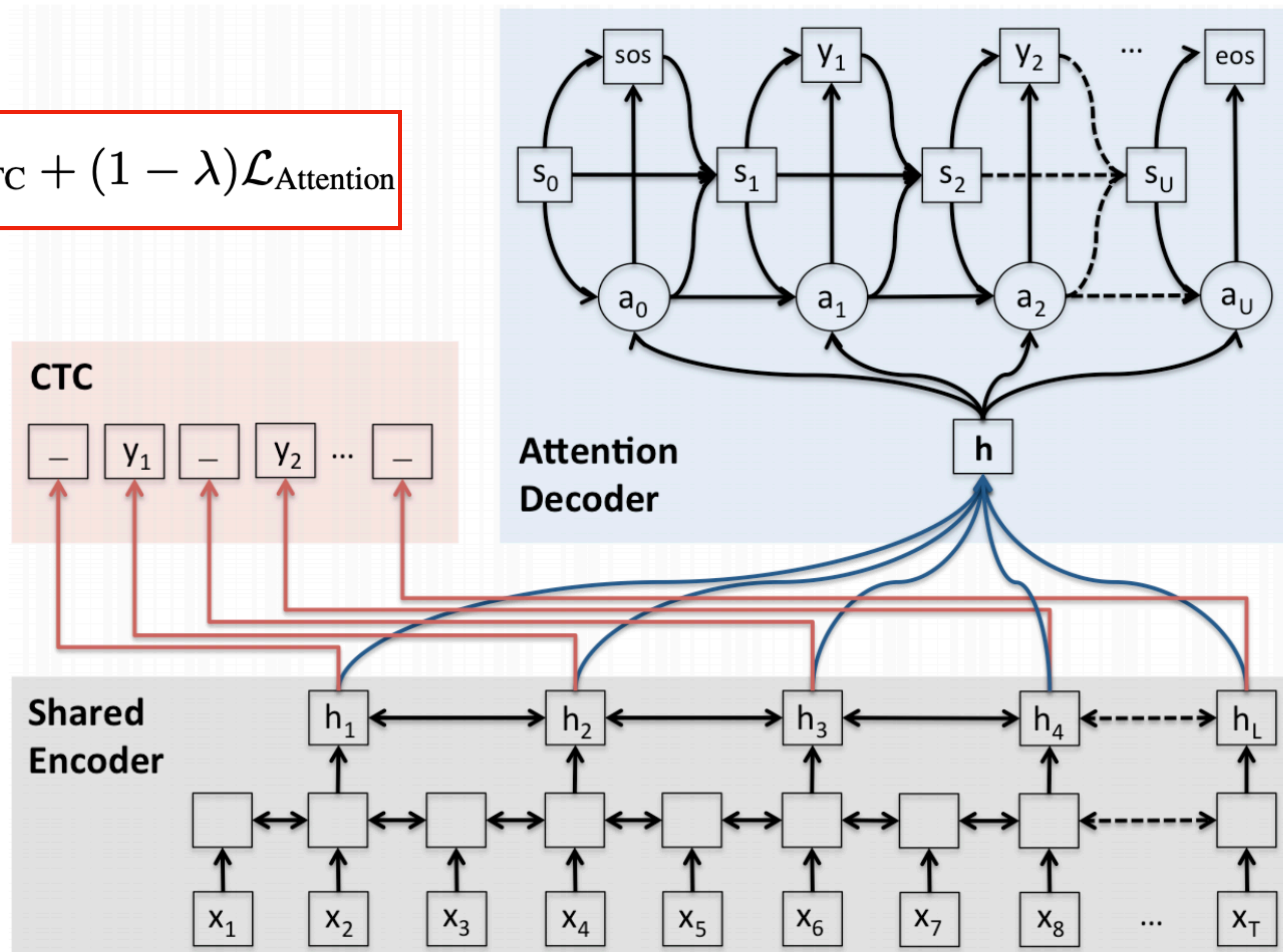
CTC models don't have these problems since they impose left-to-right constraints

Joint CTC-Attention



Joint CTC-Attention

$$\boxed{\mathcal{L}_{\text{MTL}} = \lambda \mathcal{L}_{\text{CTC}} + (1 - \lambda) \mathcal{L}_{\text{Attention}}}$$



