INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

A Comparative Study on E-Branchformer vs Conformer in Speech Recognition, Translation, and Understanding Tasks

Yifan Peng[1], *Kwangyoun Kim*[2], *Felix Wu*[2], *Brian Yan*[1], *Siddhant Arora*[1], *William Chen*[1], *Jiyang Tang*[1], *Suwon Shon*[2], *Prashant Sridhar*[2], *Shinji Watanabe*[1]

[1]Carnegie Mellon University, Pittsburgh, PA, USA
[2]ASAPP Inc., Mountain View, CA, USA

yifanpen@andrew.cmu.edu, {kkim,fwu}@asapp.com

Abstract

Conformer, a convolution-augmented Transformer variant, has become the *de facto* encoder architecture for speech processing due to its superior performance in various tasks, including automatic speech recognition (ASR), speech translation (ST) and spoken language understanding (SLU). Recently, a new encoder called E-Branchformer has outperformed Conformer in the LibriSpeech ASR benchmark, making it promising for more general speech applications. This work compares E-Branchformer and Conformer through extensive experiments using different types of end-to-end sequence-to-sequence models. Results demonstrate that E-Branchformer achieves comparable or better performance than Conformer in almost all evaluation sets across 15 ASR, 2 ST, and 3 SLU benchmarks, while being more stable during training. We will release our training configurations and pre-trained models for reproducibility, which can benefit the speech community. [1]

Index Terms: e-branchformer, conformer, speech recognition, speech translation, spoken language understanding

1. Introduction

Sequence-to-sequence (seq2seq) models have achieved remarkable success in end-to-end (E2E) speech processing. Various types of neural networks have been explored in recent years, including recurrent neural networks (RNNs) [1–3], convolutional neural networks (CNNs) [4–6] and Transformer networks based on self-attention [7–9]. These architectures have complementary capacity in sequence modeling. More recent studies have proposed to combine different networks for better performance. Conformer [10], a convolution-augmented Transformer encoder, has become the *de facto* standard architecture, due to its superior performance in many speech processing tasks [11]. Conformer captures both global and local contexts in a feature sequence through cascaded self-attention and convolution modules. An alternative approach is using parallel branches for different ranged contexts. Branchformer [12] follows this idea and shows competitive or even better results in several benchmarks for automatic speech recognition (ASR) and spoken language understanding (SLU), while being more stable for training in extreme data regimes. Further, E-Branchformer [13] enhances the vanilla Branchformer with a convolution-based merge module and Conformer-style feed-forward networks. It achieves new state-of-the-art (SOTA) results in the standard LibriSpeech ASR benchmark [14], which is highly encouraging for general speech applications.

In this work, we explore the efficacy of E-Branchformer in various speech processing tasks, including ASR, speech translation (ST) and SLU. We compare E-Branchformer versus Conformer through extensive experiments in a wide range of publicly available benchmarks (i.e., 15 ASR, 2 ST, and 3 SLU corpora). We also investigate different E2E frameworks, including connectionist temporal classification (CTC) [15], attention-based encoder-decoder (AED) and RNN-transducer (RNN-T) [16]. Results show that E-Branchformer performs equally well as or better than Conformer in almost all evaluation sets. E-Branchformer is also more stable to train when the model is large or the dataset is small. We share various training tips based on our investigations. We will also release our training configurations and pre-trained models for full reproducibility, which can significantly benefit the speech community.

[1]https://github.com/espnet/espnet

2. Models

We follow the default setup in the open-source ESPnet toolkit [17–19]. Only the encoder is changed to compare E-Branchformer [13] and Conformer [10], while the other components are the same. In a speech encoder, the raw audio waveform is first processed by a frontend to extract speech features such as log Mel filterbanks or self-supervised learning (SSL) based features. The feature sequence is then fed into a convolutional subsampling module. The downsampled feature sequence is further processed by a stack of identical encoder layers to capture high-level contextual information. Depending on the E2E framework, the final output sequence can be fed into a Transformer decoder, an RNN-T joint network or a CTC layer.

