CTC Alignments Improve Autoregressive Translation

Brian Yan, Siddharth Dalmia, Yosuke Higuchi, Graham Neubig, Florian Metze, Alan W Black, Shinji Watanabe

CTC vs. Attention

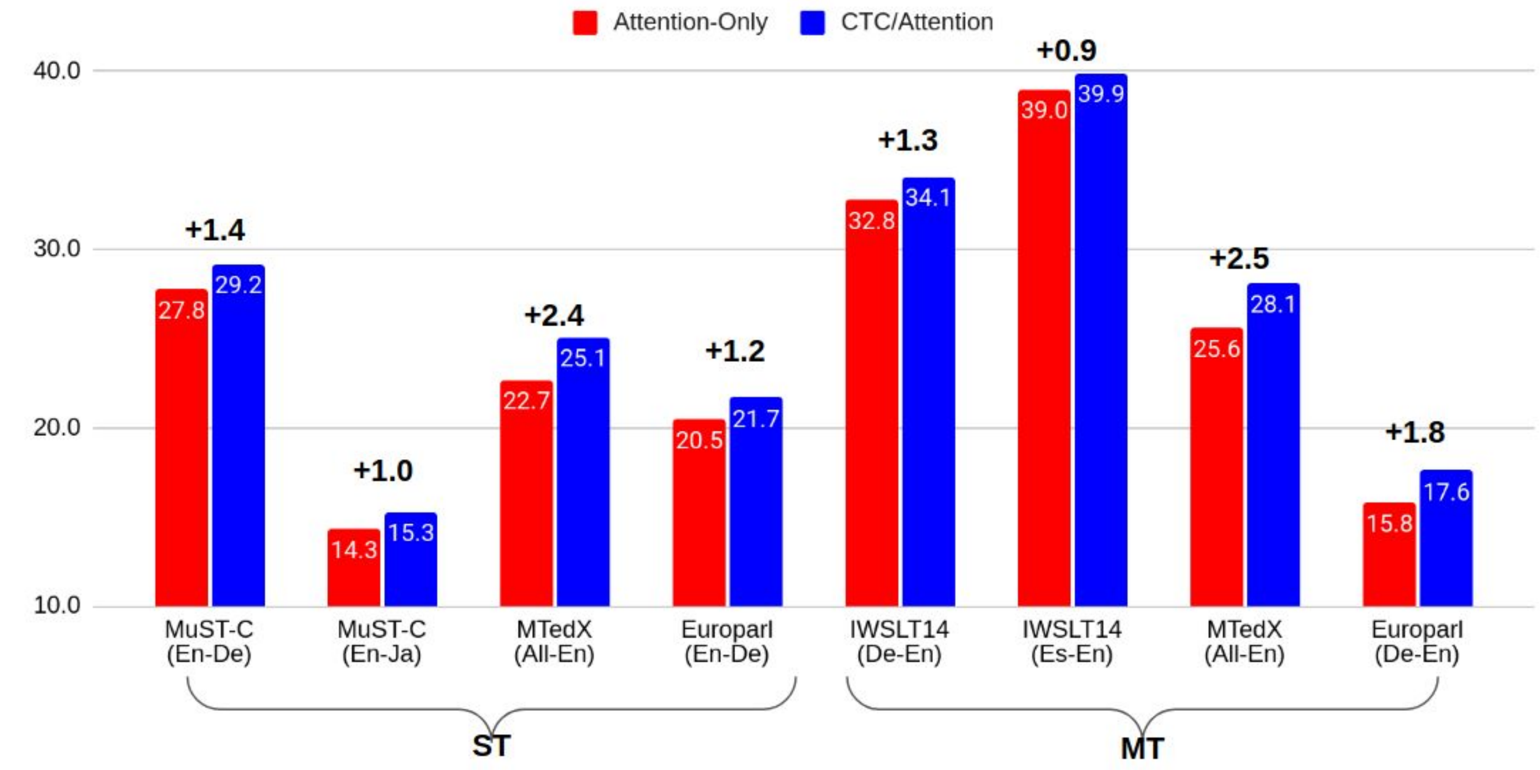
CTC	Attention
$P_{\text{CTC}}(Y\|X) \triangleq \sum_{Z\in\mathcal{Z}} \prod_{t=1}^{T} P(z_t\|X, \cancel{z_{1:t-1}})$	$P_{\text{Attn}}(Y\|X) \triangleq \prod_{l=1}^{L} P(y_l\|y_{1:l-1}, X)$
Hard Alignment Criterion only allows monotonic alignments of inputs to outputs	**Soft Alignment** Flexible attention-based input-to-output mappings may overfit to irregular patterns
Conditional Independence Assumes that there are no dependencies between each output unit given the input	**Conditional Dependence** Locally normalized models with output dependency exhibit label/exposure biases
Input-Synchronous Emission Each input representation emits exactly one blank or non-blank output token	**Autoregressive Generation** Need to detect end-points and compare hypotheses of different length in beam search