Experiments

- 3 datasets—WSJ1 (81 hours), WSJ0 (15 hours), and Chime-4 (18 hours)

- 40 Mel-scale filterbank coefficients + first and second order temporal derivatives = 120 feature values

- No LM or lexicon used

Experiments

- Encoder—4-layer Bi-LSTM
- Top 2 layers perform sequence contraction by half each
- Decoder—1-layer LSTM

Results

Model(train)	CER(valid)	CER(eval)
WSJ-train_si284 (80hrs)	dev93	eval92
CTC	11.48	8.97
Attention(content-based)	13.68	11.08
Attention(location-based)	11.98	8.17
MTL($\lambda = 0.2$)	**11.27**	**7.36**
MTL($\lambda = 0.5$)	12.00	8.31
MTL($\lambda = 0.8$)	11.71	8.45
WSJ-train_si84 (15hrs)	dev93	eval92
CTC	27.41	20.34
Attention(content-based)	28.02	20.06
Attention(location-based)	24.98	17.01
MTL($\lambda = 0.2$)	**23.03**	**14.53**
MTL($\lambda = 0.5$)	26.28	16.24
MTL($\lambda = 0.8$)	32.21	21.30
CHiME-4-tr05_multi (18hrs)	dt05_real	et05_real
CTC	37.56	48.79
Attention(content-based)	43.45	54.25
Attention(location-based)	35.01	47.58
MTL($\lambda = 0.2$)	**32.08**	**44.99**
MTL($\lambda = 0.5$)	34.56	46.49
MTL($\lambda = 0.8$)	35.41	48.34

Clean environment—possible that CTC improved generalization since its training does not use character inter-dependencies

Results

Model(train)	CER(valid)	CER(eval)
WSJ-train_si284 (80hrs)	dev93	eval92
CTC	11.48	8.97
Attention(content-based)	13.68	11.08
Attention(location-based)	11.98	8.17
MTL($\lambda = 0.2$)	**11.27**	**7.36**
MTL($\lambda = 0.5$)	12.00	8.31
MTL($\lambda = 0.8$)	11.71	8.45
WSJ-train_si84 (15hrs)	dev93	eval92
CTC	27.41	20.34
Attention(content-based)	28.02	20.06
Attention(location-based)	24.98	17.01
MTL($\lambda = 0.2$)	**23.03**	**14.53**
MTL($\lambda = 0.5$)	26.28	16.24
MTL($\lambda = 0.8$)	32.21	21.30
CHiME-4-tr05_multi (18hrs)	dt05_real	et05_real
CTC	37.56	48.79
Attention(content-based)	43.45	54.25
Attention(location-based)	35.01	47.58
MTL($\lambda = 0.2$)	**32.08**	**44.99**
MTL($\lambda = 0.5$)	34.56	46.49
MTL($\lambda = 0.8$)	35.41	48.34

