CTC Alignments Improve Autoregressive Translation

Brian Yan*[1] **Siddharth Dalmia***[1] **Yosuke Higuchi**[2]
Graham Neubig[1] **Florian Metze**[1] **Alan W Black**[1] **Shinji Watanabe**[1,3]
[1]Language Technologies Institute, Carnegie Mellon University, USA
[2]Department of Communications and Computer Engineering, Waseda University, Japan
[3]Human Language Technology Center of Excellence, Johns Hopkins University, USA
{byan, sdalmia}@cs.cmu.edu

Abstract

Connectionist Temporal Classification (CTC) is a widely used approach for automatic speech recognition (ASR) that performs conditionally independent monotonic alignment. However for translation, CTC exhibits clear limitations due to the contextual and non-monotonic nature of the task and thus lags behind attentional decoder approaches in terms of translation quality. In this work, we argue that CTC *does* in fact make sense for translation if applied in a joint CTC/attention framework wherein CTC's core properties can counteract several key weaknesses of pure-attention models during training and decoding. To validate this conjecture, we modify the Hybrid CTC/Attention model originally proposed for ASR to support text-to-text translation (MT) and speech-to-text translation (ST). Our proposed joint CTC/attention models outperform pure-attention baselines across six benchmark translation tasks.

1 Introduction

Automatic speech recognition (ASR), machine translation (MT), and speech translation (ST) have conspicuous differences but are all closely related sequence-to-sequence problems. Researchers from these respective fields have long recognized the opportunity for cross-pollinating ideas (He and Deng, 2011), starting from the coupling of statistical ASR (Huang et al., 2014) and MT (Al-Onaizan et al., 1999) which gave rise to early approaches for ST (Waibel, 1996; Ney, 1999). Notably in the end-to-end era, attentional encoder-decoder approaches emerged in both MT (Bahdanau et al., 2015) and ASR (Chorowski et al., 2015; Chan et al., 2016), rising to great prominence in both fields.

During this same period, there has been another prominent end-to-end approach in ASR: Connectionist Temporal Classification (CTC) (Graves et al., 2006). Unlike the highly flexible attention mechanism which can handle ASR, MT, and ST alike, CTC models sequence transduction as a monotonic alignment of inputs to outputs and thus fits more naturally with ASR than it does with translation. Still, many interested in non-autoregressive translation have applied CTC to MT (Libovický and Helcl, 2018) and ST (Inaguma et al., 2021b) and promising techniques have emerged, shrinking the gap between autoregressive approaches (Saharia et al., 2020; Gu and Kong, 2021; Chuang et al., 2021; Huang et al., 2022). These recent developments suggest that the latent alignment ability of CTC is a promising direction for translation – this leads us to question: *can CTC alignments improve autoregressive translation?* In particular, we are interested in frameworks that leverage the strength of CTC while minimizing its several harmful incompatibilities (see §3) with translation tasks.

Inspired by the success of Hybrid CTC/Attention in ASR (Watanabe et al., 2017), we investigate jointly modeling CTC with an autoregressive attentional encoder-decoder for translation. Our conjecture is that the monotonic alignment and conditional independence of CTC, which weaken purely CTC-based translation, counteract particular weaknesses of attentional models in joint CTC/attention frameworks. In this work, we seek to investigate *how* each CTC property interacts with corresponding properties of the attentional counterpart during joint training and decoding. We design a joint CTC/attention architecture for translation (§4) and then examine the positive interactions which ultimately result in improved translation quality compared to pure-attention baselines, as demonstrated on the IWSLT (Cettolo et al., 2012), MuST-C (Di Gangi et al., 2019), and MTedX (Salesky et al., 2021) MT/ST corpora (§6).[1]

2 Background: Joint CTC/Attn for ASR

Both the CTC (Graves et al., 2006) and attentional encoder-decoder (Bahdanau et al., 2015) frame-

[1]Models are are available in ESPnet. For ST, refer to `egs2/must_c_v2/st1` and for MT refer to `egs2/iwslt14/mt1`.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1623–1639
May 2-6, 2023

CTC	ATTENTION	JOINT CTC/ATTENTION	ASR	MT/ST
$P_{\text{CTC}}(Y\|X) \triangleq \sum_{Z \in \mathcal{Z}} \prod_{t=1}^{T} P(z_t\|X, \cancel{z_{1:t-1}})$	$P_{\text{Attn}}(Y\|X) \triangleq \prod_{l=1}^{L} P(y_l\|y_{1:l-1}, X)$	$P_{\text{Joint}}(Y\|X) \triangleq P_{\text{CTC}}(Y\|X)^{\lambda} \times P_{\text{Attn}}(Y\|X)^{1-\lambda}$	✓	✓
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	**During Training:** Hard alignment objective produces stable encoder representations allowing the decoder to more rapidly learn soft alignment patterns	✓	**L1** See §3
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	**During Decoding:** Use of conditionally independent likelihoods in joint scoring eases the exposure/label biases from conditionally dependent likelihoods	✓	**L2** See §3
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	**During Decoding:** Input-synchronous emission determines output length based on input length counteracting the autoregressive end-detection problem	✓	**L3** See §3

Table 1: Description of three reasons why joint CTC/attention modeling is powerful in ASR. In order to understand whether these positive interactions between properties of the CTC and attention frameworks are applicable to MT/ST, we must address three corresponding concerns, L1-3, about the applicability of CTC to translation (§2).

works seek to model the Bayesian decision seeking the output, $\hat{Y}$, from all possible sequences, $\mathcal{V}^{\text{tgt}}*$, by selecting the sequence which maximizes the posterior likelihood $P(Y|X)$, where $X = \{\mathbf{x}_t \in \mathcal{S}^{\text{src}} | t = 1, ..., T\}$ and $Y = \{y_l \in \mathcal{V}^{\text{tgt}} | l = 1, ..., L\}$. The source set $\mathcal{S}^{\text{src}}$ is a discrete vocabulary in the MT case and a continuous real space in the ST case while the target set $\mathcal{V}^{\text{tgt}}$ is always a discrete vocabulary. Note that the T-length of the input is assumed to be longer than the L-length output for speech tasks (Graves et al., 2006), but this is not necessarily true for MT.

What are the critical differences between the CTC and attention frameworks? As shown in the first two columns of Table 1, CTC and attention offer different formulations of the posterior likelihood, $P_{\text{CTC}}(\cdot)$ and $P_{\text{Attn}}(\cdot)$ respectively. First of all, the attention mechanism is a flexible input-to-output mapping function which allows a decoder to perform **soft alignment** of an output unit y_l to multiple input units $\mathbf{x}_{[...]}$ without restriction. One downside of this flexibility is a risk of destabilized optimization (Kim et al., 2017). CTC on the other hand marginalizes the likelihoods of all possible input to alignment sequence, $Z = \{z_t \in \mathcal{V}^{\text{tgt}} \cup \{\varnothing\} | t = 1 \dots T\}$, mappings via **hard alignment** where each output unit z_t maps to a single input unit $\mathbf{x}_t$ in a strictly monotonic pattern. $\varnothing$ is a "blank" and Z maps deterministically to Y by removing blanks and repeated emissions.

Secondly, the attentional decoder models each output unit y_1 with **conditional dependence** on not only the input X, but also the previous output units $y_{1:l-1}$. In contrast, CTC makes a **conditional independence** assumption that each z_t does not depend on $z_{1:t-1}$ if already conditioned on X (as denoted by the strike-through in Table 1) – this is a strong assumption which allows for efficient computation of marginalized likelihoods over all $Z \in \mathcal{Z}(Y, T)$ via dynamic programming. On the plus, since CTC does not model causality between output units it is not plagued by the same label and exposure biases that exist in attentional decoders due to local normalization of causal likelihoods (Bottou, 1991; Ranzato et al., 2016; Hannun, 2019).

Finally, the attentional decoder is an **autoregressive generator** that decodes the output until a stop token, <eos>. Comparing likelihoods for sequences of different lengths requires a heuristic brevity penalty. Furthermore label bias with respect to the stop token manifests as a length problem where likelihoods degenerate for unexpectedly long outputs (Murray and Chiang, 2018). In comparison, CTC is an **input-synchronous emitter** that consumes an input unit in order to produce an output unit. Therefore, CTC cannot produce an output longer than the input representation which feeds the final posterior output layer – but this also means that CTC does not require end detection.