2.1. Conformer

Figure 1a illustrates a Conformer layer [10], which consists of a position-wise feed-forward network (FFN), a multi-head self-attention (MHA) module, a convolution (Conv) module and another FFN. Each of these modules has a residual connection [20] and a layer normalization [21] in a pre-norm style [22,23]. Different modules are combined sequentially. For an input sequence $\mathbf{X} \in \mathbb{R}^{T\times d}$ of length T and feature size d, the final output of a Conformer layer $\mathbf{Y}_{\text{conf}} \in \mathbb{R}^{T\times d}$ is computed as follows:

$$\mathbf{X}^1_{\text{conf}} = \mathbf{X} + \frac{1}{2}\text{FFN}^1(\mathbf{X}), \tag{1}$$

$$\mathbf{X}^2_{\text{conf}} = \mathbf{X}^1_{\text{conf}} + \text{MHA}(\mathbf{X}^1_{\text{conf}}), \tag{2}$$

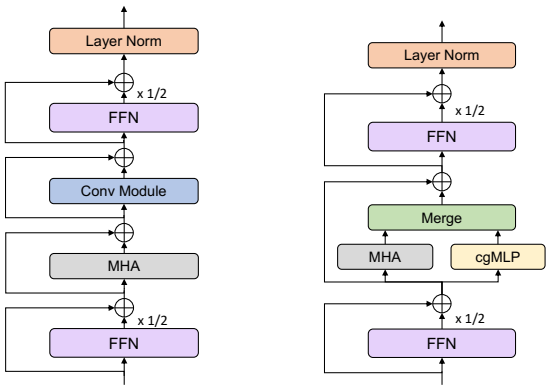
$$\mathbf{X}^3_{\text{conf}} = \mathbf{X}^2_{\text{conf}} + \text{Conv}(\mathbf{X}^2_{\text{conf}}), \tag{3}$$

$$\mathbf{X}^4_{\text{conf}} = \mathbf{X}^3_{\text{conf}} + \frac{1}{2}\text{FFN}^2(\mathbf{X}^3_{\text{conf}}), \tag{4}$$

$$\mathbf{Y}_{\text{conf}} = \text{LayerNorm}(\mathbf{X}^4_{\text{conf}}), \tag{5}$$

where $\mathbf{X}^1_{\text{conf}}, \mathbf{X}^2_{\text{conf}}, \mathbf{X}^3_{\text{conf}}, \mathbf{X}^4_{\text{conf}} \in \mathbb{R}^{T\times d}$ are intermediate outputs. Each FFN is composed of two linear projections and a

10.21437/Interspeech.2023-1194



(a) *Conformer [10]* (b) *E-Branchformer [13]*

Figure 1: *Comparison of encoder architectures.*

Swish activation [24] in between. Two separate FFNs with half-step residual weights are employed as in Macaron-Net [25]. Unlike the vanilla Transformer [7], MHA uses relative positional encodings from Transformer-XL [26]. The key component of Conformer is the Conv module which contains a point-wise convolution followed by a gated linear unit (GLU) activation [27], a 1-D depth-wise convolution, a batch normalization [28], a Swish activation and another point-wise convolution.

2.2. E-Branchformer

E-Branchformer [13] is an enhanced version of Branchformer [12]. Figure 1b shows its architecture. Similar to Conformer, it also has two Macaron-style FFNs. The difference is that E-Branchformer contains two parallel branches between the FFNs as proposed in Branchformer [12]. One branch captures global context using MHA while the other branch captures local context using multi-layer perceptron with convolutional gating (cgMLP) [29]. Two branches are merged by a concatenation operation, a 1-D depth-wise convolution and a linear projection, which is more effective than the simple concatenation followed by a linear projection used in Branchformer. For input $\mathbf{X} \in \mathbb{R}^{T \times d}$, the output $\mathbf{Y}_{\text{ebf}} \in \mathbb{R}^{T \times d}$ is defined as follows:

$$\mathbf{X}^1_{\text{ebf}} = \mathbf{X} + \frac{1}{2}\text{FFN}^1(\mathbf{X}), \tag{6}$$

$$\mathbf{X}^{2,\text{mha}}_{\text{ebf}},\ \mathbf{X}^{2,\text{mlp}}_{\text{ebf}} = \text{MHA}(\mathbf{X}^1_{\text{ebf}}),\ \text{cgMLP}(\mathbf{X}^1_{\text{ebf}}), \tag{7}$$