Noisy environment—much better than attention-based model

Results

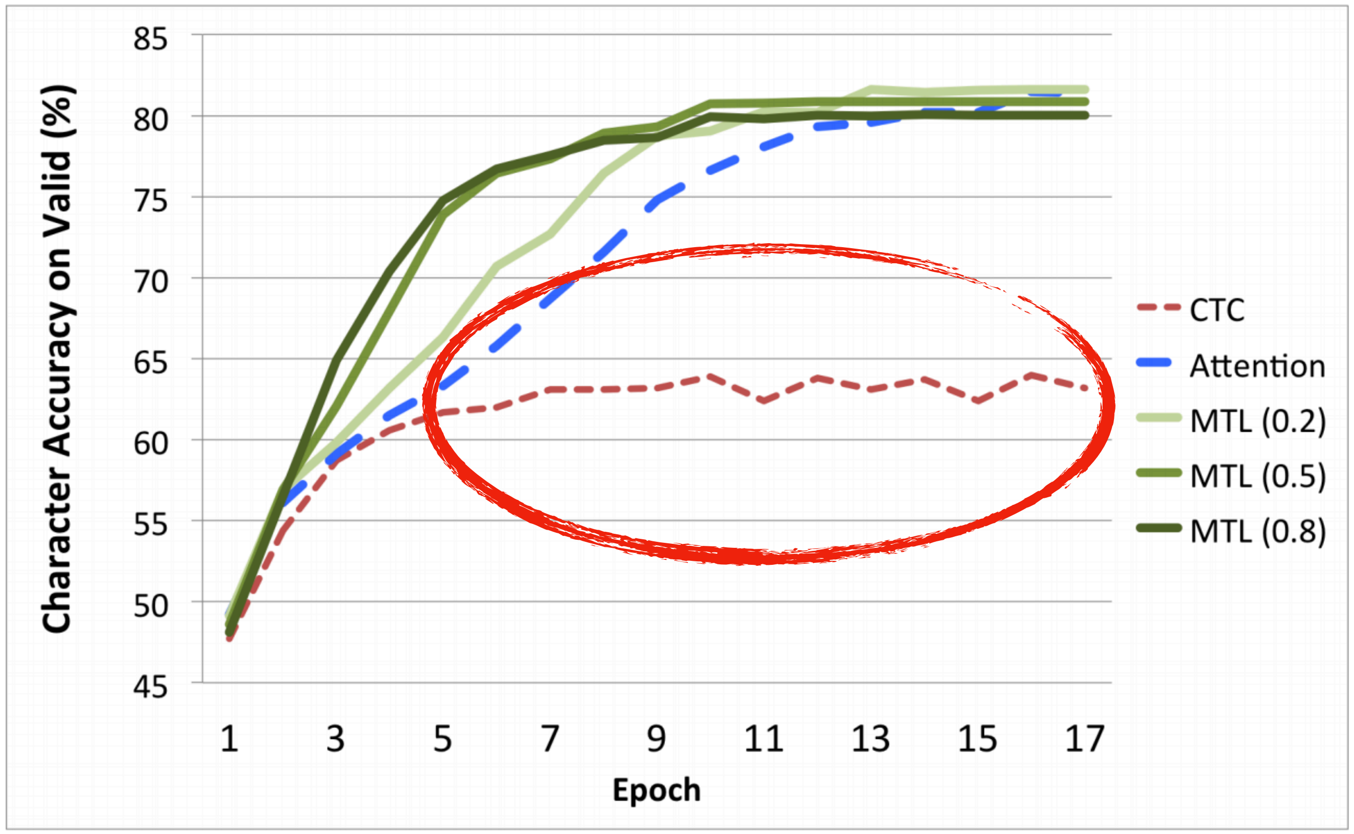


Fig. 2: Comparison of learning curves: CTC, location-based attention model, and MTL with ($\lambda = 0.2, 0.5, 0.8$). The character accuracy on the validation set of CHiME-4 is calculated by edit distance between hypothesis and reference. Note that the reference history were used in the attention and our MTL models.