As previously shown by (Kim et al., 2017; Watanabe et al., 2017), jointly modeling CTC and an attentional decoder is highly effective in ASR. The foundation of this architecture is a shared encoder, ENC, which feeds into both CTC, $P_{\text{CTC}}(\cdot)$, and attentional decoder, $P_{\text{Attn}}(\cdot)$, posteriors:

$$\mathbf{h} = \text{Enc}(X) \quad (1)$$

$$P_{\text{CTC}}(z_t|X) = \text{CTC}(\mathbf{h}_t) \quad (2)$$

$$P_{\text{Attn}}(y_l|X, y_{1:l-1}) = \text{Dec}(\mathbf{h}, y_{1:l-1}) \quad (3)$$

where $\text{CTC}(\cdot)$ denotes a linear projection to the CTC output vocabulary, $\mathcal{V}^{\text{tgt}} \cup \{\varnothing\}$, followed by softmax. $\text{DEC}(\cdot)$ denotes autoregressive decoder

layers followed by a linear projection to the decoder output vocabulary, $\mathcal{V}^{\text{tgt}} \cup \{\text{<eos>}\}$, and softmax. The joint network is optimized via a multi-tasked objective, $\mathcal{L}^{\text{ASR}} = \mathcal{L}^{\text{ASR}}_{\text{CTC}} + \lambda \mathcal{L}^{\text{ASR}}_{\text{Attn}}$, where λ interpolates the CTC and decoder cross-entropy losses.

Joint decoding is typically performed with a one-pass beam search where CTC plays a secondary role as a joint scorer while attention leads the major hypothesis expansion and end detection functions in the algorithm (Watanabe et al., 2017; Tsunoo et al., 2021). However, CTC is capable of taking over the lead role if called upon (e.g. for streaming applications) (Moritz et al., 2019).

3 Potential CTC Limitations in MT/ST

Why exactly does this joint CTC/attention framework perform so well in ASR? As summarized in column 3 of Table 1, we are particularly interested in three reasons which arise from the combination of the hard vs. soft alignment, conditional independence vs. dependence, and input-synchronous emission vs. autoregressive generation properties of CTC and attention respectively. These dynamics have become well understood in ASR, owing to the popularity of the joint framework (Watanabe et al., 2018) amongst ASR practitioners.

So can CTC and attention also complement each other when applied jointly to translation?[2] ASR, MT, and ST can all be generalized as sequence transduction tasks following the Bayesian formulation. Attentional decoders have been a predominant technical solution to each of these tasks. However, the CTC framework appears to have several limitations specific to MT/ST that are not present in ASR; this seemingly diminishes the promise of the joint CTC/attention framework for translation. In this work, we seek to address the following three concerns about MT/ST CTC which appear to inhibit the CTC/attention framework (please refer back to Table 1 as needed).

L1 *Can CTC encoders perform sophisticated input-to-output mappings required for translation?*

Unlike ASR, translation entails non-monotonic mappings due to variable word-ordering across languages. Additionally, inputs may be shorter than outputs as mappings are not necessarily one-to-one. Furthermore, the mapping task for ST is compositional where logically a speech signal first maps to a source language transcription before being mapped to the ultimate translation. All of these complications appear to directly contradict the **hard alignment** of CTC. If CTC cannot produce stable encoder representations for MT/ST, then during joint training attention does not receive the optimization benefit as in ASR (per row 2 of Table 1). Fortunately, prior works suggest that these challenges are not insurmountable. Chuang et al. (2021) showed that self-attentional encoders can perform latent model variable word orders for ST, Libovický and Helcl (2018); Dalmia et al. (2022) proposed upsampling encoders that produce expanded input representations for MT, and Sanabria and Metze (2018); Higuchi et al. (2022) proposed hierarchical CTC encoders that can compose multiple output resolutions for ASR. In §4.1, we incorporate these techniques into a unified hierarchical CTC encoding method for MT/ST which is capable of sophisticated input-to-output mappings.

L2 *Does CTC-based translation quality lag too far behind attention-based to be useful?*

CTC-based ASR has recently shown competitive performance due in large part to improved neural architectures (Gulati et al., 2020) and self-supervised learning (Baevski et al., 2020; Hsu et al., 2021), but the gap between CTC and attention for translation appears to be greater (Gu and Kong, 2021). Perhaps the **conditional independence** of CTC inhibits the quality to such a degree in MT/ST where these likelihoods cannot ease the label/exposure biases of the attentional decoder as they do in ASR (per row 3 of Table 1). The relative weakness of non-autoregressive translation approaches has been well-studied. Knowledge distillation (Kim and Rush, 2016; Zhou et al., 2019) and iterative methods (Qian et al., 2021; Chan et al., 2020; Huang et al., 2022) all attempt to bridge the gap between non-autoregressive models and their autoregressive counterparts. In §6, we address this concern empirically; we find that even CTC models with 28% relative BLEU reduction compared to attention yield improvements when CTC and attention are jointly decoded.

L3 *Is the alignment information produced by CTC-based translation models reasonable?*

In ASR, CTC alignments are reliable enough to segment audio data by force aligning inputs to target transcription outputs (Kürzinger et al., 2020) and exhibit minimal drift compared to hidden Markov models (Sak et al., 2015). However,

[2] This particular question has not been addressed in literature. For an account of related works, please see §8.

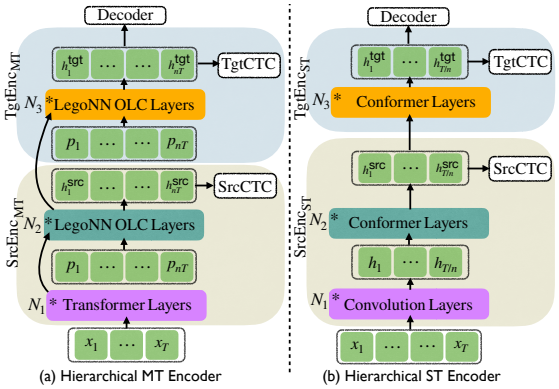

Figure 1: Hierarchical MT/ST encoders where representations are first up/down-sampled by $\text{SRCENC}_{\text{MT/ST}}$ and then re-ordered by $\text{TGTENC}_{\text{MT/ST}}$.

CTC alignments are not as well studied in translation. It is an open question of whether or not the **input-synchronous emission** of CTC for translation has sufficient alignment quality to support the end detection responsibility during joint decoding as it does in ASR (per row 4 of Table 1). Ideally, the CTC alignments are strong enough such that CTC can lead joint decoding by proposing candidates for hypothesis expansion in each beam step until all input units are consumed (at which point the end is detected), as in an input-synchronous beam search. More conservatively, the CTC alignments may be too unreliable to take the lead but could still guide the attentional decoder's end detection by penalizing incorrect lengths via joint scoring, as in an output-synchronous beam search. In §4.2, we lay out comparable forms for input and output-synchronous beam search which allows us to examine the impact on translation quality depending on whether CTC is explicitly responsible for or only implicitly contributing to end detection.

4 Joint CTC/Attention for Translation

4.1 Hierarchical CTC Encoding

Per L1 described in §3, we seek to build a CTC encoder for translation which handles sophisticated input-to-output mappings. Unlike ASR where outputs are assumed to be 1) always shorter than inputs and 2) monotonic with respect to inputs, translation needs to account for variability of lengths and and word orderings. We therefore propose to use a hierarchical CTC encoding scheme which 1) aligns inputs to length-adjusted *source*-oriented encodings before 2) aligning to re-ordered *target*-oriented encodings, as shown in Figure 1. Our encoding process thus consists of two compartmentalized functions: length-adjustment and re-ordering.

Length-adjustment For MT, we up-sample the lengths of the source-oriented encodings in order to output sequences longer than the input. For ST, we down-sample the lengths of the source-oriented encodings to coerce a discrete textual representation of the real-valued speech input. We enforce source orientations using CTC criteria that seek to align the length adjusted intermediate layer encoder representations towards source text sequences (for MT this is the same as the input and for ST this is the ASR target). By compartmentalizing length-adjustment within this initial stage, we allow subsequent encoder layers to focus solely on re-ordering.

Re-ordering We then obtain target-oriented encodings with subsequent encoder layers, where re-ordering is enforced using CTC criteria that seek to align final layer encoder representations towards target text sequences. Critically, the underlying neural network architecture must be able to model latent re-ordering as the CTC criterion itself will only consider monotonic alignments of the final encoder representation to the target.