$$\mathbf{X}^2_{\text{ebf}} = \mathbf{X}^1_{\text{ebf}} + \text{Merge}(\mathbf{X}^{2,\text{mha}}_{\text{ebf}}, \mathbf{X}^{2,\text{mlp}}_{\text{ebf}}), \tag{8}$$

$$\mathbf{X}^3_{\text{ebf}} = \mathbf{X}^2_{\text{ebf}} + \frac{1}{2}\text{FFN}^2(\mathbf{X}^2_{\text{ebf}}), \tag{9}$$

$$\mathbf{Y}_{\text{ebf}} = \text{LayerNorm}(\mathbf{X}^3_{\text{ebf}}), \tag{10}$$

where $\mathbf{X}^1_{\text{ebf}}, \mathbf{X}^{2,\text{mha}}_{\text{ebf}}, \mathbf{X}^{2,\text{mlp}}_{\text{ebf}}, \mathbf{X}^2_{\text{ebf}}, \mathbf{X}^3_{\text{ebf}} \in \mathbb{R}^{T \times d}$ are intermediate outputs. Figure 2 shows the architecture of cgMLP [29], which leverages depth-wise convolution and linear gating to extract local contextual information. For input $\mathbf{X}^1_{\text{ebf}} \in \mathbb{R}^{T \times d}$, the output $\mathbf{X}^{2,\text{mlp}}_{\text{ebf}}$ is derived as follows:

$$\mathbf{Z} = \text{GeLU}(\text{LayerNorm}(\mathbf{X}^1_{\text{ebf}})\mathbf{U}) \in \mathbb{R}^{T \times d_{\text{mlp}}}, \tag{11}$$

$$\mathbf{A}, \mathbf{B} = \text{Split}(\mathbf{Z}) \in \mathbb{R}^{T \times \frac{1}{2} d_{\text{mlp}}}, \tag{12}$$

$$\tilde{\mathbf{Z}} = \mathbf{A} \odot \text{DwConv}(\text{LayerNorm}(\mathbf{B})) \in \mathbb{R}^{T \times \frac{1}{2} d_{\text{mlp}}}, \tag{13}$$

$$\mathbf{X}^{2,\text{mlp}}_{\text{ebf}} = \text{Dropout}(\tilde{\mathbf{Z}}\mathbf{V}) \in \mathbb{R}^{T \times d}, \tag{14}$$

where $\mathbf{U} \in \mathbb{R}^{d \times d_{\text{mlp}}}$ and $\mathbf{V} \in \mathbb{R}^{\frac{1}{2} d_{\text{mlp}} \times d}$ are two linear projections. $\odot$ denotes element-wise product.

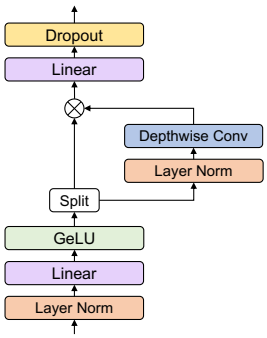


Figure 2: *Architecture of cgMLP [29].*

3. Speech recognition experiments

3.1. Setups

Data. A total of 15 public ASR datasets are utilized, covering various languages (English, Chinese, Spanish, Japanese, Italian, or even multilingual with 102 languages), recording environments (clean, noisy, far-field), speech types (spontaneous, read), and sizes (10 to 10k hours). Disordered speech from AphasiaBank [30] is also evaluated. The evaluation metric is character error rate (CER) or word error rate (WER). The total model size and encoder's multiply-accumulate operations (MACs) for a 10-second audio are also reported.

Models. We mainly use the attention-based encoder-decoder (AED) with joint CTC training and decoding [31, 32]. The encoder is either Conformer or E-Branchformer, while the decoder is a 6-layer Transformer. Log Mel filterbanks are extracted by default, except that FLEURS uses an SSL frontend as in [33]. FLEURS also exploits intermediate CTC [34] and self-condition CTC [35] in the encoder. We also conduct experiments using pure CTC or RNN-T models in a subset of datasets.

Training. We follow the ESPnet2 recipes[2] for data preparation, model training and decoding. Most recipes perform speed perturbation with ratios $\{0.9, 1.0, 1.1\}$ and SpecAugment [36]. A medium-sized model with hidden size $d = 256$ is employed by default. For FLEURS, GigaSpeech and LibriSpeech 960h, a larger model with $d = 512$ is trained instead. The Adam optimizer [37] with warmup learning rate schedule [7] is employed. Training hyperparameters such as the learning rate, weight decay and warmup steps are from existing baselines. We will release our detailed setups to ensure reproducibility.