Weaker for translation (Gu+ 2021, Huang+ 2022)

But attention is not perfect (Murray+ 2018, Hannun+ 2019, Watanabe+ 2018)

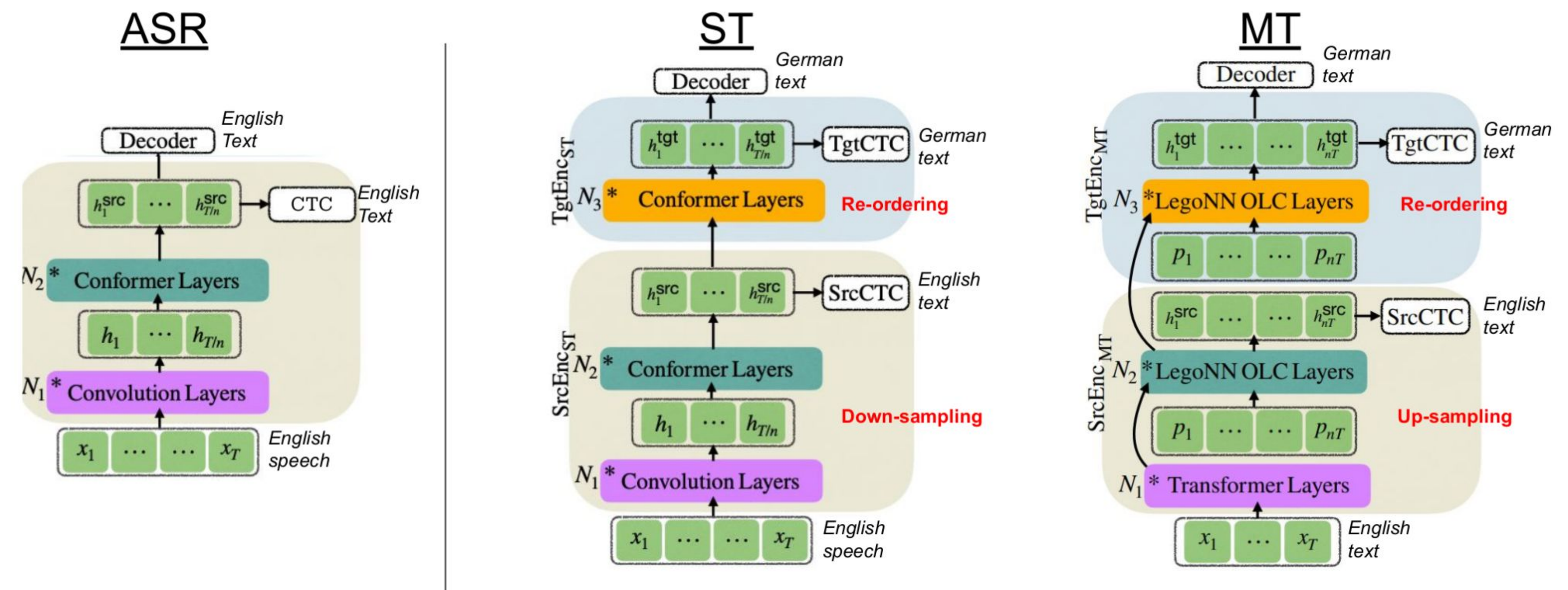
Joint CTC/Attention: Results

- Joint CTC/attention outperforms pure-attention by an average of **+1.6 BLEU**



Joint Training with Hierarchical CTC

- For ASR, CTC and attention simply share a monolithic encoder
- For MT/ST, we decompose the encoder into 2 stages: **1) length-adjustment 2) re-ordering**



$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{CTC}} + \lambda_2 \mathcal{L}_{\text{ATTN}}$

$\mathcal{L} = \mathcal{L}_{\text{SrcCTC}} + \lambda_1 \mathcal{L}_{\text{TgtCTC}} + \lambda_2 \mathcal{L}_{\text{ATTN}}$

Joint Decoding with Output-Sync Beam Search

- Attention plays a primary role while CTC plays a secondary role (Watanabe+ 2018)