Our proposed MT/ST hierarchical encoders consist of the following components:

$$\mathbf{h}^{\text{SRC}} = \text{SRCENC}_{\text{MT/ST}}(X) \quad (4)$$

$$P_{\text{CTC}}(z_t^{\text{SRC}}|X) = \text{SRCCTC}_{\text{MT/ST}}(\mathbf{h}_t^{\text{SRC}}) \quad (5)$$

$$\mathbf{h}^{\text{TGT}} = \text{TGTENC}_{\text{MT/ST}}(\mathbf{h}^{\text{SRC}}) \quad (6)$$

$$P_{\text{CTC}}(z_t^{\text{TGT}}|X) = \text{TGTCTC}_{\text{MT/ST}}(\mathbf{h}_t^{\text{TGT}}) \quad (7)$$

The hierarchical encoders are jointly optimized with an attentional decoder using a multi-tasked objective, $\mathcal{L} = \mathcal{L}_{\text{SRCCTC}} + \lambda_1 \mathcal{L}_{\text{TGTCTC}} + \lambda_2 \mathcal{L}_{\text{ATTN}}$, where λ's interpolate source-oriented CTC, target-oriented CTC, and decoder cross-entropy losses.

As shown in Figure 1.a, $\text{SRCENC}_{\text{MT}}(\cdot)$ consists of N_1 Transformer (Vaswani et al., 2017) layers followed by N_2 up-sampling Output Length Controller (OLC) layers used in LegoNN (Dalmia et al., 2022) – the layer-wise positional embeddings of the OLC architecture enable latent length-adjustment of textual inputs. $\text{TGTENC}_{\text{MT}}(\cdot)$ consists of N_3 non-up-sampling OLC layers – the layer-wise attention of the OLC architecture enables latent re-ordering. Our ST encoder is similar, but uses Conformers (Gulati et al., 2020) to capture the local and global dependencies in speech, as shown in Figure 1.b. $\text{SRCENC}_{\text{ST}}(\cdot)$ consists of N_1 convolutional blocks (Dong et al., 2018) for down-sampling fol-

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

```
1:  procedure OutputStep(prtHs, X, l, p, maxL)
2:      newPrtHs = {}; endHs = {}
3:      for y_{1:l-1} ∈ prtHs do
4:          attnCnds = top-k(P_Attn(y_l|X, y_{1:l-1}), k = p)
5:          for c ∈ attnCnds do
6:              y_{1:l} = y_{1:l-1} ⊕ c                          ← Hypothesis Expansion
7:              α_CTC = CTCScore(y_{1:l}, X_{1:T})
8:              α_Attn = AttnScore(y_{1:l}, X_{1:T})             ← Joint Scoring
9:              β = LengthPen(y_{1:l})
10:             P_Beam(y_{1:l}|X) = α_CTC + α_Attn + β
11:             if (c is <eos>) or (l is maxL) then              ← End Detection
12:                 endHs[y_{1:l}] = P_Beam(·)
13:             else
14:                 newPrtHs[y_{1:l}] = P_Beam(·)
15:             end if
16:         end for
17:     end for
18:     return newPrtHs, endHs
19: end procedure
```

Algorithm 2 *Input*-Synchronous Step Function: CTC proposes candidates to expand hypotheses which are all produced from t input units at step t.

```
1:  procedure InputStep(prtHs, X, t, p, T)
2:      newPrtHs = {}; endHs = {}
3:      CTCCnds = top-k(P_CTC(z_t|X), k = p)
4:      for y ∈ prtHs do
5:          for c ∈ CTCCnds do
6:              if (c is ∅) or (c is repeat) then
7:                  ỹ = y                                        ← Hypothesis Expansion
8:              else
9:                  ỹ = y ⊕ c
10:             end if
11:             α_CTC = CTCScore(ỹ, X_{1:t})
12:             α_Attn = AttnScore(ỹ, X_{1:T})                   ← Joint Scoring
13:             β = LengthPen(ỹ)
14:             P_Beam(ỹ|X) = α_CTC + α_Attn + β
15:             if t is T then                                   ← End Detection
16:                 endHs[ỹ] = P_Beam(·)
17:             else
18:                 newPrtHs[ỹ] = P_Beam(·)
19:             end if
20:         end for
21:     end for
22:     return newPrtHs, endHs
23: end procedure
```

lowed by N_2 Conformer layers – this stage is analogous to the ASR sub-task of ST where a long speech signal is length-adjusted to a shorter latent textual representation. $\text{TGTENC}_{\text{ST}}(\cdot)$ consists of N_3 Conformer layers – this stage is analogous to the MT sub-task of ST where latent re-ordering is enabled by self-attention. LegoNN and Conformer are further described in §A.4.

4.2 Input/Output-Synchronous Decoding

Per L2 and L3 described in §3, we seek to design a joint decoding algorithm with input and output-synchronous variants of one-pass beam search which differ only in whether CTC or attention takes the leading role. As shown in Algorithms 1 and 2, we propose to align the input and output beam-step functions along three common functions: hypothesis expansion, joint scoring, and end detection. Using these mirrored forms, let us now interpret the respective roles of CTC and attention.

Output-Synchrony Consider first that attention is in the leading role, which means that we are working with an output-synchronous beam search. Note that this is the algorithm originally proposed by Hori et al. (2017). OutputStep performs *hypothesis expansion* by computing the attentional decoder's output posterior at label step l, $P_{\text{Attn}}(y_l|X, y_{1:l-1})$ for each partial hypothesis, $y_{1:l-1}$. A pre-beam size, p, is then used to select the top candidate output units (Seki et al., 2019), attnCnds, which are used to expand the partial hypotheses via concatenation, denoted by $\oplus$. In the *joint scoring* block, the attentional decoder likelihood, $\text{AttnScore}(\cdot)$, and length penalty/reward, $\text{LengthPen}(\cdot)$ yield the estimated joint likelihood P_{Beam}. Finally in *end detection*, OutputStep must check for the stop token, <eos>, which may be proposed by attnCnds.

Input-Synchrony Now let us consider the differences when CTC is in the leading role. Note that this algorithm extends Hannun et al. (2014)'s CTC beam search algorithm to include joint scoring with attentional likelihoods. InputStep performs *hypothesis expansion* by computing CTC's alignment posterior at time step t, $P_{\text{CTC}}(z_t|X)$. Unlike in output-synchrony, here each hypothesis expansion also consumes one step of the input. The same pre-beam size, p, is used to select top candidate alignment units, CTCCnds, but partial hypotheses are only expanded for non-blank and non-repeat candidates. The *joint scoring* block is identical to output-synchrony except for one difference: CTC likelihood, $\text{CTCScore}(\cdot)$, is applied over the full input, $X_{1:T}$, in OutputStep and over the partial input, $X_{1:t}$, in InputStep. This difference engenders a speed vs. accuracy trade-off, which we discuss in D2 of §7. Finally, *end detection* simply occurs when all input units have been consumed ($t = T$). Therefore, InputStep does not require checking for the stop token as all hypotheses at time T are ended.

We propose this particular form of input-

#	Model Name	Model Type			MT			ST		
		Joint Train?	Joint Decode?	Decoding Method	IWSLT14 De-En	IWSLT14 Es-En	MTedX All-En	MuST-C-v2 En-De	MuST-C-v2 En-Ja	MTedX All-En
1	Pure-Attn (Prior)	✗	✗	Attn Only	(32.2)†	(39.0)†	-◇	25.8‡	12.4‡	-◇
2	Pure-Attn (Ours)	✗	✗	Attn Only	32.8 (33.7)	39.0 (39.9)	25.6	27.8	14.3	22.7
3	Joint CTC/Attn	✓	✗	CTC Only	27.3	33.8	22.4	24.4	10.2	21.4
4	Joint CTC/Attn	✓	✗	Attn Only	33.6	39.5	28.0	28.3	14.2	23.7
5	Joint CTC/Attn	✓	✓	Joint I-Sync	33.7	39.7	27.8	**29.2**	15.1	**25.1**
6	Joint CTC/Attn	✓	✓	Joint O-Sync	**34.1**	**39.9**	**28.1**	**29.2**	**15.3**	**25.1**

Table 2: Test set performances, as measured by BLEU (↑), of our proposed joint CTC/Attention models compared to pure-attention baselines. Joint CTC/Attention models are always jointly trained, but can be either jointly decoded using input/output synchrony or decoded using only their CTC or attention branches. For IWSLT14, we mention (*tokenized BLEU*) for comparison with prior works: †Raunak et al. (2020) and ‡Inaguma et al. (2020). ◇Prior MTedX works show only All-All or pair-wise settings.

synchronous beam search in order to exactly mirror the functions of its output-synchronous counterpart; without this mirrored formulation, we cannot attribute differences in decodings to the swapped roles of CTC and attention. For instance, now we can answer questions such as *can CTC perform hypothesis expansion on par with attention*, allowing us to address concerns about applying weaker joint CTC models during decoding per L2 and L3 in §3. To the best of our knowledge, we are the first to examine the theoretical and empirical differences of input and output synchrony through a unified formulation, as discussed further in §7. Other forms of input-synchronous beam search in prior works cannot directly be used for this purpose. Triggered Attention (Moritz et al., 2019) is one such example which is purpose-fit for streaming to a degree where several core components (e.g. look-ahead and re-triggering) cannot trivially be re-factored into an output-synchronous variant.

