

How do Hyenas deal with Human Speech?

Speech Recognition and Translation with ConfHyena

Marco Gaido 🐶🐶, Sara Papi 🐶🐶, Matteo Negri 🐶, Luisa Bentivogli 🐶

🐶 Fondazione Bruno Kessler, Italy

{mgaido, spapi, negri, bentivo}@fbk.eu

Abstract

The attention mechanism, a cornerstone of state-of-the-art neural models, faces computational hurdles in processing long sequences due to its quadratic complexity. Consequently, research efforts in the last few years focused on finding more efficient alternatives. Among them, Hyena (Poli et al., 2023) stands out for achieving competitive results in both language modeling and image classification, while offering sub-quadratic memory and computational complexity. Building on these promising results, we propose ConfHyena, a Conformer whose encoder self-attentions are replaced with an adaptation of Hyena for speech processing, where the long input sequences cause high computational costs. Through experiments in automatic speech recognition (for English) and translation (from English into 8 target languages), we show that our best ConfHyena model significantly reduces the training time by 27%, at the cost of minimal quality degradation ($\sim 1\%$), which, in most cases, is not statistically significant.

Keywords: speech, recognition, translation, ASR, ST, Hyena, operator, convolutions, complexity

1. Introduction

The attention mechanism (Bahdanau et al., 2016a) is the core of today’s neural architectures in many AI fields (Lin et al., 2022), including speech processing (Latif et al., 2023). However, attention is known to be computationally expensive due to its quadratic computational and memory complexity with respect to the input length, which hinders the adoption of attention-based models in use cases that entail long input sequences. This has complicated their application in speech tasks like automatic speech recognition (ASR) and speech translation (ST), where audio sequences are typically ~ 8 -10 times longer than the equivalent text. In fact, current models have to downsample the input sequence by a factor of 4 with strided convolutions before applying attention-based layers (Bérard et al., 2018; Di Gangi et al., 2019). Besides causing information loss (Salesky et al., 2019; Papi et al., 2021), this workaround does not entirely solve the problem. The sequences can indeed remain very long, thereby resulting in high training costs with important social and environmental negative impact (Strubell et al., 2019).

While attention alternatives with sub-quadratic complexity have been proposed in other fields (Tay et al., 2022), none of them have achieved widespread adoption, as their efficiency comes at the cost of non-negligible performance degradation in many tasks, including ASR and ST (Alastruey et al., 2021). More recently, Poli et al. (2023) introduced the Hyena operator. Based on a recurrence of implicitly parametrized long convolutions and

data-controlled gating, Hyena has been the first attention alternative to claim comparable results (in language modeling and image classification). Building upon these promising results, in this paper, we explore the adaptation and application of Hyena to ASR and ST, two tasks where training state-of-the-art models (Zhang et al., 2023; Radford et al., 2023) is extremely demanding in terms of computational resources, hardware, and time.¹

Along this line of work, our contributions are:

1. We adapt the original Hyena operator, which is *causal* (i.e., it prevents accessing future information when encoding each element of a sequence), into a *non-causal* version, allowing for richer representations by accessing information across the entire input sequence;
2. We present two models, ConfHyena and Hybrid ConfHyena,² based on the state-of-the-art Conformer (Gulati et al., 2020) architecture, which replace the self-attention with our adapted Hyena operator in *all* encoder layers and *only* in initial layers, respectively;
3. We show that on ASR (en) and ST (en \rightarrow {de, es, fr, it, nl, pt, ro, ru}), our Hybrid ConfHyena reduces training time by 27%, at the cost of a minimal ($\sim 1\%$) and mostly not statistically significant quality degradation.

¹Attempts to build models similar to Whisper (Radford et al., 2023), even on a smaller training dataset (less than 25% of Whisper), require more than 15,000 GPU/hours on NVIDIA A100 GPUs (Peng et al., 2023b).

²Code and pre-trained models are released open-source under Apache 2.0 license at: <https://github.com/hlt-mt/FBK-fairseq>.