Algorithm 1 *Output*-Synchronous Step Function: attentional decoder proposes candidates to expand hypotheses which are all of l-length at step l.

```
procedure OUTPUTSTEP(prtHs, X, l, p, maxL)
    newPrtHs = {}; endHs = {}
    for y_{1:l-1} ∈ prtHs do
        attnCnds = top-k(P_Attn(y_l | X, y_{1:l-1}), k = p)
        for c ∈ attnCnds do
            y_{1:l} = y_{1:l-1} ⊕ c
            α_CTC = CTCScore(y_{1:l}, X_{1:T})
            α_Attn = AttnScore(y_{1:l}, X_{1:T})
            β = LengthPen(y_{1:l})
            P_Beam(y_{1:l} | X) = α_CTC + α_Attn + β
            if (c is <eos>) or (l is maxL) then
                endHs[y_{1:l}] = P_Beam(·)
            else
                newPrtHs[y_{1:l}] = P_Beam(·)
            end if
        end for
    end for
    return newPrtHs, endHs
end procedure
```

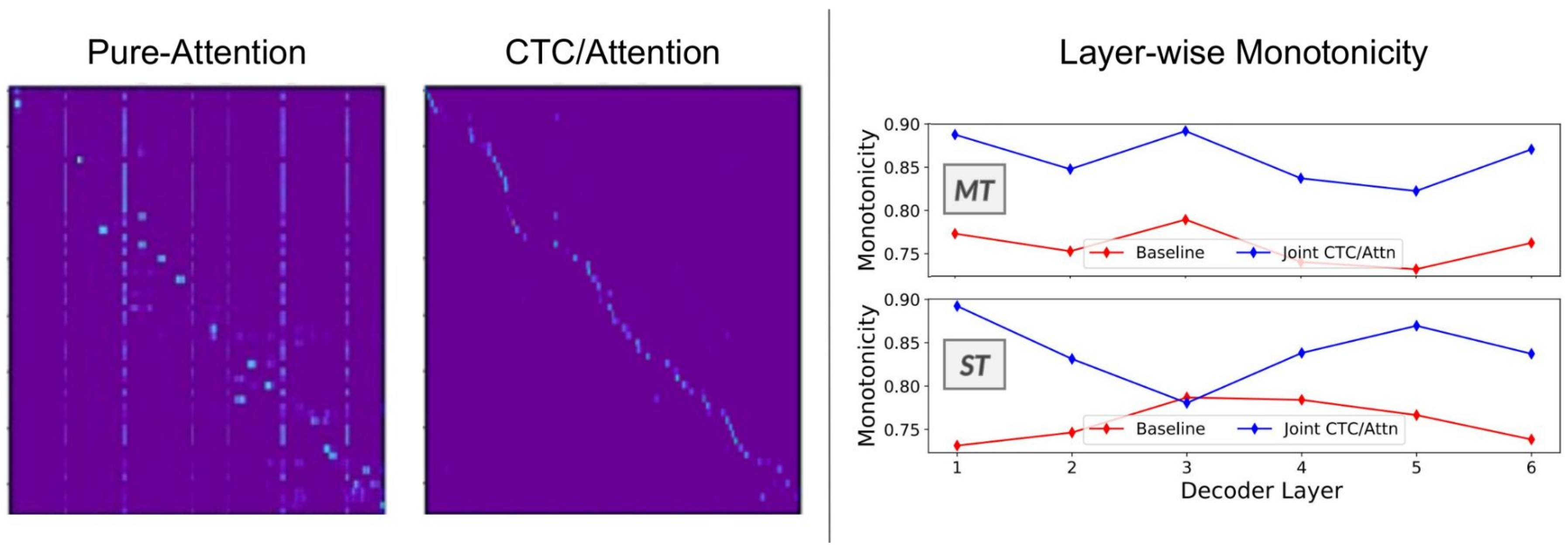
Hypothesis Expansion — Choose top-p candidates per hypothesis (e.g. p=1.5*b) **from attention posteriors**

Joint Scoring — Interpolate label sequence likelihoods **from attention and CTC**

End Detection — End loop based on conditions **from attention** (CTC's role in end-detection is implicit)