5 Experimental Setup

Data We examine the efficacy of our proposed approaches on two language pairs for each of the MT and ST tasks. For MT, we use German-to-English (De-En) and Spanish-to-English (Es-En) from IWSLT14 (Cettolo et al., 2012). For ST, we use English-to-German (En-De) and English-to-Japanese (En-Ja) from MuST-C-v2, reporting tst-COMMON results (Di Gangi et al., 2019). We also examine the multilingual setting of 6 European languages to English (All-En) from MTedX (Salesky et al., 2021) for both tasks. Full dataset descriptions for reproducibility are in §B.

Modeling We compare our joint CTC/Attention models to purely attentional encoder-decoder baselines. All proposed and baseline models were tuned separately, using validation sets only, within the same hyperparameter search spaces for training and decoding to ensure fair comparison. All experiments were conducted using ESPnet-ST (Inaguma et al., 2020). Full descriptions of model sizes, hyperparameters, and pre-processing are in §B.[3]

Evaluation: Unless otherwise indicated, we measure performance with detokenized case-sensitive BLEU (Post, 2018) on punctuated 1-references.[4]

6 Results and Analyses

In this section, we first present our main results on 6 benchmark MT and ST tasks. We then present evidence that hierarchical encoding (§4.1) produces stable encoder representations that simplify the decoder's source attention (addressing L1 in §3). Next we present evidence that joint decoding is beneficial despite the fact that CTC-only performance lags behind that of attention-only (addressing L2 in §3). Finally, we present evidence that CTC's alignment information alleviates attention's end-detection problem in both input and output synchronous joint decoding (§4.2) (addressing L3 in §3).

6.1 Joint CTC/Attention Models Outperform CTC-only and Attention-only Baselines

As shown in Table 2, joint CTC/Attention with output-synchronous decoding outperforms pure-

[3]We compare our baselines for MuST-C-v2 to the default recipes in ESPnet in Table 2. For back-compatibility with additional prior works using MuST-C-v1 En-De, see §A.2.

[4]Evaluation with additional metrics is provided in §A.1.

attention across *all* MT and ST tasks (line 2 vs. 6). Joint training while only decoding with the attention branch still outperforms pure-attention models without any joint training (line 2 vs. 4). Note that *CTC is consistently the weaker* of the two branches in jointly trained models (line 3 vs. 4). Joint input/output-synchronous decodings yield further improvements overall, confirming that *both joint training and decoding are beneficial* (line 4 vs. 5/6). However, we find that input-synchrony lags behind output-synchrony (line 5 vs. 6); this phenomenon is discussed further in §7.

6.2 Hierarchical Encoding Reduces Attention's Alignment Burden

We examine the regularization effect that CTC joint training has on the attentional decoder, per L1 in §3, by first quantifying the monotonicity, m of a (L, T) shaped source attention pattern, A:

$$m = \Big(\sum_{2<l\leq L} [\operatorname*{argmax}_{t\in T} A_l \geq \operatorname*{argmax}_{t\in T} A_{l-1}] \Big) / L$$

where $[\cdot]$ denotes the Iverson bracket. In other words, we define monotonicity m as the rate at which the decoder at step l attends most sharply on an input index, $\operatorname{argmax}_{t\in T} A_l$, which is greater than or equal to that of the previous step $l-1$, $\operatorname{argmax}_{t\in T} A_{l-1}$. We compute m over all examples in our validation sets for De-En MT and En-De ST and show the layer-wise averages over all examples and attention heads in Figure 2. It can be seen that *the decoder source attention patterns are more monotonic* when using jointly trained hierarchical encoders. Per line 2 of Table 1, we argue that this greater monotonicity allows the decoder to more rapidly learn soft alignment patterns – ultimately this advantage is reflected in the overall performance gains observed from joint training without joint decoding (line 2 vs. 4 in Table 2).

For a qualitative example illustrating the increased monotonicity of decoder source attention patterns, please see §A.7. We also found that *increased monotonicity leads improves multilingual parameter sharing* in our All-En MT and ST models, suggesting that the target-orientation of our encoder reduced the decoder's burden of soft-aligning target English outputs to source languages with varying word-orders (discussed further in §A.5).

What are the respective contributions of SRCCTC *and* TGTCTC*?* TGTCTC holds elevated importance as joint decoding is not possible without it.

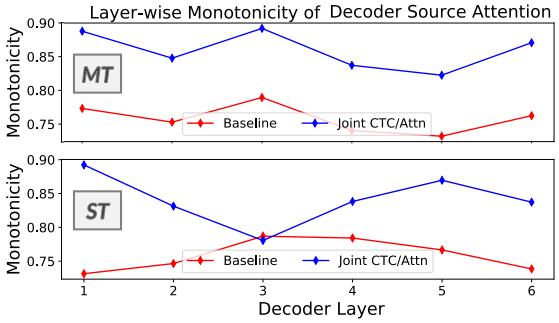

Figure 2: Layer-wise monotonicity (↑) of the source-attention patterns produced by MT/ST decoders.

		MT (DE-EN)	ST (EN-DE)
SRCCTC	TGTCTC	IWSLT14	MuST-C-v2
✗	✗	32.1	27.7
✓	✗	34.1	27.8
✗	✓	33.3	28.1
✓	✓	**34.8**	**28.3**

Table 3: Ablation on the impacts of SRCCTC and TGTCTC CTC components of hierarachical encoding, as measured by performance on validation sets. Only attention is used in decoding to enable fair comparisons.

However, we'd like to understand how each component contributes to the benefits observed from joint training without joint decoding in §6.1. In Table 3, we ablate SRCCTC and TGTCTC in order to confirm that both contribute to performance gains. Note that SRCCTC on its own appears to contribute more to MT than it does to ST, suggesting that the length adjustment stage is more critical in MT.

6.3 Even Weak CTC Models Strengthen Joint CTC/Attention Models

We examine the generalization effect that augmenting autoregressive likelihoods with conditionally independent likelihoods has during inference, per L2 in §3, by evaluating De-En MT and En-De ST models on out-of-domain EuroParl test sets (Iranzo-Sánchez et al., 2020). As shown in Table 4, joint CTC/Attention models outperform pure-attention baselines across in-domain (In-D) and out-of-domain (Out-D) settings. When decoding only the CTC branch of joint models (denoted as CTC I-Sync in the table) performance is significantly degraded compared to the attention branch of the same models (denoted as Attn O-Sync in the table). This gap appears slightly lessened in the out-of-domain setting where CTC's conditional independence may offer some robustness. Nonetheless

Model	Decoding Method	MT (De-En) In-D	MT (De-En) Out-D	ST (En-De) In-D	ST (En-De) Out-D
Pure-Attn	Attn Only	32.8	15.8	27.8	20.5
Joint C/A	Attn Only	33.6	17.1	28.3	21.0
	+CTC Rescore	33.6	17.1	28.3	21.0
Joint C/A	Joint O-Sync	**34.1**	**17.6**	**29.2**	**21.7**
Joint C/A	CTC Only	27.3	13.1	24.4	16.5
	+Attn Rescore	29.5	13.9	26.2	17.8
Joint C/A	Joint I-Sync	**33.7**	**17.4**	**29.2**	**21.1**

Table 4: In/out domain test performances of joint CTC/attention models with various decoding methods.