The authors contributed equally.

2. Background

2.1. Self-Attention

Attention (Bahdanau et al., 2016b) is a function that maps a query matrix Q and a pair of key-value matrices K and V to an output sequence $A(Q, K, V)$. In the case of self-attention, Q , K , and V are computed from the same input sequence $x \in \mathbb{R}^{L \times d}$ through three different trainable matrices $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ as $Q = xW_q$, $K = xW_k$, and $V = xW_v$. The scaled dot-product attention (Chan et al., 2016) used in the Transformer (Vaswani et al., 2017) is then obtained by:

$$A(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

Since the matrix multiplication between $Q \in \mathbb{R}^{L \times d}$ and $K^T \in \mathbb{R}^{d \times L}$ results in a $\mathbb{R}^{L \times L}$ matrix, its overall memory and time complexity is $\mathcal{O}(L^2)$.

2.2. Conformer

The self-attention is a cornerstone not only of the Transformer but also of the more recent Conformer (Gulati et al., 2020), an architecture tailored for speech processing that significantly outperformed the Transformer in both ASR and ST (Inaguma et al., 2021). The Conformer modifies the encoder layer structure of the Transformer by implementing three key modifications. First, it adds relative sinusoidal positional encodings (Dai et al., 2019) into the self-attention computation, eliminating the absolute positional embeddings added to the input in the Transformer. Second, it replaces the Transformer feed-forward network (FFN) with a Macaron-Net (Lu et al., 2019) consisting of two FFNs that wrap the module. Lastly, it introduces a new convolution module after the self-attention. This module is made of a pointwise convolution, a Gated Linear Unit (GLU) activation function (Dauphin et al., 2017), a depthwise convolution, a batch normalization (Ioffe and Szegedy, 2015), a Swish activation function (Ramachandran et al., 2017), and another pointwise convolution.

2.3. Hyena

Hyena (Poli et al., 2023) has been introduced as an alternative to self-attention with the goal of achieving the same representativeness (i.e., the ability to model dependencies between time steps of the input, regardless of their distance) while avoiding the quadratic complexity w.r.t. the input length. The core of this operator is a recurrence of two operations: an implicitly parametrized long convolution (i.e., a convolution where the kernel has the same size as the input), and an element-wise multiplicative gating (i.e., an element-wise multiplication with

a matrix of learned parameters). Hyena can be formalized as:

$$\begin{aligned} u_{0, \dots, N}, z_0 &= \text{ShortConv}(W_{0, \dots, N+1}x) \\ z_{i+1} &= u_i \odot \text{LongConv}_i(z_i), i \in [0, N) \\ y &= z_N \end{aligned}$$

where x is the input sequence, $W_{0, \dots, N+1}$ are learned weights and N is a hyperparameter named *order*, which controls the depth of the recurrence and is usually set to 2.

The short convolution (ShortConv) models short-term dependencies and is a Conv1D with kernel size 3 and stride 1, which does not alter the shape of the input. The long convolution (LongConv) models long-range dependencies and its efficient implementation is the key to the sub-quadratic complexity of the Hyena operator. A naive implementation of the LongConv would require learning a kernel of the same size as the input, which slides over the padded input L times. This approach would result in a quadratic complexity, akin to self-attention. In Hyena, instead, the kernel is implicitly learned by applying a FFN over complex exponential positional embeddings (Wang et al., 2020a), making the number of the kernel parameters independent from the input length. Furthermore, Hyena bases its LongConv implementation on two principles: *i*) the convolution theorem (Bracewell and Kahn, 1966), which states that a circular convolution corresponds to an element-wise multiplication in the discrete Fourier domain (Oppenheim and Schaffer, 2009), and *ii*) the equivalence between a linear convolution over a sequence of length L with a kernel k of length K and a circular convolution with a kernel k zero-padded to length $K + L$ (Oppenheim and Schaffer, 1975). Building on them, Hyena computes the $\text{LongConv}_i(x)$ as:

$$\begin{aligned} z &= \text{iFFT}(\text{FFT}(\text{pad}(x)) \odot \text{FFT}(\text{pad}(k_i))) \\ y &= z[:L] \end{aligned} \quad (1)$$

where FFT is the Fast Fourier Transform (Selesnick and Burrus, 1997), iFFT is the inverse FFT and $\text{pad}(\cdot)$ pads the input with 0s up to a $2L$ length. The output selection $z[:L]$ preserves *causality* (i.e., the output at a given position depends only on the past), which is necessary for autoregressive models such as Transformer decoders. The resulting LongConv and, in turn, Hyena implementation has a sub-quadratic computational complexity of $\mathcal{O}(L \log_2 L)$. While replacing the self-attention with sub-quadratic solutions based on the Fourier transform is not a new proposal (Lee-Thorp et al., 2022), Hyena has been the first to claim no performance loss, which makes it a promising alternative to self-attention with quadratic complexity.

Model	ASR (en)	en-de	en-es	en-fr	en-it	en-nl	en-pt	en-ro	en-ru	ST avg
Conformer	10.64	24.97	30.48	36.43	26.25	30.31	30.09	24.67	17.35	27.57
ConfHyena	10.88	24.90	30.28	35.71*	25.69*	29.49*	30.51*	23.93*	17.34	27.23
- non-causal Hyena	10.99*	24.28	29.42*	35.57*	25.38*	29.55*	29.99	23.92*	16.99*	26.89
Hybrid ConfHyena	10.75	25.22	30.15*	36.19	26.04	29.76*	30.43	23.78*	17.06	27.33
- non-causal Hyena	11.27*	24.84	30.28	36.09	25.81*	29.53*	30.69*	23.96*	16.98*	27.27

Table 1: ConfHyena and Hybrid ConfHyena results on MuST-C v1.0 tst-COMMON for all language pairs. ASR scores are WER \downarrow while ST scores are BLEU \uparrow . * means that the difference with the baseline (Conformer) is statistically significant as per bootstrap resampling (Koehn, 2004) with 95% CI.

3. Hyena for Speech Processing

As just seen, the original Hyena operator is built to preserve the causality property. However, in the context of speech processing, constraining the encoder to access only past elements could reduce performance, as it is typically designed to look at the entire sequence to create context-aware encodings from the complete input representation (Chorowski et al., 2015). For this reason, we introduce **ConfHyena**, a Conformer-based model (§2.2) where encoder self-attentions are replaced by a non-causal version of the Hyena operator that can access the entire input sequence. The differences with the causal version of Hyena are two:

1. the ShortConv operator processes the current, preceding, and following time frames at each step, rather than the current and the two preceding frames. This is realized by setting the left padding to $\lceil \frac{K}{2} \rceil$ instead of $K - 1$;
2. the LongConv operator is similarly adjusted, with the non-causal version implemented by altering the selection in Eq. 1 to $y = z[L/2 : -L/2]$ to select the portion of the circular convolution that corresponds to a linear convolution with *same* padding (Goodfellow et al., 2016).

With such modifications, ConfHyena has access to the same information as the self-attention in speech-processing models, while retaining a lower computational and memory complexity. The lower *computational* complexity entails faster computation of long input sequences. The lower *memory* complexity, instead, enables ConfHyena to process longer inputs compared to attention-based models without incurring out-of-memory issues and to run on GPUs with less VRAM.

In addition, we integrate a CTC-compression module (Gaido et al., 2021) into the middle of the encoder of all our systems. This module is based on the connectionist temporal classification or CTC (Graves et al., 2006) and collapses intermediate context-aware encodings to reduce their redundancy, making them more effectively encoded by successive layers, with benefits in terms of both output quality and efficiency (Liu et al., 2020; Zhang

et al., 2020; Zhao et al., 2022). As the sequences resulting from this module are significantly shorter, we posit that their length is sufficiently small so that they can be efficiently processed by self-attention. As such, we propose **Hybrid ConfHyena**, an architecture where the non-causal Hyena operator is introduced only in the layers before the CTC compression while preserving the self-attention in the subsequent ones.