3.2. Results

Table 1 summarizes the ASR results of AED models with joint CTC, which is the most widely used setup in ESPnet. Compared to those well-established Conformer baselines, E-Branchformer achieves comparable or superior results with a similar model size and computational complexity in almost all benchmarks. **We have only observed a slight degradation in one set among 39 evaluation sets across 15 corpora.** The improvements are especially remarkable in AphasiaBank, CHiME4, Fisher-Callhome, FLEURS, JSUT, MuST-C and TEDLIUM2, indicating that E-Branchformer has strong modeling capacities for various speech types.

Conformer configurations can vary across different datasets, with some datasets benefiting from deeper networks while others may benefit from wider networks. Table 2 compares E-Branchformer with two Conformer baselines in four

[2] https://github.com/espnet/espnet/tree/master/egs2

Table 1: *CER or WER (%) on speech recognition benchmarks using attention-based encoder-decoder (AED) with joint CTC. The total number of parameters ($\times 10^6$) and encoder's multiply-accumulate (MAC) operations ($\times 10^9$) are also reported. † means a frozen SSL frontend is used but not counted. ‡ means a language model is used with shallow fusion following existing ESPnet2 recipes.*

Dataset	Token	Metric	Evaluation Sets	Conformer			E-Branchformer		
				Params	MACs	Results ↓	Params	MACs	Results ↓
AIDATATANG [38]	Char	CER	dev / test	46.0	14.7	‡ 3.6 / 4.3	45.4	15.5	‡ **3.4 / 4.1**
AISHELL [39]	Char	CER	dev / test	46.3	15.3	4.3 / 4.6	45.7	15.5	**4.2 / 4.4**
AphasiaBank [30]	Char	WER	patients / control	44.2	30.1	40.3 / 35.3	45.7	32.0	**36.2 / 31.2**
CHiME4 [40]	Char	WER	{dt05,et05}_{simu,real}	30.4	8.8	‡ 7.8 / 9.5 / 12.5 / 14.8	30.8	8.8	‡ **6.8 / 8.4 / 10.8 / 13.0**
Fisher-Callhome [41]	BPE	WER	dev / dev2 / test / devtest / evltest	43.8	11.6	20.7 / 20.9 / 19.4 / 38.3 / 38.8	43.2	12.1	**20.5 / 20.2 / 18.7 / 37.8 / 37.6**
FLEURS [42]	BPE	CER	dev / test	†126.6	48.8	‡ 10.1 / 10.4	†127.4	50.1	‡ **9.3 / 9.2**
GigaSpeech [43]	BPE	WER	dev / test	116.2	20.0	10.9 / 10.8	148.9	26.1	**10.6 / 10.5**
JSUT [44]	Char	CER	dev / eval1	45.1	11.6	‡ 12.3 / 13.6	44.2	12.1	‡ **11.8 / 13.0**
LibriSpeech 100h [14]	BPE	WER	{dev,test}_{clean,other}	39.0	10.3	6.3 / 17.0 / 6.6 / 17.2	38.5	9.9	**6.1 / 16.7 / 6.3 / 17.0**
LibriSpeech 960h [14]	BPE	WER	{dev,test}_{clean,other}	147.8	42.5	‡ 1.72 / 3.65 / **1.85** / 3.95	148.9	42.7	‡ **1.67 / 3.64 / 1.85 / 3.71**
MuST-C [45]	BPE	WER	tst-{COMMON, HE}.en-de	46.1	12.0	7.7 / 6.7	37.7	9.9	**7.3 / 6.0**
Switchboard [46]	BPE	WER	eval2000 (callhm / swbd)	36.7	10.3	13.5 / 7.4	36.2	9.9	**13.4 / 7.3**
TEDLIUM2 [47]	BPE	WER	dev / test	35.5	10.3	7.5 / 7.6	35.0	9.9	**7.3 / 7.1**
VoxForge [48]	Char	CER	dt_it / et_it	35.2	13.2	9.0 / 8.1	34.7	12.6	**8.8 / 8.0**
WSJ [49]	Char	WER	dev93 / eval92	35.2	13.2	‡ **6.5 / 4.1**	34.7	12.6	‡ **6.5** / 4.3