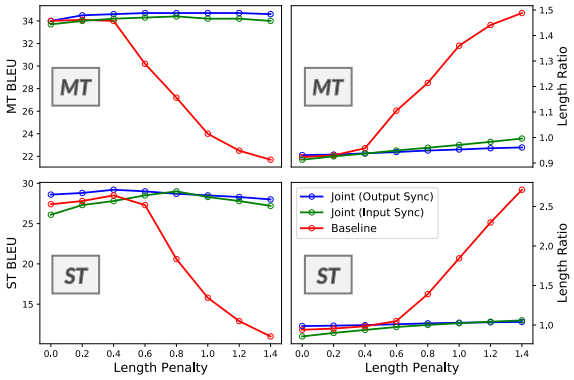

Figure 3: Elasticity of BLEU and length ratios (|hyp|/|ref|) w.r.t length penalty in validation sets.

these weak CTC models still boost their stronger attention counterparts during joint decoding (both via input and output-synchrony), suggesting that *ensembling of conditionally independent and dependent likelihoods is a powerful technique*.

Further, synchronous joint decoding methods outperform their two-pass re-scoring counterparts (discussed in D2 of §7), suggesting that *joint selection of the hypothesis set is necessary* for easing the respective weaknesses of autoregressive and conditionally independent likelihood estimation.

6.4 CTC's Alignment Information Resolves Attention's End-Detection Problem

Finally, we examine the effect that CTC's alignment information has on end detection during decoding, per L3 in §3. In Figure 3, we observe the change in translation quality (as measured by BLEU) and output length (as measured by hypothesis-to-reference length ratio) when the length penalty (denoted as $\mathrm{LengthPen}(\cdot)$ in Algorithms 1 and 2) is gradually increased, forcing decodings to produce longer outputs. Pure-attention baselines rapidly degenerate when forced to produce hypotheses that are longer than references as they struggle to detect the ends of hypotheses (Murray and Chiang, 2018). On the other hand, joint decoding produces gradually longer outputs regardless of whether CTC is in a primary role (input-synchrony) or a secondary role (output-synchrony), demonstrating that *CTC alignments ease the decoder's end-detection problem* by explicitly or implicitly ruling out hypotheses of incorrect lengths.

7 Discussion: More on Joint Decoding

D1 *Why do input vs. output-synchronous joint decodings yield slightly different results?*

By comparing the CTC likelihood estimation in InputStep vs. OutputStep, it can be seen that there is a trade-off between speed vs. accuracy. First, note that in OutputStep, $\mathrm{CTCScore}(y_{1:l}, X_{1:T})$, is a marginalization over the likelihoods of all possible alignments of the partial hypothesis $y_{1:l}$ to the full input $X_{1:T}$ (Seki et al., 2019). On the other hand, $\mathrm{CTCScore}(\tilde{y}, X_{1:t})$ in InputStep is an estimation of the marginalized likelihoods of the partial hypothesis $y_{1:l}$ to the *partial* input $X_{1:t}$ (Graves, 2012; Hannun et al., 2014). Even at step T, these two $\mathrm{CTCScore}(\cdot)$'s are not equivalent. Since $\mathrm{CTCCnds}$ may include the blank token, InputStep may prune partial hypotheses at a previous beam step which would have merged with $y_{1:l}$. Therefore, $\mathrm{CTCScore}(\cdot)$ in *input-synchrony is less accurate*. However, *input-synchrony requires fewer computations*. Using dynamic programming, output-synchrony computes $\mathrm{CTCScore}(\cdot)$ for all partial hypothesis within a single beam step with $\mathcal{O}(bpT)$ log-additions (Watanabe et al., 2017) while input-synchrony uses only $\mathcal{O}(bp)$ log-additions (Hannun et al., 2014).

Decoding Type		Accuracy		Speed
Method	Beam Size	BLEU	Search Error	RTF
Joint O-Sync	5	29.1	0.73%	0.9
Joint O-Sync	10	29.2	0.44%	1.7
Joint O-Sync	50	29.0	0.36%	9.0
Joint I-Sync	5	28.1	1.02%	0.4
Joint I-Sync	10	28.6	1.09%	0.9
Joint I-Sync	50	29.0	0.87%	6.4

Table 5: Speed vs. accuracy for joint input/output-sync decoding of En-De ST val. set as a fxn. of beam size.

In Table 5, we perform an experimental validation of our theoretical understanding of the speed vs. accuracy trade-off between the two synchronous joint decoding variants. To quantify

speed, we compute the real-time factor (RTF) as the ratio of decoding time over the duration of input speech. To quantify accuracy beyond the BLEU metric, we compute the search error rate (Meister et al., 2020) by counting the sequences for which the hypothesis has higher exact likelihood than the reference. *For the same beam size, output is slower but more accurate than input-synchronous.* We conclude that input-synchrony may in fact be preferable in applications with latency constraints.

D2 *Why did synchronous joint decodings outperform re-scoring decodings in Table 4?*

There is a family of two-pass decoding algorithms (Watanabe et al., 2017; Sainath et al., 2019), which also achieve joint decoding by first estimating the likelihoods of a subset of sequences $\mathcal{V}'$ with one module and then re-scoring the estimates with the other module. In these approaches, the subset $\mathcal{V}'$ is determined asynchronously, meaning the joint likelihood is not considered until the re-scoring step; this delayed consideration of the joint likelihood is the main drawback compared to the synchronous approaches. If the attentional decoder is used to determine $\mathcal{V}'$, then $\mathcal{V}'$ would suffer from exposure/label bias and the length problem (§2). On the other hand, if CTC is used to determine $\mathcal{V}'$, the lack of causal modeling in CTC leads to poor estimates of $\mathcal{V}'$ – particularly for translation.

8 Related Works

The idea of using latent alignments to improve autoregressive translation has been explored previously by Haviv et al. (2021) who concluded that CTC alignments are not compatible with teacher forcing. The key difference is that we train CTC and autoregressive models jointly while Haviv et al. (2021) sought to apply CTC to train autoregressive models, replacing cross-entropy entirely. More recently in a concurrent work, Zhang et al. (2022) have also shown the effectiveness of jointly training CTC and attention in the context of ST for unwritten languages where no ASR transcriptions are available. We believe that our contribution showing the effectiveness of also jointly decoding CTC and attention demonstrates an additional technique which can further improve their direction. Our work also differs in that we seek to incorporate the ASR objective into ST via hierarchical encoding.

Other concurrent works integrated CTC and attention within blockwise streaming (Deng et al., 2022) and compositional multi-decoder (Yan et al., 2022) architectures for ST in particular. Our work supports their findings by addressing *why* CTC is helping, and we provide a unified approach that generalizes to both MT and ST. Prior works have also used the non-autoregressive property of CTC as means for speeding up autoregressive models during inference (Inaguma et al., 2021a; Gaido et al., 2021), but these works focus on latency and do not apply CTC to improve translation quality.

Decoding Type		Speed	
Method	Beam Size	RTF	%Δ
Pure-Attn O-Sync	5	0.9	-
Pure-Attn O-Sync	10	1.2	-
Pure-Attn O-Sync	50	3.5	-
Joint CTC/Attn O-Sync	5	0.9	+0%
Joint CTC/Attn O-Sync	10	1.7	+42%
Joint CTC/Attn O-Sync	50	9.0	+157%
Joint CTC/Attn I-Sync	5	0.4	-56%
Joint CTC/Attn I-Sync	10	0.9	-25%
Joint CTC/Attn I-Sync	50	6.4	+85%

Table 6: **Limitations Table:** comparison of joint decoding and pure-attention RTFs across different beam sizes. %Δ between the joint RTF and pure-attention RTF for the same beam size is shown, where positive %'s indicate slow-downs and negative %'s indicate speed-ups.

9 Conclusion

We propose to jointly train and decode CTC/attention models for MT and ST using 1) hierarchical encoding to resolve incompatibilities between CTC and the non-monotonic mappings in translation and 2) synchronous decoding to ease the exposure/label biases of autoregressive decoders with CTC's conditionally independent alignment information. Our analyses reveal several reasons why even weak CTC models benefit autoregressive translation via joint modeling, suggesting that future explorations into jointly modeling attentional decoders with other latent alignment models (Graves, 2012; Ghazvininejad et al., 2020; Saharia et al., 2020) may uncover similar benefits.

Limitations

There are several potential limitations pertaining to the increased computational overhead and latency of the joint modeling approach. One concern is that joint decoding is much slower, but

we found that input-synchronous joint decoding is actually faster than pure-attention decoding for smaller beam sizes, as shown in Table 6.