4. Experimental Settings

For both ASR and ST, we use the MuST-C v1.0 dataset (Cattoni et al., 2021), comprising (*audio, transcription, translation*) triplets in the TED-talks domain. Audios and transcriptions are in English while translations cover 8 languages (de, es, fr, it, nl, pt, ro, ru). For ASR, we use the en-es section.

The input is represented by 80-dimensional log mel-filterbank features (on 25ms windows sliding every 10ms) that are processed by 2 layers of CNN with stride 2, realizing a total downsampling of 4. For all models, we use 12 encoder layers, 6 decoder layers, 8 heads, an embedding size of 512, and 2048 FFN width. The kernel size of both point- and depth-wise convolutions in Conformer layers is set to 31. In Hyena operators, the order is set to 2, the width is 3 times the embedding size, and the filter FFN has 4 layers with 64 neurons and sine activations. Dropout is set to 0.1 for all models.

To ensure the reliability of our findings, we used a Conformer implementation that is padding-safe (i.e., it does not change the output according to the amount of padding) and we tested our Hyena implementation with `pangolin` to ensure the same property (Papi et al., 2023).

We train with Adam optimizer, label-smoothed cross-entropy loss (smoothing factor 0.1), and CTC loss (Graves et al., 2006) with weight factor 0.5 to ease convergence. We apply CTC compression after the 8th encoder layer. The learning rate is 2e-3 with Noam scheduler (Vaswani et al., 2017) and 25,000 warm-up steps. SentencePiece unigram vocabularies (Kudo and Richardson, 2018) are used with size 5,000 for English and 8,000 for all target languages. We early stop the training after 10 epochs without improvement on the dev loss

and average 5 checkpoints around the best. We train on 2 NVIDIA A40 40GB VRAM GPUs with 40k tokens per batch and 4 as update frequency and generate with 1 NVIDIA K80 16GB VRAM GPU. All other settings are the default of Fairseq-ST (Wang et al., 2020b). Evaluation is performed using WER for ASR and sacreBLEU (Post, 2018)³ for ST.

5. Results

For a comprehensive evaluation, in this section, we first analyze the output quality of ConfHyena and Hybrid ConfHyena (§5.1), and then we study their efficiency (§5.2), as both aspects are critical in determining the success of an architecture.

5.1. Output Quality

Table 1 compares our ConfHyena and Hybrid ConfHyena models with Conformer in terms of transcription (ASR) and translation (ST) quality. Through ablation tests, we also report the results of the proposed architectures with the original causal Hyena operator (- *non-causal Hyena*).

Focusing on the effect of the causality property, the results confirm its negative impact on model performance. Indeed, the causal versions of both ConfHyena and Hybrid ConfHyena are consistently outperformed by those equipped with the non-causal operator both in ASR and ST. The difference is more pronounced in ASR, where causality produces statistically significant degradations up to 0.52 WER. This holds also for ST, albeit with less significant gaps (on average over the 8 language pairs, -0.34 and -0.06 BLEU for ConfHyena and Hybrid ConfHyena, respectively). These results confirm the usefulness of accessing the entire input sequence to maximize transcription and translation quality and, therefore, **the superiority of the non-causal Hyena** in speech encoders.

Comparing ConfHyena and Hybrid ConfHyena, the latter is superior both in terms of transcription and translation quality. The gap between the two models, however, is marginal (-0.13 WER, +0.10 BLEU), showing that **the Hybrid variant has similar quality as the full ConfHyena encoder**.

