INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Optimally Encoding Inductive Biases into the Transformer Improves End-to-End Speech Translation

Piyush Vyas[1], *Anastasia Kuznetsova*[1,2], *Donald S. Williamson*[1]

[1] Department of Computer Science, Indiana University Bloomington, USA
[2] Department of Linguistics, Indiana University Bloomington, USA

{piyush, anakuzne}@iu.edu, williads@indiana.edu

Abstract

Transformer-based encoder-decoder architectures have recently shown promising results in end-to-end speech translation. However, the content-based attention mechanism employed by the Transformer was designed for text sequences and can only encode global inductive bias, that alone is not sufficient for learning good representations from speech signals. In this work, we address this by putting architectural constraints on the Transformer to allow encoding of both local and global inductive biases. This is accomplished by replacing the Transformer encoder with a Conformer encoder that, in contrast to the Transformer encoder, employs convolution in addition to self-attention and feed-forward. As a result, the new model named Conformer-Transformer has an encoder that captures both local feature correlations and long-range dependencies from speech signals. Experiments on seven non-English to English language directions show that the Conformer-Transformer, compared to strong Transformer-based baselines, achieves up to 3.54 BLEU score improvements with a pre-trained encoder and up to 10.53 BLEU score improvements when trained from scratch.

Index Terms: speech translation, transformer, inductive biases, self-attention, convolution

1. Introduction

Speech translation (ST) is a complex machine learning task that maps *source* language speech signals to *target* language text. Conventional methods for ST employ a cascade system [1, 2, 3, 4, 5] wherein the source language speech signals go through an automatic speech recognition (ASR) model that generates source language text transcriptions, which are passed to a machine translation (MT) model that generates text translations in the target language. Since cascade systems employ two separate models, they are slow at inference, prone to the propagation of errors from the ASR model to the MT model, and cannot effectively be applied to low-resource languages for which source text transcripts are hard to collect. To overcome the drawbacks of cascade systems, [6, 7, 8] propose end-to-end systems that only employ a single recurrent encoder-decoder model that does not require source language text transcripts. Recurrent networks, however, are very slow at training due to their sequential manner of processing information.

A Transformer [9], on the other hand, is an encoder-decoder architecture that replaced recurrence and convolution with self-attention and achieved state-of-the-art results in MT. It was first used for ST in [10] and showed promising results, that made it a viable end-to-end model. Recently, Transformer-based end-to-end ST models have outperformed both cascade and previous recurrent end-to-end systems in the shared task of offline speech translation [11]. However, the Transformer employs a content-based attention mechanism that cannot encode local inductive bias, which is essential to learn good representations from data with a local proximity bias (e.g., speech signals).

Inductive biases (or learning biases) are one of the main attributes that guide a learning algorithm towards generalization [12, 13]. Learning algorithms lacking suitable inductive biases can be easily tempted by local minima on the loss surface [14]. Hence, strong inductive biases are needed for a learning algorithm to descend towards global minima and achieve good generalization performance. For speech data, a local inductive bias helps capture short-range dependencies whereas a non-local inductive bias helps capture long-range dependencies. Since in ST the Transformer Encoder has to deal with speech signals, the absence of a local inductive bias will result in bad generalization performance.

Previous works have accounted for the absence of a local inductive bias in the Transformer encoder. In [15], a convolution-based 2D attention is proposed to allow modeling of both temporal and spectral dynamics in speech inputs, but it was evaluated only for speech recognition. A linear distance map is introduced in [16] to model the relative distance between words in a sentence, but was evaluated for sentence encoding. The approach in [17] adds a soft Gaussian mask to the attention energies to provide attention heads with explicit control over the context range. In a similar approach, [18] penalizes the encoder attention weights with a logarithmic distance penalty in addition to employing the convolution-based 2D attention [15]. Though all of these methods have shown promising results in the tasks they are evaluated on, we argue that adjusting the self-attention mechanism to encode local inductive bias is not an optimal method and will lead the Transformer to sub-optimal solutions. In contrast, [19] proposes a convolution augmented Transformer that encodes both local and global inductive biases in a parameter-efficient and optimal manner. However, it was also evaluated only on speech recognition. Moreover it uses an long short-term memory (LSTM) decoder and is trained in a RNN-Transducer [20] framework that uses a joint network to output a distribution over all possible input-output alignments.