CTC trains quickly but low accuracy

Results

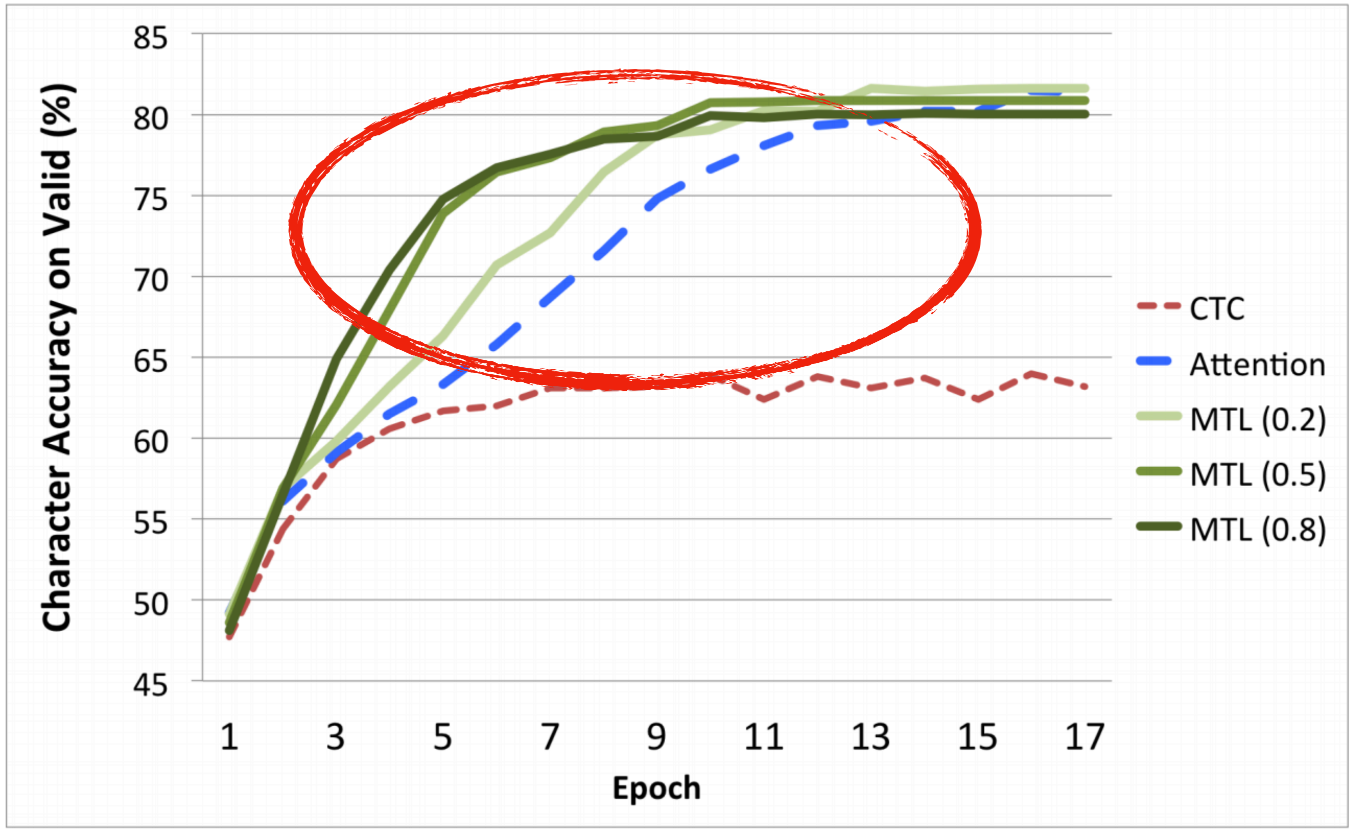


Fig. 2: Comparison of learning curves: CTC, location-based attention model, and MTL with ($\lambda = 0.2, 0.5, 0.8$). The character accuracy on the validation set of CHiME-4 is calculated by edit distance between hypothesis and reference. Note that the reference history were used in the attention and our MTL models.