Table 2: *CER or WER (%) of different configurations using AED with joint CTC. "Conformer-Deep" has 15 encoder layers with 1024 FFN units, while "Conformer-Wide" has 12 encoder layers with 2048 FFN units. Evaluation sets are the same as in Table 1, except that LibriSpeech 100h only shows test sets.*

Dataset	Conformer-Deep		Conformer-Wide		E-Branchformer	
	Params	Results ↓	Params	Results ↓	Params	Results ↓
LibriSpeech 100h	39.0	6.6 / 17.2	46.8	6.6 / 17.1	38.5	**6.3 / 17.0**
Switchboard	36.7	13.5 / 7.4	44.5	13.8 / 7.5	36.2	**13.4 / 7.3**
TEDLIUM2	35.5	7.5 / 7.6	43.4	7.5 / 7.5	35.0	**7.3 / 7.1**
VoxForge	35.2	9.0 / 8.1	43.0	8.9 / **8.0**	34.7	**8.8 / 8.0**

Table 3: *CER or WER (%) of pure CTC-based models. Greedy search is performed without a language model.*

Dataset	Conformer		E-Branchformer	
	Params	Results ↓	Params	Results ↓
AISHELL	26.8	5.8 / 6.3	26.2	**5.4 / 6.0**
LibriSpeech 100h	27.0	9.4 / 22.5 / 9.9 / **23.1**	26.4	**9.2 / 22.4 / 9.6 / 23.1**
TEDLIUM2	25.8	9.1 / 9.0	25.3	**8.7 / 8.3**

benchmarks. The hidden sizes are $d = 256$ for all models. E-Branchformer consists of 12 encoder layers with 1024 FFN units and 1024 MLP units (d_{mlp} in Eq. (11)). "Conformer-Deep" has 15 encoder layers with 1024 FFN units, while "Conformer-Wide" has 12 layers with 2048 FFN units. E-Branchformer consistently achieves lower CERs or WERs than "Conformer-Deep" with a similar size. It even shows better or similar performance than "Conformer-Wide" whose size is 20% larger.

Different E2E ASR frameworks (i.e., AED, CTC and RNN-T) typically share a similar encoder. To investigate the generalizability of Conformer and E-Branchformer encoders, we apply them to pure CTC and RNN-T models. Table 3 and Table 4 summarize the two sets of experiments, respectively. Similar to the AED results, E-Branchformer achieves consistent improvements over Conformer in various evaluation sets under the same training condition. This demonstrates that E-Branchformer encoders are capable of extracting better contextualized speech representations which generally benefit E2E ASR.

3.3. Discussions

The following are some observations and training tips based on our investigations.

- When the model is large or the data is small, E-Branchformer (and Branchformer) can offer greater training stability than Conformer, as observed empirically. For instance, in an experiment with TEDLIUM2 where we set the peak learning rate to 2e-3 and used pure CTC with 33M parameter encoders, **Conformer failed in 6 out of 10 random trials, while E-Branchformer only failed twice.** One reason for this may be that Conformer's sequential combination of modules increases the encoder's depth and makes it harder to converge. Moreover, in the successful trials, E-Branchformer exhibited lower validation loss in an early stage of training. It is worth noting that although a lower learning rate could improve training stability, in our case, it degraded the WERs.
- A similar configuration of E-Branchformer performs well in many ASR corpora. Using Macaron-style FFNs like those in Conformer is generally beneficial. For medium-sized models, we recommend setting the hidden size to $d = 256$ and using $4\times$ expansion in FFNs and cgMLP. For large models, we suggest following the original paper [13] and using $d = 512$ with $6\times$ expansion in cgMLP and $2\times$ expansion in FFNs.
- The same training hyperparameters (e.g., batch size, learning rate, warmup steps, total epochs) used by Conformer usually work well for E-Branchformer assuming their sizes are close. In most experiments, we did not change these configurations.
- Stochastic depth [51] is disabled in most of our experiments for fair comparison with prior baselines. However, we do find that it can slightly improve the final results for datasets like FLEURS and LibriSpeech 960h. Moreover, it can be applied to both encoder and decoder, e.g., with dropout probabilities 0.1 and 0.2, respectively.
- With joint CTC training [31], the auxiliary CTC loss can be a reliable metric for assessing the success of the training process in an early stage. If the validation loss does not decrease in the first few epochs, we need to reduce the learning rate or increase warmup steps. This also applies to Conformer.