Encoding inductive biases into learning algorithms can be done in many ways: architectural constraints, parameter sharing, objective function, curriculum, or optimization method. We focus our efforts on using architectural constraints as a means of encoding inductive biases into the Transformer. One very simple, yet powerful, architectural constraint to inject local and non-local inductive biases into the Transformer is to employ both convolution and self-attention in the encoder. We hypothesize that, as a result of this simple architectural constraint on the Transformer encoder, both convolution and self-attention will inject their underlying inductive biases into the encoder, enabling the encoder to capture fine-grained local correlations and long-range relationships from speech signals. Our contri-

http://dx.doi.org/10.21437/Interspeech.2021-2007

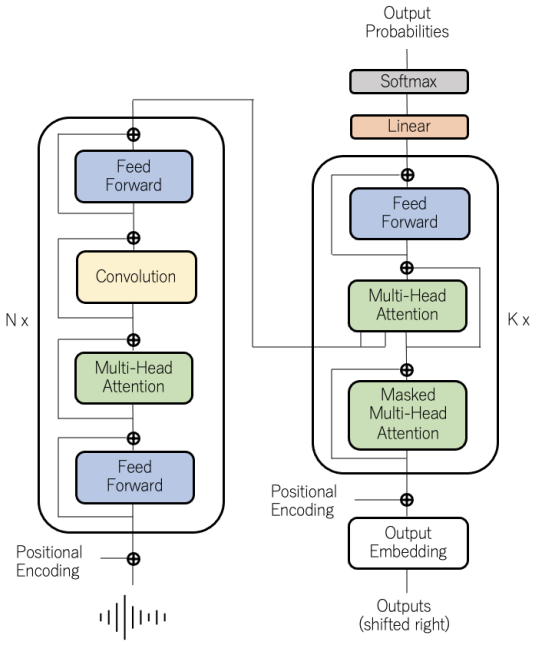

Figure 1: *Proposed Conformer-Transformer model with vanilla Transformer Decoder (right) and Conformer Encoder (left).*

butions are as follows: We show that encoding suitable inductive biases into the Transformer using simple architectural constraints improves end-to-end speech translation. We propose to employ self-attention and convolution in an optimal manner that is - use self-attention for capturing long-range dependencies and convolution for learning local correlations. To the best of our knowledge, this is the first attempt in end-to-end ST that, to encode local inductive bias and extract local feature patterns from speech signals, leaves self-attention unmodified and relies solely on convolution.

The rest of the paper is laid out as follows: Section 2 introduces our proposed model, Section 3 outlines the dataset, data pre-processing, and experimental setups, Section 4 presents the results, Section 5 discusses about the findings, and finally Section 6 summarizes the results of the work.

2. Conformer-Transformer

Our proposed model named Conformer-Transformer, depicted in Figure 1 has an encoder-decoder architecture similar to that of the vanilla Transformer [9], however, we replace the encoder with a Conformer Encoder [19] to encode both local and non-local inductive biases into the encoder.

Before feeding the speech features to the Conformer Encoder, we apply two temporal convolution sub-sampling layers each with a kernel size of 5 and stride of 2 to achieve a 4x reduction across the temporal dimension, resulting in a final input feature vectors at 40ms rate. Each convolution layer is followed by layer normalization [21] and a GELU activation [22]. Sinusoidal positional encodings [9] are added to the outputs of the last convolution layer.

2.1. Conformer Encoder

A Conformer Encoder consists of *N* identical encoder blocks. Each encoder block has four modules: Feed-Forward, Multi-Head Self-Attention, Convolution, and another Feed-Forward.

The feed-forward module consists of two linear transformations with a ReLU activation in between. The two linear transformations are analogous to pointwise convolutions (or convolutions with a 1 x 1 kernel).