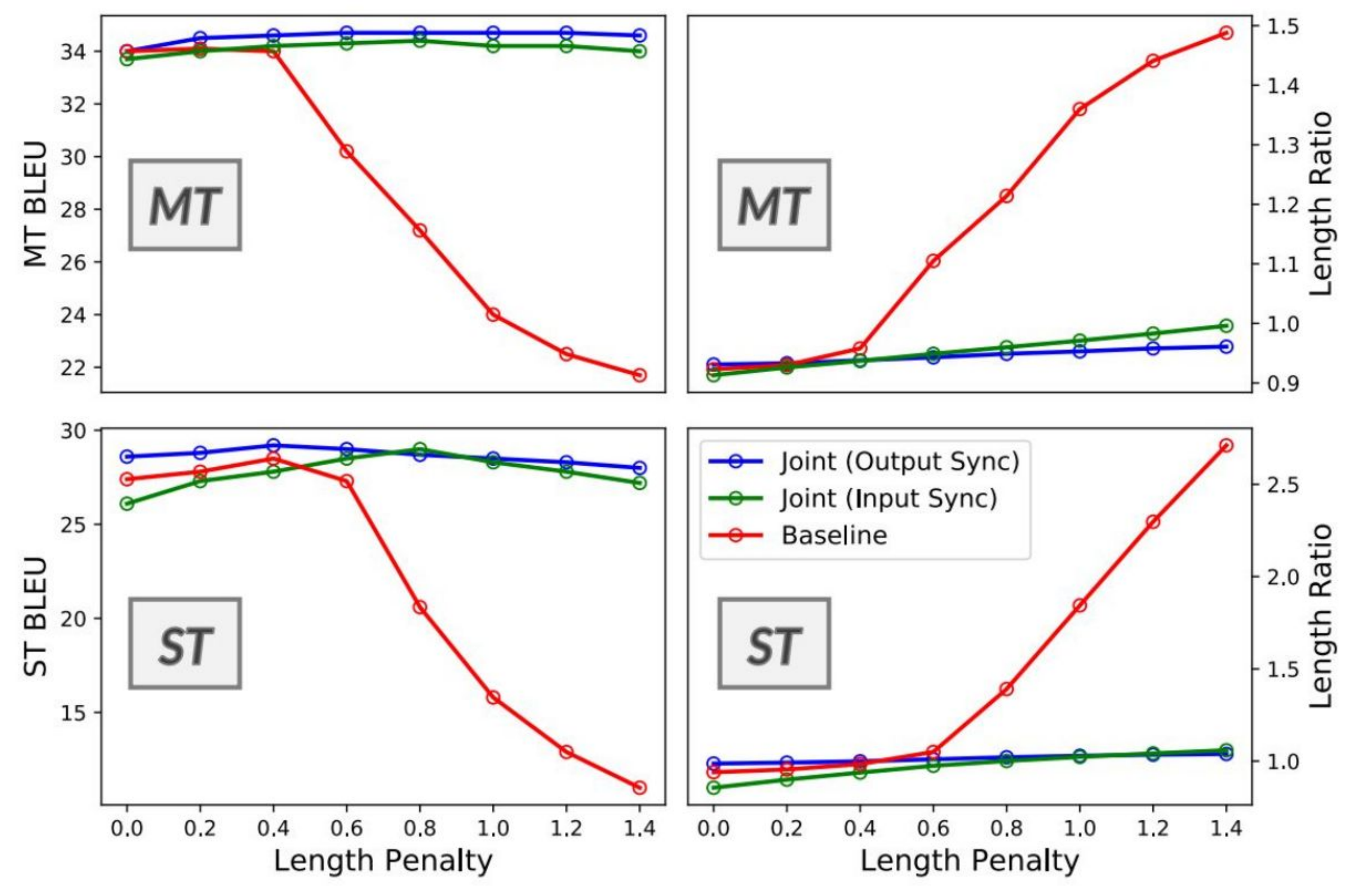
Reduced Soft Alignment Burden

- **Hard alignment** + **soft alignment**
  - *Conjecture*: hard alignment objective produces stable encoder representations allowing the decoder to **more easily learn soft alignment patterns during training**
  - *Concern*: can CTC encoders perform input-to-output mappings for translation?
  - *Finding*: joint training results in more regular, diagonal source-attention patterns