The other limitation is that our MT models up-sample input representations in the early layers of the encoder, thereby increasing the computations in subsequent encoder layers and the decoder's cross-attention. We can use LegoNN-based encoders (Dalmia et al., 2022) to adjust the up-sampling rate to a fractional value, minimizing the computations given dataset statistics. Alternatively, we may avoid the need for up-sampling by applying a larger byte-pair encoding size (Kudo and Richardson, 2018) to the target language compared to the source language. CTC's use in guiding efficient down-sampling of representations in ST (Gaido et al., 2021) suggest that it may also be applied for efficient up-sampling for MT – we leave this study on efficiency to future work.

Finally, note that the corpora used for the MT experiments in this work are considered medium resourced. Prior work (Murray and Chiang, 2018) has shown that the autoregressive end-detection problem exists across low to high resourced scenarios; suggesting that the CTC/attention approach would be generally beneficial. We leave the study of joint CTC/attention modeling on higher resourced MT corpora to future work.

Acknowledgements

Brian Yan and Shinji Watanabe are supported by the Human Language Technology Center of Excellence. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns et al., 2014), which is supported by National Science Foundation grant number ACI-1548562; specifically, the Bridges system (Nystrom et al., 2015), as part of project cis210027p, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center.

References

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah A Smith, and David Yarowsky. 1999. Statistical machine translation. In *Final Report, JHU Summer Workshop*, volume 30.

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Léon Bottou. 1991. *Une Approche théorique de l'Apprentissage Connexionniste: Applications à la Reconnaissance de la Parole*. Ph.D. thesis, Université de Paris XI, Orsay, France.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit3: Web inventory of transcribed and translated talks. In *Conference of european association for machine translation*, pages 261–268.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

William Chan, Chitwan Saharia, Geoffrey Hinton, Mohammad Norouzi, and Navdeep Jaitly. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *International Conference on Machine Learning*, pages 1403–1413. PMLR.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28.

Shun-Po Chuang, Yung-Sung Chuang, Chih-Chiang Chang, and Hung-yi Lee. 2021. Investigating the reordering capability in CTC-based non-autoregressive end-to-end speech translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1068–1077, Online. Association for Computational Linguistics.

Siddharth Dalmia, Dmytro Okhonko, Mike Lewis, Sergey Edunov, Shinji Watanabe, Florian Metze, Luke Zettlemoyer, and Abdelrahman Mohamed. 2022. LegoNN: Building modular encoder-decoder models. *arXiv preprint arXiv:2206.03318*.

Siddharth Dalmia, Brian Yan, Vikas Raunak, Florian Metze, and Shinji Watanabe. 2021. Searchable hidden intermediates for end-to-end models of decomposable sequence tasks. In *Proceedings of the 2021*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1882–1896.

Keqi Deng, Shinji Watanabe, Jiatong Shi, and Siddhant Arora. 2022. Blockwise Streaming Transformer for Spoken Language Understanding and Simultaneous Speech Translation. In *Proc. Interspeech 2022*, pages 1746–1750.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Linhao Dong, Shuang Xu, and Bo Xu. 2018. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888. IEEE.

Yichao Du, Zhirui Zhang, Weizhi Wang, Boxing Chen, Jun Xie, and Tong Xu. 2022. Regularizing end-to-end speech translation with triangular decomposition agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10590–10598.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2021. CTC-based compression for direct speech translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 690–696.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. In *International Conference on Machine Learning*, pages 3515–3523. PMLR.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Jiatao Gu and Xiang Kong. 2021. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 120–133.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *Proc. Interspeech 2020*, pages 5036–5040.

Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, et al. 2021. Recent developments on espnet toolkit boosted by conformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5874–5878. IEEE.

Awni Hannun. 2019. The Label Bias Problem. `https://awni.github.io/label-bias`.

Awni Y Hannun, Andrew L Maas, Daniel Jurafsky, and Andrew Y Ng. 2014. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. *arXiv preprint arXiv:1408.2873*.

Adi Haviv, Lior Vassertail, and Omer Levy. 2021. Can latent alignments improve autoregressive machine translation? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2637–2641, Online. Association for Computational Linguistics.

Xiaodong He and Li Deng. 2011. Speech recognition, machine translation, and speech translation—a unified discriminative learning paradigm [lecture notes]. *IEEE Signal Processing Magazine*, 28(5):126–133.

Yosuke Higuchi, Keita Karube, Tetsuji Ogawa, and Tetsunori Kobayashi. 2022. Hierarchical conditional end-to-end ASR with CTC and multi-granular subword units. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7797–7801. IEEE.

Takaaki Hori, Shinji Watanabe, and John Hershey. 2017. Joint CTC/attention decoding for end-to-end speech recognition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 518–529, Vancouver, Canada. Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Chenyang Huang, Hao Zhou, Osmar R Zaïane, Lili Mou, and Lei Li. 2022. Non-autoregressive translation with layer-wise prediction and deep supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10776–10784.

Xuedong Huang, James Baker, and Raj Reddy. 2014. A historical perspective of speech recognition. *Communications of the ACM*, 57(1):94–103.

Hirofumi Inaguma, Siddharth Dalmia, Brian Yan, and Shinji Watanabe. 2021a. Fast-MD: Fast multi-decoder end-to-end speech translation with non-autoregressive hidden intermediates. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 922–929. IEEE.

Hirofumi Inaguma, Yosuke Higuchi, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2021b. Non-autoregressive end-to-end speech translation with parallel autoregressive rescoring. *arXiv preprint arXiv:2109.04411*.

Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe. 2020. ESPnet-ST: All-in-one speech translation toolkit. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 302–311, Online. Association for Computational Linguistics.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerda, Javier Jorge, Nahuel Roselló, Adria Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-ST: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233. IEEE.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Ludwig Kürzinger, Dominik Winkelbauer, Lujun Li, Tobias Watzel, and Gerhard Rigoll. 2020. CTC-segmentation of large corpora for german end-to-end speech recognition. In *International Conference on Speech and Computer*, pages 267–278. Springer.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. 2020. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3520–3533.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Clara Meister, Tim Vieira, and Ryan Cotterell. 2020. Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8:795–809.

Niko Moritz, Takaaki Hori, and Jonathan Le Roux. 2019. Triggered attention for end-to-end speech recognition. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5666–5670.

Moses-SMT. 2018. `multi-bleu.perl`. `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.

H. Ney. 1999. Speech translation: coupling of recognition and translation. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 1, pages 517–520 vol.1.

Nicholas A Nystrom, Michael J Levine, Ralph Z Roskies, and J Ray Scott. 2015. Bridges: a uniquely flexible hpc resource for new communities and data analytics. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, pages 1–8.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*.

Vikas Raunak, Siddharth Dalmia, Vivek Gupta, and Florian Metze. 2020. On long-tailed phenomena in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3088–3095, Online. Association for Computational Linguistics.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, Online. Association for Computational Linguistics.

Tara N Sainath, Ruoming Pang, David Rybach, Yanzhang He, Rohit Prabhavalkar, Wei Li, Mirkó Visontai, Qiao Liang, Trevor Strohman, Yonghui Wu, et al. 2019. Two-pass end-to-end speech recognition. *Proc. Interspeech 2019*, pages 2773–2777.

Haşim Sak, Andrew Senior, Kanishka Rao, Ozan Irsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk. 2015. Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4280–4284. IEEE.

Elizabeth Salesky, Matthew Wiesner, Jacob Bremerman, Roldano Cattoni, Matteo Negri, Marco Turchi, Douglas W Oard, and Matt Post. 2021. The multilingual TEDx corpus for speech recognition and translation. *arXiv preprint arXiv:2102.01757*.

Ramon Sanabria and Florian Metze. 2018. Hierarchical multitask learning with CTC. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 485–490. IEEE.

Hiroshi Seki, Takaaki Hori, Shinji Watanabe, Niko Moritz, and Jonathan Le Roux. 2019. Vectorized beam search for CTC-attention-based speech recognition. In *INTERSPEECH*, pages 3825–3829.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231.

J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr. 2014. Xsede: Accelerating scientific discovery. *Computing in Science & Engineering*, 16(5):62–74.