We use the same multi-head self-attention (MHSA) mechanism as in [9] wherein a query and a set of key-value pairs are mapped to an output. The set of queries, keys, and values are all packed together into matrices Q, K, and V respectively. The final matrix of outputs is computed using the scaled dot-product attention function as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where d_k is the key dimension. The attention function is applied to h heads in parallel. All h heads are first concatenated and than projected, resulting in the final outputs as depicted in Equation 2.

$$\begin{aligned} \text{MHSA}(Q, K, V) &= \text{Concat}(\text{head}_i, ..., \text{head}_h)W^O \\ \text{where head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2}$$

where $i \in [1, h]$ and W^Q, W^K, W^V, W^O are query, key, value, and output projection matrices.

The convolution module includes the following components: Pointwise convolutions, Gated linear unit (GLU) [23], 1D depth-wise convolution, Batch normalization [24] and a Swish activation [25]. The inputs to the module first go through a gated convolution - pointwise convolution with GLU activation, followed by a 1D depth-wise convolution. Batch normalization and Swish activation are applied after 1D depth-wise convolution. This is followed by another pointwise convolution.

In each Conformer block, residual connections are employed after each module, however, we use half-step residual weights in the feed-forward modules. According to the ablation studies by [19], using half-step residual weights yields better performance. Hence, we adopt this strategy as well. Dropout [26] is applied before each residual connection. Inputs to each Conformer block are processed as:

$$\begin{aligned} \bar{x}_i &= x_i + \frac{1}{2}\text{FF}(\text{LayerNorm}(x_i)) \\ \grave{x}_i &= \bar{x}_i + \text{MHSA}(\text{LayerNorm}(\bar{x}_i)) \\ \acute{x}_i &= \grave{x}_i + \text{Conv}(\text{LayerNorm}(\grave{x}_i)) \\ y_i &= \acute{x}_i + \frac{1}{2}\text{FF}(\text{LayerNorm}(\acute{x}_i)) \end{aligned} \tag{3}$$

where x_i and y_i are the inputs and outputs of the i-th Conformer Block and $i \in [1, ..., N]$. FF, MHSA, and Conv indicate feed-forward, multi-head self-attention, and convolution modules respectively. Layer Normalization is abbreviated as LayerNorm.

2.2. Transformer Decoder

Our decoder is the same as the vanilla Transformer decoder as in [9]. It consists of $K = 6$ identical decoder blocks each with three modules: multi-head self-attention (MHSA), masked

multi-head self-attention (mMHSA), and feed-forward. The mMHSA module applies a look-ahead mask to the attention weights to prevent positions from attending to subsequent positions. Besides, the keys and values for the mMHSA come from the encoder. Similar to the encoder, inputs to each module in the decoder block are layer normalized and a residual connection is employed between the outputs and the pre-normalized inputs of each module. The decoder is defined in Equation 4 where j is the index of an decoder block and $j \in [1, M]$. K, V are key and value matrices from the encoder.

$$\begin{aligned} \bar{x}_j &= x_j + \text{mMHSA}(\text{LayerNorm}(x_j)) \\ \grave{x_j} &= \bar{x}_j + \text{MHSA}(\text{LayerNorm}(\bar{x}_j, K, V)) \\ y_j &= \grave{x_j} + \text{FF}(\text{LayerNorm}(\grave{x_j})) \end{aligned} \tag{4}$$

In the original transformer implementation, the output embedding weights are shared with the final linear layer of the network following the decoder. However, we avoid this weight-sharing as our early experiments did not show any benefit in doing so.

3. Experiments

3.1. Data

We evaluate our model on seven language directions: {French (Fr), German (De), Spanish (Es), Italian (It), Russian (Ru), Chinese (Zh), Portuguese (Pt)} → English (En) from the large-scale speech translation benchmark CoVoST 2 [27]. For English ASR pre-training of the encoder (when applicable), we use the Common Voice Corpus 4 [28].