Table 4: *CER or WER (%) of RNN-T models. The decoder is a single-layer LSTM [50]. Beam search with beam size 10 is performed without a language model.*

Dataset	Conformer		E-Branchformer	
	Params	Results ↓	Params	Results ↓
AISHELL	29.9	**4.9** / 5.3	29.4	**4.9 / 5.2**
LibriSpeech 100h	30.5	**6.6** / 17.9 / 6.9 / 18.1	30.0	**6.6 / 17.6 / 6.8 / 18.0**
TEDLIUM2	26.8	8.1 / 7.7	26.3	**7.6 / 7.4**

Table 5: *Speech translation results.*

Dataset	Conformer		E-Branchformer	
	Params	BLEU ↑	Params	BLEU ↑
MuST-C [45]	74.5	28.6	71.4	**28.7**
Fisher [41]	69.8	55.5	66.8	**55.6**
Callhome [41]	69.8	21.2	66.8	**21.9**

4. Speech translation experiments

4.1. Setups

Data. Two ST datasets are used. MuST-C [45] is a TED-Talk corpus for English-to-X translation. We use the 400 hour v2 set of English-to-German. Fisher-Callhome [41] is a 170 hour conversational corpus for Spanish-to-English translation.

Models. We use the attention-based encoder-decoder (AED) with joint CTC training and decoding [31, 32]. Our ST models are similar to our ASR models, except they use hierarchical CTC encoders [52] which consist of a 12-layer Conformer or E-Branchformer attached to an ASR CTC criterion followed by another 6-layer Conformer or 8-layer E-Branchformer attached to an ST CTC criterion. Given the greater number of layers for ST models, we use a slightly higher number of layers for our E-Branchformer models to keep parameter counts in a similar range – E-Branchformer models are still smaller. ST models use ASR pre-training for the first 12 encoder layers.

Training. We follow the ESPnet2 recipes for data preparation, model training and decoding. Speed perturbation with ratios $\{0.9, 1.0, 1.1\}$ and SpecAugment [36] are performed. The medium-sized model with hidden size $d = 256$ is used. The Adam [37] optimizer with warmup [7] is employed.

4.2. Results

Table 5 shows the ST results. E-Branchformer achieves a higher BLEU score than Conformer on Callhome (21.9 vs. 21.2) and also shows minor improvements on MuST-C and Fisher with a slightly smaller model size. This suggests that E-Branchformer is capable of handling non-monotonic sequence transductions such as translation where source-to-target word re-ordering may occur.

5. Speech understanding experiments

5.1. Setups

Data. Three SLU datasets are used. SLURP [53] is a multi-domain corpus for intent classification and entity recognition, which contains single-turn user interactions with a home assistant. SLUE [54] is a low-resource benchmark containing naturally produced speech for named entity recognition (NER) and sentiment analysis. STOP [55] is a large-scale corpus for spoken task-oriented semantic parsing.

Models. As in ESPnet-SLU [19], SLU tasks are formulated as seq2seq problems. The input is a sequence of speech features, and the output is a sequence of text tokens including special SLU labels. Then, the same E2E ASR models can be applied. Specifically, we employ AED models with joint CTC [31, 32], where the decoder is a 6-layer Transformer. Similar to [56], an SSL frontend is used for SLUE and STOP.

Training. We follow the ESPnet2 recipes for data preparation, training and decoding. Speed perturbation and SpecAugment [36] are performed for data augmentation. The Adam [37] optimizer with warmup [7] is employed.