Attention-based model reaches same accuracy as MTL but takes twice as much time

Results

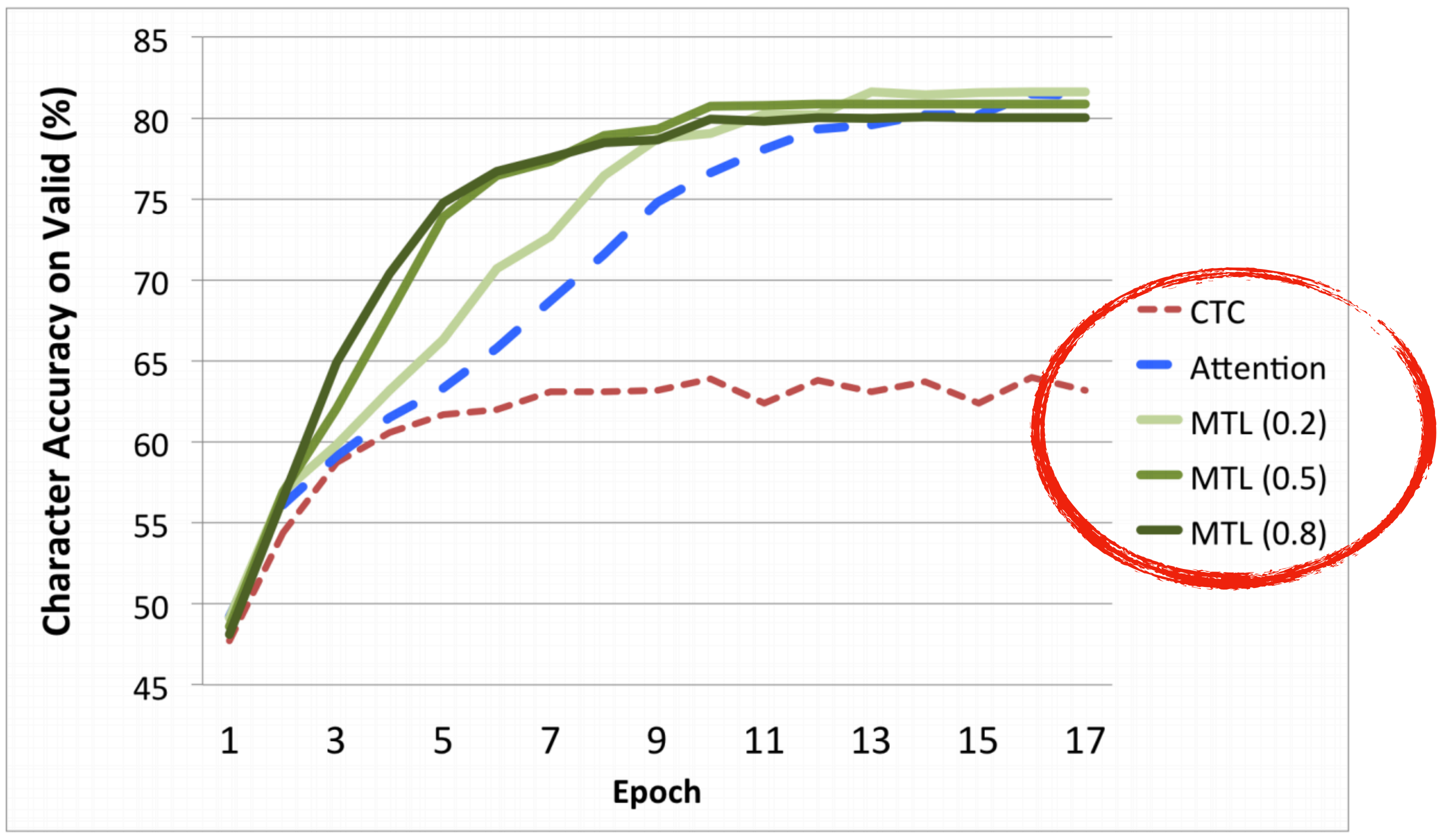
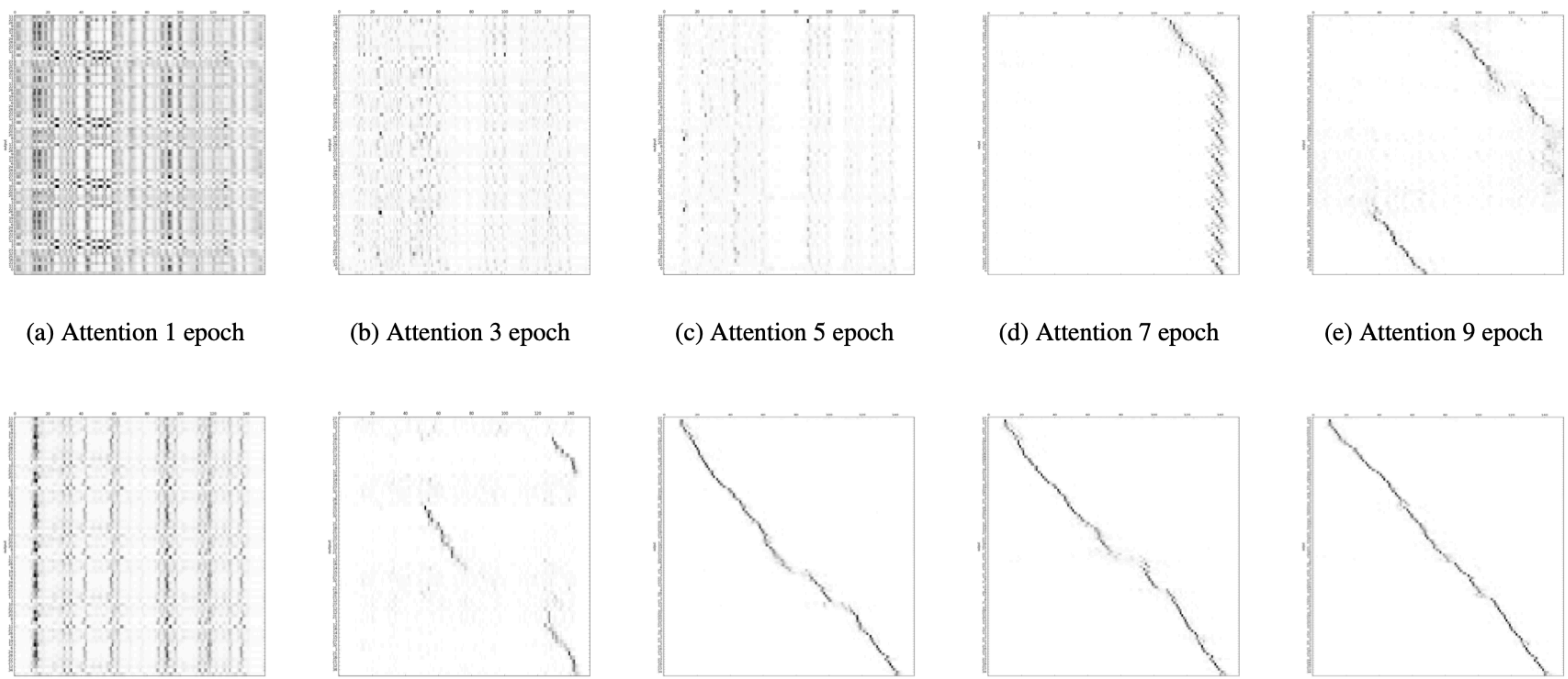