3.2. Experimental Setups

3.2.1. Data Preprocessing

We extract 80-dimensional log-mel filterbank features composed from a 25ms window with a stride of 10ms. Filterbanks are normalized to have zero mean and unit variance. Spectrogram data augmentation [29] with time and frequency masking (LB policy) is applied during training to prevent over-fitting. Character vocabularies with 100% coverage are built on training data using SentencePiece text tokenizer [30].

3.2.2. Training and Inference

Our models have input-output embedding dimension $d_{model} = 256$, inner-layer dimension $d_{ff} = 1024$, and $h = 4$ attention heads each with dimension $d_{model}/h = 64$. Our *base* model has 6 encoder and 6 decoder layers. For our *deep* model, we use 12 encoder layers. For [18] we used their best model, S-Transformer Big+Log, that has 6 encoder and 6 decoder layers with a higher input-output embedding dimension $d_{model} = 512$ and $h = 8$ attention heads. The model from [27] has 12 encoder and 6 decoder layers with a higher inner-layer dimension $d_{ff} = 2048$.

Pre-training on English ASR has shown to benefit ST [31]. Hence, we train two versions of all models - one trained from scratch and one that uses an encoder pre-trained on English ASR. All models are trained for 60k steps using Adam optimizer [32] with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. During training, the learning rate was varied according to the formula proposed in [9]: it was linearly increased in the warm-up stage and then decreased at each step by the factor of $1/\sqrt{steps}$ accordingly. We used 10k warmup steps. In addition, Gradient Clipping [33] with threshold 10.0, Label smoothing [34] with smoothing parameter $\epsilon = 0.1$, and Dropout [26] with $p = 0.1$ are employed.

Model evaluation is performed by averaging the last *N* validation checkpoints. We only report the best of the two results obtained by setting $N = 5$ and $N = 10$. At inference, we use beam search with a beam size of 5 and length penalty of 1. We report case-sensitive and detokenized BLEU scores [35] obtained using sacreBLEU [36].

4. Results

We compare our proposed model with two strong Transformer-based baselines. First, referred further as Wang et al. [27], is a vanilla Transformer and state-of-the-art (SOTA) on CoVoST 2 [27]. Second is the model proposed by Di Gangi et al. [18]. This model is based on the Transformer as well but has been adjusted for speech inputs: It uses Convolution-based 2D Attention mechanism [15] before the self-attention and applies a logarithmic distance penalty to encoder attention weights so as to bias the model towards local context.

4.1. Training from Scratch

We first compare the BLEU scores obtained by our proposed model with those obtained by Wang et al. [27] and Di Gangi et al. [18] when all the models are trained from scratch. In Table 1 under the heading Training from Scratch, we can see that our base model is able to improve previous SOTA results on six language directions and set new SOTA results on two language directions: De→En and Ru→En, that too with just 16 million parameters. It achieves a minimum of 1.5 to a maximum of 9.6 BLEU score improvements over Wang et al. [27] and a minimum of 2.1 to a maximum of 21.2 BLEU score improvements over Di Gangi et al. [18]. On the other hand, our deep model outperforms our base model and sets new SOTA results on five language directions: Fr→En, Es→En, It→En, Zh→En, and Pt→En. With just 25M parameters, it produces performance improvements ranging from 0.9 to 10.5 BLEU scores over Wang et al. [27] and from 2.1 to 22.4 BLEU score improvements over Di Gangi et al. [18]. We can assess the quality of our proposed models by looking at their perplexity curves. Figure 2 shows the perplexity curves for De→En. It can be seen that our models have lower perplexities than that of Wang et al. [27] and Di Gangi et al. [18].