Table 6: *Spoken language understanding results on test sets. SLURP shows intent classification accuracy (%) and SLU-F1 (%). SLUE-voxpopuli shows micro and macro F1 (%). SLUE-voxceleb shows macro F1 (%). STOP shows exact match accuracy (%). † means a frozen SSL frontend is used but not counted.*

Dataset	Conformer		E-Branchformer	
	Params	Results ↑	Params	Results ↑
SLURP [53]	109.4	86.5 / 76.9	110.2	**87.4 / 77.6**
SLUE-voxpopuli [54]	† 32.4	68.6 / 55.8	† 33.5	**68.7 / 55.9**
SLUE-voxceleb [54]	† 32.4	**38.5**	† 33.5	38.1
STOP [55]	† 114.3	73.2	† 146.8	**74.0**

5.2. Results

Table 6 shows SLU results. E-Branchformer has superior performance over Conformer on SLURP with a similar model size. It also achieves a higher accuracy on STOP, but the model size is larger due to the reuse of configuration from LibriSpeech 960h. For low-resource SLUE-voxpopuli (NER) and SLUE-voxceleb (sentiment analysis) corpora, training is done 3 times with different random seeds, and metrics are averaged. We observe that E-Branchformer is slightly better on SLUE-voxpopuli but worse on SLUE-voxceleb. This indicates that the frozen SSL frontend might be more important than the additional encoder layers in low-resource SLU tasks.

6. Conclusion

This work investigates the effectiveness of E-Branchformer in various speech processing tasks, including ASR, ST, and SLU, and compares it to Conformer using different E2E frameworks. Our extensive experiments in publicly available benchmarks have shown that E-Branchformer outperforms Conformer in a wide variety of tasks, and can be more stable when training with large model size or on small datasets. In addition, we share various training tips and we will release our training configurations and pre-trained models to ensure full reproducibility, which can greatly benefit the speech community. Future research directions would include evaluating E-Branchformer on more diverse and challenging datasets in low-resource languages and noisy environments. Additionally, the E-Branchformer architecture may also improve the performance of SSL models such as WavLM [57] and data2vec 2.0 [58].

7. Acknowledgements

This work used PSC Bridges2 and NCSA Delta through allocation CIS210014 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

8. References

[1] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016.

[2] A. Zeyer *et al.*, "Improved Training of End-to-end Attention Models for Speech Recognition," in *Proc. Interspeech*, 2018.

[3] C.-C. Chiu, T. N. Sainath *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018.

[4] T. N. Sainath, B. Kingsbury *et al.*, "Improvements to Deep Convolutional Neural Networks for LVCSR," in *Proc. ASRU*, 2013.

[5] J. Li, V. Lavrukhin *et al.*, "Jasper: An End-to-End Convolutional Neural Acoustic Model," in *Proc. Interspeech*, 2019.

[6] W. Han, Z. Zhang, Y. Zhang *et al.*, "ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context," in *Proc. Interspeech*, 2020.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.

[8] S. Karita, N. Chen, T. Hayashi *et al.*, "A comparative study on transformer vs rnn in speech applications," in *Proc. ASRU*, 2019.

[9] Q. Zhang, H. Lu, H. Sak *et al.*, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *Proc. ICASSP*, 2020.

[10] A. Gulati, J. Qin *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020.

[11] P. Guo, F. Boyer, X. Chang *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *Proc. ICASSP*, 2021.

[12] Y. Peng *et al.*, "Branchformer: Parallel MLP-attention architectures to capture local and global context for speech recognition and understanding," in *Proc. ICML*, 2022.

[13] K. Kim, F. Wu *et al.*, "E-Branchformer: Branchformer with Enhanced Merging for Speech Recognition," in *Proc. SLT*, 2022.

[14] V. Panayotov *et al.*, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[15] A. Graves *et al.*, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006.

[16] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[17] S. Watanabe, T. Hori, S. Karita *et al.*, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech*, 2018.

[18] H. Inaguma *et al.*, "ESPnet-ST: All-in-one speech translation toolkit," in *Proc. ACL: System Demonstrations*, 2020.

[19] S. Arora *et al.*, "ESPnet-SLU: Advancing Spoken Language Understanding Through ESPnet," in *Proc. ICASSP*, 2022.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016.

[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[22] Q. Wang, B. Li, T. Xiao *et al.*, "Learning deep transformer models for machine translation," in *Proc. ACL*, 2019.

[23] T. Q. Nguyen and J. Salazar, "Transformers without tears: Improving the normalization of self-attention," in *Proc. IWSLT*, 2019.

[24] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[25] Y. Lu *et al.*, "Understanding and improving transformer from a multi-particle dynamic system point of view." in *Proc. ICLR Workshop on Integrat. of Deep Neural Models and Diff. Eq.*, 2019.