Fig. 2: Comparison of learning curves: CTC, location-based attention model, and MTL with ($\lambda = 0.2, 0.5, 0.8$). The character accuracy on the validation set of CHiME-4 is calculated by edit distance between hypothesis and reference. Note that the reference history were used in the attention and our MTL models.

More weight to CTC loss-> faster convergence

Results

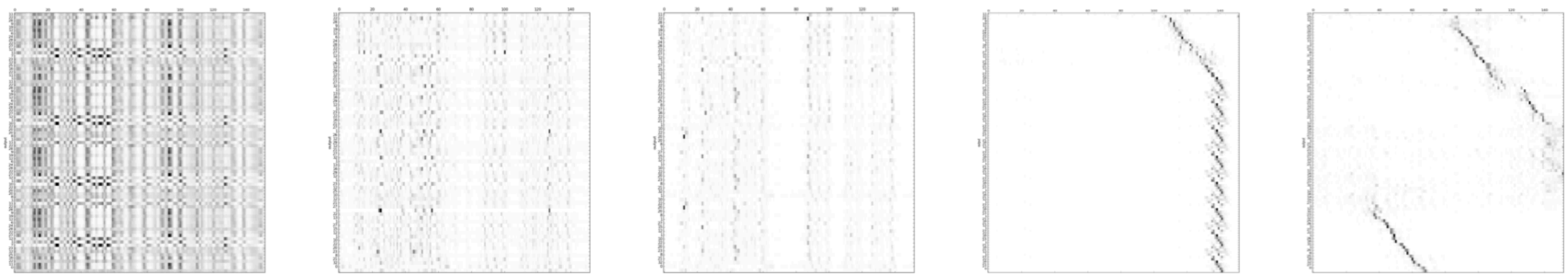


(a) Attention 1 epoch (b) Attention 3 epoch (c) Attention 5 epoch (d) Attention 7 epoch (e) Attention 9 epoch

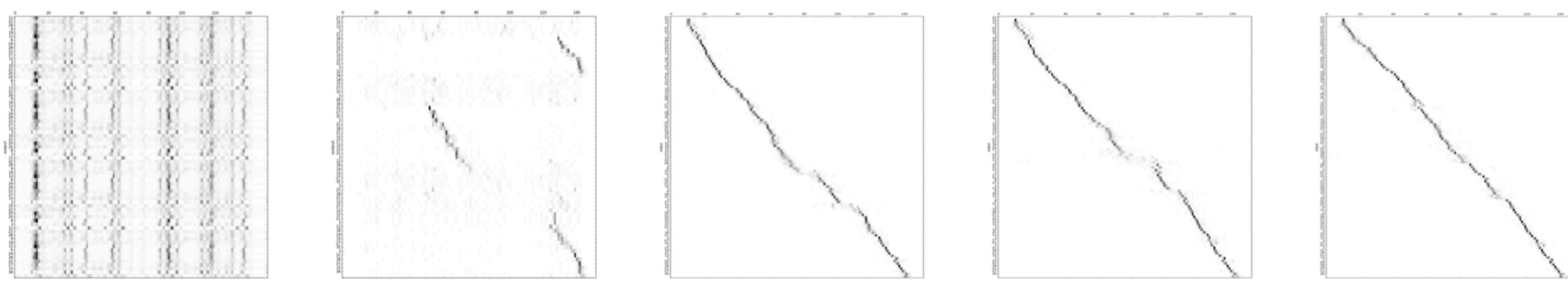
(f) MTL 1 epoch (g) MTL 3 epoch (h) MTL 5 epoch (i) MTL 7 epoch (j) MTL 9 epoch