Emiru Tsunoo, Yosuke Kashiwagi, and Shinji Watanabe. 2021. Streaming transformer asr with blockwise synchronous beam search. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 22–29.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Waibel. 1996. Interactive translation of conversational speech. *Computer*, 29(7):41–48.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Brian Yan, Patrick Fernandes, Siddharth Dalmia, Jiatong Shi, Yifan Peng, Dan Berrebbi, Xinyi Wang, Graham Neubig, and Shinji Watanabe. 2022. CMU's IWSLT 2022 dialect speech translation system. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 298–307, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Rico Sennrich. 2022. Revisiting end-to-end speech-to-text translation from scratch. In *International Conference on Machine Learning*.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2019. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.

A Supplementary Information

A.1 Additional Translation Metrics

To supplement our BLEU evaluation in Table 2, we also measure the translation quality of our models using Translation Error Rate (TER) (Snover et al., 2006) and METEOR (Banerjee and Lavie, 2005). As shown in Table 7, our findings are consistent across all three metrics for both MT and ST models.

A.2 MuST-C-v1 Back-Compatibility

See Table 9 for results compared to prior works.

A.3 Valid Set performances

Table 8 presents the validation performances for our ST and MT models.

A.4 Description of Encoder Architectures

LegoNN Encoder (Dalmia et al., 2022) is a stacked multi-block architecture that was introduced to encode and re-sample the input sequence into a sequence of representations of a desired length, which is typically a factor of the input sequence. It first encodes the input using transformer encoder blocks (Vaswani et al., 2017) and then re-encodes them into a sequence of latent representations using cross-attention. Starting from a sequence of learnable positional embeddings (Gehring et al., 2017), these latent representations are learned using another stack of transformer encoder layers with an added cross-attention component over the input representations in each block.

The Conformer encoder (Gulati et al., 2020) is a stacked multi-block architecture and has shown consistent improvement over a wide range of E2E speech processing applications (Guo et al., 2021). It includes a multi-head self-attention module, a convolution module, and a pair of position-wise feed-forward modules in the Macaron-Net style. While the self-attention module learns the long-range global context, the convolution module aims to model the local feature patterns synchronously.

A.5 Increased Cross-Attentional Monotonicity Leads to Increased Multilingual Parameter Sharing

We further examine the source attention parameters in our All-En models to understand the impact that the increased monotonicity of decoder attention (§6.2) has on multilingual parameter sharing. To do so, we extract sparse subnets for each language pair following the Lottery Ticket Sparse Fine-Tuning proposed by Ansell et al. (2022) and compute the pair-wise sharing across the 6 source languages, as measured by the count of overlapping parameters between subnets. In Figure 4, we show the relative change ($\Delta\%$) in multilingual sharing when using hierarchical encoding compared to the baseline. The broad increases suggest that the target-orientation of our encoder *reduced the decoder's burden of soft-aligning* target English outputs to source languages with varying word-orders, allowing for more efficient allocation of capacity.

A.6 compare_mt.py Length Analysis

As shown in Figure 5, both joint synchronous decodings are more robust than pure-attention for long output lengths across both MT and ST. Input-synchrony appears particularly more robust in generation of very long outputs for ST.

A.7 View of Regularized Attention

See Figure 6 for a qualitative example of monotonic source attention patterns (supplementary to the quantitative monotonicity in Figure 2).

B Reproducibility

B.1 Dataset Descriptions

See Table 10 for dataset descriptions. Data preparation was done using ESPnet recipes.

B.2 Model Architectures

See Table 11 for model architectures.

B.3 Training/Decoding Hyperparameters

See Tables 12-15 for hyperparameter descriptions.

B.4 Metrics

Sacrebleu signature for all non-Japanese:

```
BLEU+case.mixed+numrefs.1
+smooth.exp+tok.13a+version.1.5.1
```

Sacrebleu signature for Japanese:

```
BLEU+case.mixed+lang.en-ja+numrefs.1
+smooth.exp+tok.ja-mecab-0.996-IPA
+version.1.5.1
```

For tokenized BLEU in the IWSLT MT datasets we used `mutibleu.perl` (Moses-SMT, 2018)

B.5 Computing

ST models were trained on 2 x V100 for 2 days. MT models were trained on 1 x A6000 for 1 day.

Model Name	Decoding Method	IWSLT14 (De-En) BLEU (↑)	TER (↓)	METEOR (↑)	MuST-C-v2 En-De BLEU (↑)	TER (↓)	METEOR (↑)
Pure-Attn (Ours)	Attn-only	32.8	50.9	29.4	27.8	59.1	38.6
Joint CTC/Attn	Attn-only	33.6	50.7	30.0	28.3	58.4	39.2
Joint CTC/Attn	Joint I-Sync	33.7	50.6	30.0	**29.2**	57.8	40.1
Joint CTC/Attn	Joint O-Sync	**34.1**	**49.9**	**30.2**	**29.2**	**57.5**	**40.2**

Table 7: Test set performances, as measured by BLEU (↑), TER (↓), and METEOR (↑), of our proposed joint CTC/Attention models compared to pure-attention baselines.

Model Name	Decoding Method	IWSLT14 De-En	IWSLT14 Es-En	MuST-C-v2 En-De	MuST-C-v2 En-Ja
Pure-Attn (Ours)	Attn O-sync	34.1	41.2	28.5	11.3
Joint CTC/Attn	Joint I-sync	34.6	42.0	29.0	12.4
Joint CTC/Attn	Joint O-sync	35.0	42.3	29.2	12.4

Table 8: Valid set performances, as measured by BLEU (↑).

Model Name	MuST-C-v1 En-De
ESPnet-ST[1]	22.9
Dual-Decoder[2]	23.6
E2E-ST-TDA[3]	25.4
Multi-Decoder[4]	26.4
Pure-Attn (ours)	27.1
Joint CTC/Attn w/ Joint O-Sync	**28.2**

Table 9: Comparison of our best MuST-C-v1 En-De Joint CTC/Attn model and our Pure-Attn baseline with prior works: [1]Inaguma et al. (2020), [2]Le et al. (2020), [3]Du et al. (2022), [4]Dalmia et al. (2021)

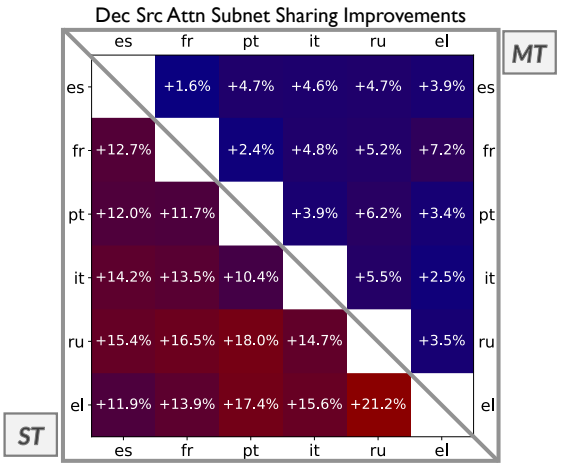

Figure 4: Improvement of multilingual sharing in MT/ST decoder source attention parameters when using joint CTC/Attention vs. attention-only training, as measured by pair-wise $\Delta\%$ in sparse subnet overlap.

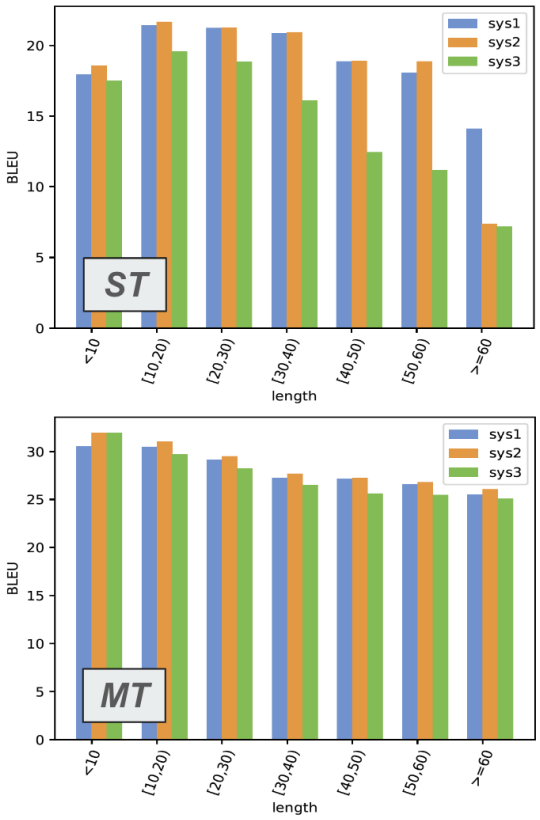

Figure 5: Compare-mt (Neubig et al., 2019) output sentence length to BLEU for joint decoding vs pure-attention models. Model codes: sys1 = Joint Input-Sync, sys2 = Joint Output-Sync, sys3 = Pure-Attn