[26] Z. Dai *et al.*, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. ACL*, 2019.

[27] Y. N. Dauphin, A. Fan *et al.*, "Language modeling with gated convolutional networks," in *Proc. ICML*, 2017.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[29] J. Sakuma, T. Komatsu, and R. Scheibler, "MLP-based architecture with variable length input for automatic speech recognition," 2022. [Online]. Available: https://openreview.net/forum?id=RA-zVvZLYIy

[30] M. M. Forbes *et al.*, "AphasiaBank: A resource for clinicians," in *Proc. Seminars in speech and language*, 2012.

[31] S. Kim *et al.*, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. ICASSP*, 2017.

[32] T. Hori, S. Watanabe, and J. R. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. ACL*, 2017.

[33] W. Chen, B. Yan *et al.*, "Improving Massively Multilingual ASR With Auxiliary CTC Objectives," in *Proc. ICASSP*, 2023.

[34] J. Lee and S. Watanabe, "Intermediate loss regularization for ctc-based speech recognition," in *Proc. ICASSP*, 2021.

[35] J. Nozaki and T. Komatsu, "Relaxing the Conditional Independence Assumption of CTC-Based ASR by Conditioning on Intermediate Predictions," in *Proc. Interspeech*, 2021.

[36] D. S. Park, W. Chan, Y. Zhang *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech*, 2019.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[38] "aidatatang_200zh, a free Chinese Mandarin speech corpus by Beijing DataTang Technology Co., Ltd (www.datatang.com)."

[39] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Proc. O-COCOSDA*, 2017.

[40] E. Vincent *et al.*, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer speech & language*, vol. 46, pp. 535–557, 2017.

[41] M. Post *et al.*, "Improved speech-to-text translation with the fisher and callhome Spanish-English speech translation corpus," in *Proc. IWSLT*, 2013.

[42] A. Conneau *et al.*, "FLEURS: Few-Shot Learning Evaluation of Universal Representations of Speech," in *Proc. SLT*, 2022.

[43] G. Chen *et al.*, "GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio," in *Proc. Interspeech*, 2021.

[44] R. Sonobe, S. Takamichi, and H. Saruwatari, "Jsut corpus: free large-scale japanese speech corpus for end-to-end speech synthesis," *arXiv preprint arXiv:1711.00354*, 2017.

[45] R. Cattoni, M. A. Di Gangi, L. Bentivogli *et al.*, "Must-c: A multilingual corpus for end-to-end speech translation," *Computer speech & language*, vol. 66, p. 101155, 2021.

[46] J. Godfrey *et al.*, "SWITCHBOARD: telephone speech corpus for research and development," in *Proc. ICASSP*, 1992.

[47] A. Rousseau, P. Deléglise, Y. Esteve *et al.*, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks." in *LREC*, 2014, pp. 3935–3939.

[48] "VoxForge." [Online]. Available: http://www.voxforge.org/

[49] D. B. Paul and J. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. Workshop on Speech and Natural Language*, 1992.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[51] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. ECCV*, 2016.

[52] B. Yan *et al.*, "CTC alignments improve autoregressive translation," *arXiv preprint arXiv:2210.05200*, 2022.

[53] E. Bastianelli, A. Vanzo *et al.*, "SLURP: A Spoken Language Understanding Resource Package," in *Proc. EMNLP*, 2020.

[54] S. Shon, A. Pasad, F. Wu, P. Brusco, Y. Artzi, K. Livescu, and K. J. Han, "Slue: New benchmark tasks for spoken language understanding evaluation on natural speech," in *Proc. ICASSP*, 2022.

[55] P. Tomasello, A. Shrivastava, D. Lazar *et al.*, "STOP: A Dataset for Spoken Task Oriented Semantic Parsing," in *Proc. SLT*, 2022.

[56] Y. Peng, S. Arora, Y. Higuchi *et al.*, "A study on the integration of pre-trained ssl, asr, lm and slu models for spoken language understanding," in *Proc. SLT*, 2022.

[57] S. Chen *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, 2022.

[58] A. Baevski *et al.*, "Efficient self-supervised learning with contextualized target representations for vision, speech and language," *arXiv preprint arXiv:2212.07525*, 2022.