Attention alignments between characters and acoustic frames

Results



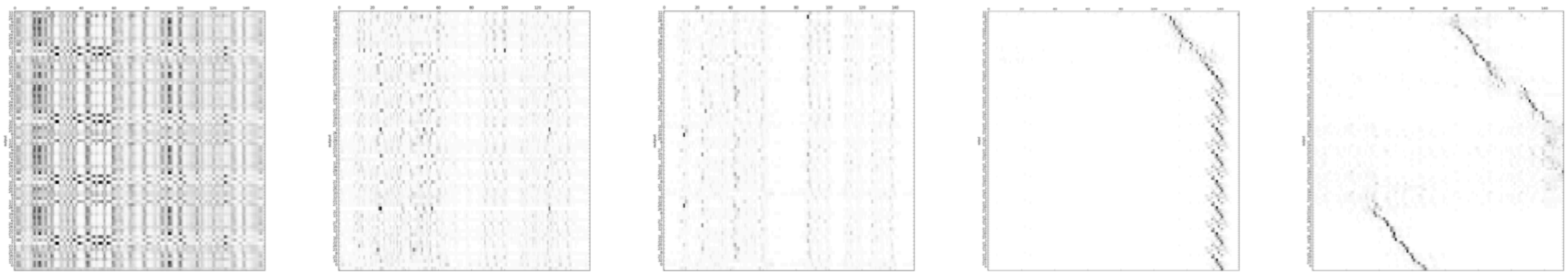
(a) Attention 1 epoch (b) Attention 3 epoch (c) Attention 5 epoch (d) Attention 7 epoch (e) Attention 9 epoch



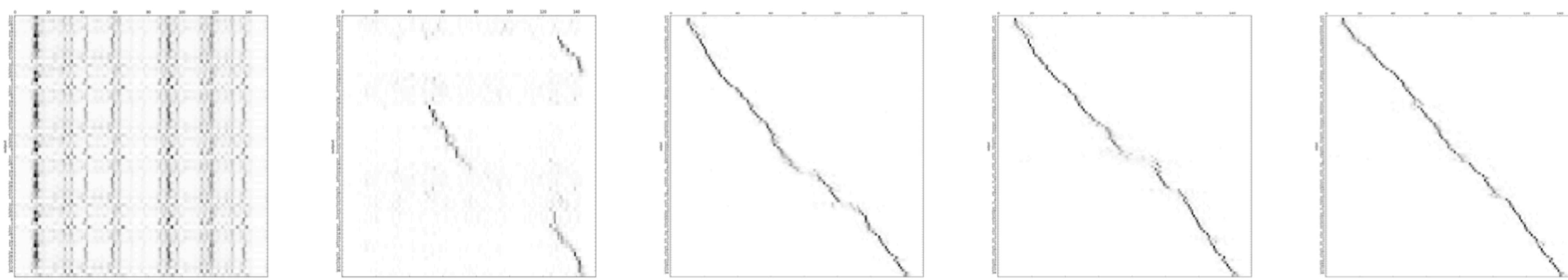
(f) MTL 1 epoch (g) MTL 3 epoch (h) MTL 5 epoch (i) MTL 7 epoch (j) MTL 9 epoch

Does not learn desired alignments even after 9 epochs

Results



(a) Attention 1 epoch (b) Attention 3 epoch (c) Attention 5 epoch (d) Attention 7 epoch (e) Attention 9 epoch



(f) MTL 1 epoch (g) MTL 3 epoch (h) MTL 5 epoch (i) MTL 7 epoch (j) MTL 9 epoch

Learns desired alignment after 5 epochs

Key takeaways

- Combining CTC and attention performs better on both clean and noisy data
- Speeds up training significantly
- Also gives desired alignments unlike attention

Thank you!

Questions? Comments?