Comparing Hybrid ConfHyena with Conformer, we observe that Conformer achieves the best results in most cases (ASR, and 6 out of 8 ST directions, with en-de and en-pt being the exceptions). However, although Conformer consistently yields the best average score of 27.57 BLEU in ST, it is worth remarking that the overall margin over Hybrid ConfHyena is very narrow (-0.11 WER and +0.24 BLEU on average), corresponding to a $\sim 1\%$ relative difference. Moreover, the difference is statistically significant only in 3 ST directions. We can

³BLEU|c:mixed|e:no|tok:13a|s:exp|v:2.0.0

Model	# param.	train time↓	inf time↓
Conformer	114.9M	×1.00	×1.00
ConfHyena	112.0M	×1.04	×0.95
Hybrid ConfHyena	112.9M	× 0.73	× 0.93

Table 2: Relative inference and training time averaged over all languages and tasks of Conformer, ConfHyena, and Hybrid ConfHyena.

therefore conclude that, in terms of mere output quality, **Hybrid ConfHyena performs closely to the attention-based Conformer model**.

5.2. Training and Inference Efficiency

After establishing that the output quality of the Conformer and ConfHyena models is similar, we now turn to assess their efficiency. Looking at Table 2, we notice that, in these terms, the differences are instead significant.

Notably, Hybrid ConfHyena emerges as the most efficient architecture by a large margin, reducing both training time by 27% and inference time by 7% in comparison to the Conformer model. The greater savings in training time can be attributed to the autoregressive nature of the models. In fact, while during training the number of forward (and backward) passes on the encoder and the decoder are the same (one per batch), during inference the encoder performs a single forward pass while most of the time is taken up by the multiple forward passes on the autoregressive decoder.

The training and inference time of ConfHyena is, instead, comparable to that of the Conformer, and much higher than that of Hybrid ConfHyena. Although this may seem counterintuitive, the explanation is straightforward: the sub-quadratic complexity of Hyena makes it more efficient when sequences are long, while for shorter sequences, such as those obtained after the CTC compression, attention is faster.

In summary, we can conclude that **Hybrid ConfHyena substantially reduces computational costs compared to Conformer and ConfHyena**. Drawing on this and the comparable output quality of the three models, we can state that **Hybrid ConfHyena offers the most favorable balance between quality and efficiency**.

6. Reducing Downsampling

The lower memory complexity of Hyena (as discussed in §2.3) opens up the possibility of reducing the initial downsampling performed by the two initial convolutions in speech encoders without incurring out-of-memory issues. A potential advantage of this operation would lie in mitigating the information

loss caused by the context-uninformed compression of the input speech sequence. which is inherent to downsampling (see §1). For this reason, we conclude this work by investigating the potential advantage of halving downsampling from a factor of 4 to a factor of 2. In practice, this is obtained using stride 1 in the first convolution, while keeping stride 2 in the second.

The results in Table 3 show that there is no advantage in reducing the initial downsampling: in fact, halving it does not increase the translation ability of our models. Rather, this operation inflates both training and inference time by up to 50% and 35%, respectively. In light of this, we believe that future works should pursue other directions (e.g., different hyperparameters) to close the small quality gap without losing the efficiency gains.

Model	BLEU \uparrow	train time \downarrow	inf time \downarrow
ConfHyena	24.90	$\times 1.00$	$\times 1.00$
+ downsample 2	24.79	$\times 1.39$	$\times 1.35$
Hybrid ConfHyena	25.22	$\times 1.00$	$\times 1.00$
+ downsample 2	25.22	$\times 1.50$	$\times 1.13$

Table 3: BLEU score and relative training/inference time of ConfHyena and Hybrid ConfHyena with downsample 2 on MuST-C en-de.

7. Conclusions

In order to reduce the high computational costs of ASR and ST models, in this paper we proposed ConfHyena, a Conformer-based model that replaces self-attentions with non-causal Hyena operators, and a Hybrid version that mixes ConfHyena and Conformer layers. Our experiments on English ASR and 8 ST directions demonstrated that Hybrid ConfHyena has the best quality/efficiency trade-off, as it significantly reduces training time by 27% with a minimal quality degradation of $\sim 1\%$ compared to the Conformer model.