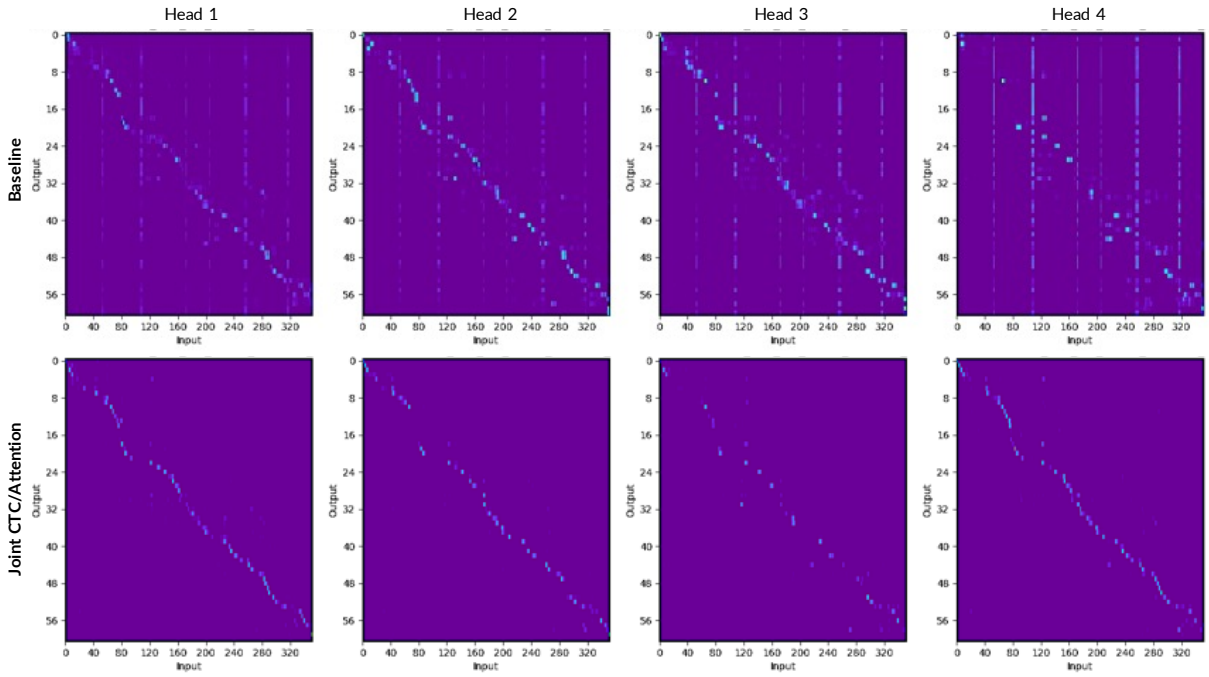

Figure 6: Visualization of source attention patterns produced by pure-attention baseline (top) vs. joint CTC/attention (bottom) ST models. Qualitative example extracted from the final decoder layer. Irregular patterns are observable in the pure-attention plots, but not in the joint CTC/attention plots.

Dataset	Task	Source Lang(s)	Target Lang(s)	Domain	# Train/Valid/Test Utts	# Speech Train Hours
IWSLT17 (Cettolo et al., 2012)	MT	De	En	TED Talk	160k/7k/7k	-
IWSLT17 (Cettolo et al., 2012)	MT	De	Es	TED Talk	160k/7k/7k	-
MuST-C-v2 (Di Gangi et al., 2019)	ASR/ST	En	De	TED Talk	250k/1k/3k	450h
MuST-C-v2 (Di Gangi et al., 2019)	ASR/ST	En	Ja	TED Talk	330k/1k/3k	540h
MTedX (Salesky et al., 2021)	MT	Es, Fr, Pt, It, Ru, El	En	TED Talk	130k/6k/6k	-
MTedX (Salesky et al., 2021)	ASR	Es, Fr, Pt, It, Ru, El	En	TED Talk	400k/6k/6k	730h
MTedX (Salesky et al., 2021)	ST	Es, Fr, Pt, It, Ru, El	En	TED Talk	130k/6k/6k	250h
EuroParl (Iranzo-Sánchez et al., 2020)	MT	De	En	Parliament Speech	- /-/2k	-
EuroParl (Iranzo-Sánchez et al., 2020)	ST	En	De	Parliament Speech	-/-/1k	-

Table 10: MT/ST/ASR dataset descriptions. Utterance counts are rounded to the nearest thousand. Language codes: De=German, En=English, Es=Spanish, Ja=Japanese, Fr=French, Pt=Portuguese, It=Italian, Ru=Russian, El=Greek

Model	Task	# Encoder Layers [$\mathcal{S}$]	# Decoder Layers	SrcCTC Layer	Up/Down-Sample	Pre-Train Init	Src BPE Size	Tgt BPE Size	# Params
Pure-Attn	MT	12 [6,12,18]	6	-	-	-	10k (joint)		54M
Joint CTC/Attn	MT	18 [6,12,18]	6	6	3x	-	10k (joint)		95M
Pure-Attn	ST	18 [12, 18]	6	-	1/4x	Enc lyr 1-12 from ASR	4k	4k	74M
Joint CTC/Attn	ST	18 [12, 18]	6	12	1/4x	Enc lyr 1-12 from ASR	4k	4k	72M
Pure-Attn	ASR	12	6	-	-	-	4k	4k	46M

Table 11: MT/ST/ASR model descriptions. The best MT/ST Encoder layers settings were selected over a search space indicated by $\mathcal{S}$. Parameter counts are rounded to the nearest million. Note that the 12 layer pure-attn model outperformed the 18 layer version and that the 12 layer joint model still outperformed these baselines.

Hyperparameter	Value
Hidden Dropout	0.3
Attention dropout	0.3
Activation dropout	0.3
LR schedule	inv. sqrt. (Vaswani et al., 2017)
Max learning rate	best of [1e-3, 3e-3]
Warmup steps	10000
Number of steps	200 epoch
Adam eps	1e-9
Adam betas	(0.9, 0.98)
Weight decay	1e-4
$\lambda 1$, $\lambda 2$	(1, 2)

Table 12: Training Hyperparameters for MT Models.

Hyperparameter	Value
Hidden Dropout	0.1
Attention dropout	0.1
Activation dropout	0.1
LR schedule	inv. sqrt. (Vaswani et al., 2017)
Max learning rate	0.002
Warmup steps	25000
Number of steps	40 epoch
Adam eps	1e-9
Adam betas	(0.9, 0.98)
Weight decay	0.0001
$\lambda 1$, $\lambda 2$	(2, 5)

Table 13: Training Hyperparameters for ST Models.

Decoding Type	Hyperparameter	Value
Pure Attn	Max Length Ratio	[1, 1.2, 1.4, 1.6, 1.8, 2, 2.5, 3]
	Penalty	[0, 0.2, 0.4, 0.6, 0.8, 1.0]
	Beam Size	5
Joint O-Sync	Max Length Ratio	1
	Penalty	[0, 0.2, 0.4, 0.6, 0.8, 1.0]
	CTC Weight	0.3
	Beam Size	5
Joint I-Sync	Max Length Ratio	1
	Penalty	[0, 0.2, 0.4, 0.6, 0.8, 1.0]
	Blank Penalty	[0.5, 0.75, 1.0]
	CTC Weight	[0.3, 0.5]
	Beam Size	[10, 30]

Table 14: Decoding Search Space MT Models.

Decoding Type	Hyperparameter	Value
Pure Attn	Max Length Ratio	1
	Penalty	[0,0.2,0.4,0.6,0.8,1.0]
	Beam Size	[10, 30, 50]
Joint O-Sync	Max Length Ratio	1
	Penalty	[0,0.2,0.4,0.6,0.8,1.0]
	CTC Weight	[0.3, 0.5]
	Beam Size	[10, 30, 50]
Joint I-Sync	Max Length Ratio	1
	Penalty	[0,0.2,0.4,0.6,0.8,1.0]
	Blank Penalty	1
	CTC Weight	[0.3, 0.5]
	Beam Size	[10, 30, 50]

Table 15: Decoding Search Space ST Models.