4.2. Encoder ASR Pre-Training

We first compare the Word Error Rates (WER) for the encoder English ASR pre-training as shown in Table 2. Though our proposed models achieve a lower WER than Di Gangi et al. [18], they are still higher than Wang et al. [27]. These results suggest that the ASR WER may not be a good indicator of final ST performance. The BLEU scores obtained by all models trained using the pre-trained encoder weights are summarized in Table 1 under the heading Encoder ASR Pre-Training. We can see that our base model again outperforms the previous SOTA, but this time on all seven langauge directions. It shows gains from +0.1 to +2.4 BLEU scores over Wang et al. [27] and from +1.1 to +5.2 BLEU scores over Di Gangi et al. [18]. Our deep model yet again outperforms our base model to set new SOTA on six language directions: Fr→En, De→En, Es→En, It→En, Ru→En, and Pt→En. Compared to Wang et al. [27], our deep model shows 0.3-3.5 BLEU score improvements, even with a higher WER (26.89 ours vs 25.56 Wang et al. [27]). When compared with Gangi et al. [18], our deep model shows 1-5.6

Table 1: *BLEU ↑ scores for X→En Speech Translation on CoVoST 2 test set. Results in bold are where our model improves previous end-to-end state-of-the-art and new state-of-the-art results are shown as underlined. For each source language the amount of training data available (in hours) is indicated in round brackets.*

Training from Scratch									
Model	**Fr (264)**	**De (184)**	**Es (113)**	**It (44)**	**Ru (18)**	**Zh (10)**	**Pt (10)**	**Params**	**Enc Layers**
Wang et al. [27]	24.30	8.40	12.00	0.25	1.20	1.40	0.50	27M	12
Di Gangi et al. [18]	4.67	0.44	0.13	0.02	0.11	0.83	0.17	32M	6
Ours (base)	22.75	**12.65**	**21.40**	**8.21**	**10.80**	**2.95**	**2.28**	16M	6
Ours (deep)	**25.23**	8.07	**22.53**	**8.51**	**10.37**	**3.01**	**3.02**	25M	12
Encoder ASR Pre-Training									
Model	**Fr (264)**	**De (184)**	**Es (113)**	**It (44)**	**Ru (18)**	**Zh (10)**	**Pt (10)**	**Params**	**Enc. Layers**
Wang et al. [27]	25.05	17.58	22.28	11.14	14.92	5.80	6.10	27M	12
Di Gangi et al. [18]	24.05	15.98	21.18	7.90	11.81	5.08	5.03	32M	6
Ours (base)	**25.22**	**18.01**	**24.71**	**12.58**	**17.08**	**6.15**	**7.83**	16M	6
Ours (deep)	**27.26**	**20.01**	**25.82**	**13.55**	**17.94**	**6.10**	**9.02**	25M	12

BLEU score improvements.

5. Discussion

The vanilla Transformer model used by Wang et al. [27] only encodes non-local inductive bias, whereas our proposed model encodes both local and non-local inductive biases. As a result, we can see from the results that our proposed model is able to outperform it on all seven language directions not only when encoders are pre-trained on English ASR but also when the models are trained from scratch. In comparison with the model by Di Gangi et al. [18], which uses a 2D convolution-based attention and log distance penalty to capture local context, our model achieves better results on all seven language directions. These results strengthen our argument that - adjusting self-attention to encode local inductive bias is a sub-optimal method and will lead to sub-optimal solutions.

It is interesting to see that our deep model (trained from scratch) outperforms the SOTA results of Wang et al. [27] (uses pre-trained encoder) on two language directions: Fr→En and Es→En by 0.18 and 0.25 BLEU scores respectively.

Table 2: *WER results for Encoder ASR pre-training on Common Voice Corpus 4 English test set.*

Model	**WER ↓**
Di Gangi et al. [18]	33.73
Wang et al. [27]	25.56
Ours (base model)	29.38
Ours (deep)	26.89

6. Conclusions

We have shown that encoding both local and non-local inductive biases into the Transformer brings large performance gains in end-to-end ST task. Our proposed model, Conformer-Transformer, exploits both local and global context to learn meaningful representations from speech signals and outperforms strong Transformer-based baselines by large margins. Results on seven language directions warrant our hypothesis that indeed use of self-attention for encoding non-local inductive bias and convolution for encoding local inductive bias leads the learning algorithm to generalize to better solutions.