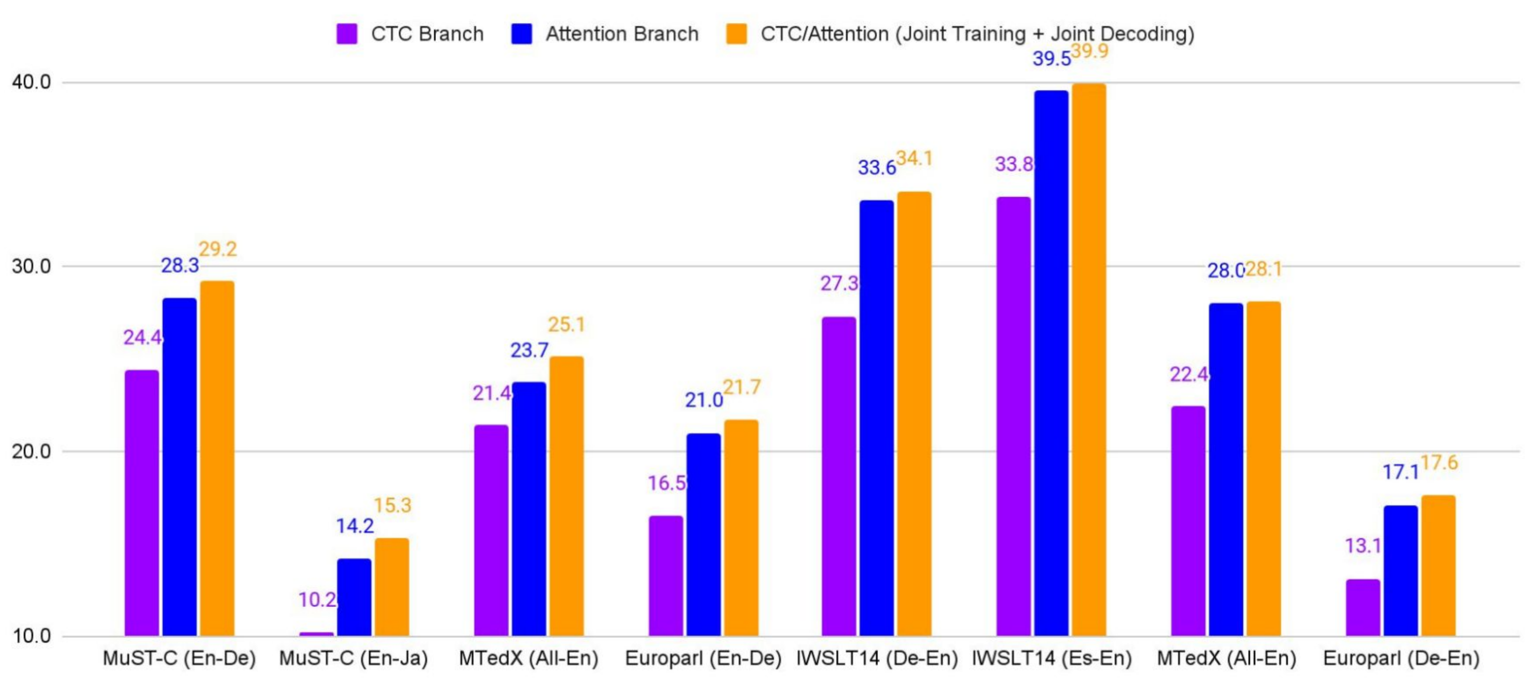
Robust End-Detection

- **Input synchronous emission** + **autoregressive generation**
  - *Conjecture*: input-synchronous emission determines output length based on input length **counteracting the autoregressive end-detection problem during decoding**
  - *Concern*: is the alignment information from CTC translation models reasonable?
  - *Finding*: Joint CTC/attention decoding has low length penalty elasticity
    - Sanity check: input-sync beam search, where CTC plays the
    - Pure-attention models are highly sensitive to length penalty → easily overturned



Positive Ensembling Effect

- **Conditional independence** + **conditional dependence**
  - *Conjecture*: use of conditionally independent likelihoods in joint scoring **eases the exposure/label biases** from conditionally dependent likelihoods **during decoding**
  - *Concern*: does CTC translation quality lag too far behind attention to be useful?
  - *Finding*: even weak CTC models provide a positive ensembling effect
    - Joint decoding yields an average of **+0.7** BLEU improvement over the attention branch
    - For jointly trained models, attention branch outperforms CTC branch by an avg. of **+4.5** BLEU



TLDR

- What did we do?
  - Applied joint CTC/attention to MT/ST with only minor changes from ASR
- Why did we do it?
  - CTC may be weaker for translation, but attention is not perfect either
  - CTC and attention have complementary properties
- How can I try it too?
  - Code and recipes are open-sourced in ESPnet

Carnegie Mellon University Language Technologies Institute

CMU-LTI WAVLab

byan@cs.cmu.edu