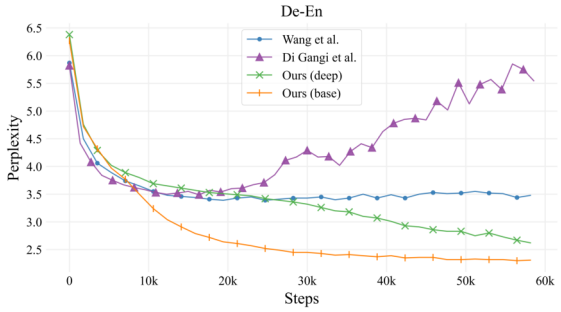

Figure 2: *Perplexity curves for German (De) → English (En) computed on CoVoST 2 validation set. All models are trained from scratch.*

7. References

[1] H. Ney, "Speech translation: coupling of recognition and translation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999, pp. 517–520.

[2] E. Matusov, S. Kanthak, and H. Ney, "On the integration of speech recognition and statistical machine translation," in *Proc. Interspeech*, 2005, pp. 3177–3180.

[3] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2013.

[4] G. Kumar, M. Post, D. Povey, and S. Khudanpur, "Some insights from translating conversational telephone speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3231–3235.

[5] G. Kumar, G. Blackwood, J. Trmal, D. Povey, and S. Khudanpur, "A coarse-grained model for optimal coupling of ASR and SMT systems for speech translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1902–1907.

[6] A. Bérard, O. Pietquin, L. Besacier, and C. Servan, "Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text

Translation," in *NIPS Workshop on end-to-end learning for speech and audio processing*, 2016.

[7] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Proc. Interspeech*, 2017, pp. 2625–2629.

[8] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6224–6228.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[10] L. Cross Vila, C. Escolano, J. A. R. Fonollosa, and M. R. Costa-Jussà, "End-to-End Speech Translation with the Transformer," in *Proc. IberSPEECH*, 2018, pp. 60–63.

[11] E. Ansari, A. Axelrod, N. Bach, O. Bojar, R. Cattoni, F. Dalvi, N. Durrani, M. Federico, C. Federmann, J. Gu, F. Huang, K. Knight, X. Ma, A. Nagesh, M. Negri, J. Niehues, J. Pino, E. Salesky, X. Shi, S. Stüker, M. Turchi, A. Waibel, and C. Wang, "FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN," in *Proceedings of the 17th International Conference on Spoken Language Translation*. Association for Computational Linguistics, 2020, pp. 1–34.

[12] T. M. Mitchell, "The need for biases in learning generalizations," Rutgers University, Tech. Rep., 1980.

[13] J. Baxter, "A model of inductive bias learning," *Journal of Artificial Intelligence Research*, vol. 12, p. 149–198, 2000.

[14] S. Abnar, M. Dehghani, and W. Zuidema, "Transferring inductive biases through knowledge distillation," *arXiv preprint arXiv:2006.00555*, 2020.

[15] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[16] J. Im and S. Cho, "Distance-based self-attention network for natural language inference," *arXiv preprint arXiv:1712.02047*, 2017.

[17] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, "Self-attentional acoustic models," *arXiv preprint arXiv:1803.09519*, 2018.

[18] M. A. Di Gangi, M. Negri, and M. Turchi, "Adapting transformer to end-to-end spoken language translation," in *Proc. Interspeech*, 2019, pp. 1133–1137.

[19] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[20] A. Graves, "Sequence transduction with recurrent neural networks," in *International Conference of Machine Learning (ICML) Workshop on Representation Learning*, 2012.

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

[22] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2020.

[23] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, 2017, pp. 933–941.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

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

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[27] C. Wang, A. Wu, and J. Pino, "Covost 2 and massively multilingual speech-to-text translation," *arXiv preprint arXiv:2007.10310*, 2020.

[28] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2020.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019.

[30] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 66–71.

[31] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," 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)*, 2019, pp. 58–68.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.

[33] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 1310–1318.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015.

[35] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[36] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, 2018, pp. 186–191.