Limitations

The “good results” conundrum. By proposing and experimenting with Hybrid ConfHyena, we introduced an alternative architecture that improves model efficiency without substantial performance degradation. Admittedly, in our experiments, we recognize that Hybrid ConfHyena exhibits a gap, albeit limited, with respect to the current, state-of-the-art Conformer model. However, we refrain from interpreting our results as inherently “negative” or from considering a minor performance gap as an invalidating limitation. Indeed, we echo the recent criticism of purely leaderboard-based evaluation of new systems (Ethayarajh and Jurafsky, 2020),

advocating for a more comprehensive perspective that considers a wider range of factors, including efficiency, ethics, and environmental sustainability (Wynsberghe, 2021). As such, we believe that achieving higher scores should not be the sole objective of research efforts, especially if it comes at the cost of resource-intensive training procedures (Ligozat et al., 2021) that sacrifice efficiency (Peng et al., 2023a). This approach also aligns with best practices recently adopted by major corporations that prioritize cost-effectiveness, environmental sustainability (Rolnick et al., 2022), and accessibility in model design (de Laat, 2021) to reduce their impact and democratize their use. Therefore, we underscore that the notable reduction in complexity and, in turn, in training time achieved by Hybrid ConfHyena should definitely be accounted as a positive advancement, regardless of the minimal quality degradation it implies.

Portability and scalability. The validity of our findings should be confirmed across a wider range of datasets since our experiments only focused on the MuST-C corpus, which is the typical resource used in recent ST research. The robustness of our results across different domains (MuST-C is composed of TED talks) and source languages (limited to English) have to be further verified, although there is no reason to believe that other settings may be more or less favorable to our proposed architecture. Furthermore, we did not scale our experiments to large training data, such as those employed in the training of state-of-the-art models like Whisper (Radford et al., 2023), which are $\sim 100-1,000$ times larger than MuST-C. Unfortunately, conducting such experiments is extremely expensive and demands access to high-performance hardware. Evaluating the savings achievable by our proposed model in such a scenario is an interesting future step for this research, as well as confirming that the performance gap it suffers from remains limited.

Efficient Attention Implementation. In our work we did not use efficient attention implementations, such as Flash Attention (Dao et al., 2022). While these engineering optimizations can significantly reduce resource requirements and speed up computation, they are hardware-specific. For instance, the Flash Attention is not supported by the K80 GPUs used in our experimental settings. For this reason, we address the efficiency of the attention mechanism from a theoretical perspective, whose benefits are maintained regardless of the hardware employed.

Acknowledgements

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

Bibliographical References

- Belen Alastruey, Gerard I. Gállego, and Marta R. Costa-jussà. 2021. [Efficient transformer for direct speech translation](#).
- Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philémon Brakel, and Yoshua Bengio. 2016a. [End-to-end attention-based large vocabulary speech recognition](#). In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949.
- Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philémon Brakel, and Yoshua Bengio. 2016b. [End-to-end attention-based large vocabulary speech recognition](#). In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949.
- Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-End Automatic Speech Translation of Audiobooks. In *Proceedings of ICASSP 2018 - IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, Alberta, Canada.
- Ron Bracewell and Peter B. Kahn. 1966. [The Fourier Transform and Its Applications](#). *American Journal of Physics*, 34(8):712–712.
- Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. [Must-c: A multilingual corpus for end-to-end speech translation](#). *Computer Speech & Language*, 66:101155.
- 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.
- 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.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. [Transformer-XL: Attentive language models beyond a fixed-length context](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*.
- Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17*, page 933–941. JMLR.org.
- Paul de Laat. 2021. [Companies committed to responsible ai: From principles towards implementation and regulation? Philosophy & Technology](#), 34.
- Mattia A. Di Gangi, Matteo Negri, Roldano Cattoni, Roberto Dessi, and Marco Turchi. 2019. Enhancing Transformer for End-to-end Speech-to-Text Translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 21–31, Dublin, Ireland. European Association for Machine Translation.
- Kawin Ethayarajh and Dan Jurafsky. 2020. [Utility is in the eye of the user: A critique of NLP leaderboards](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853, Online. Association for Computational Linguistics.
- 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, Online.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Convolutional networks. In *Deep learning*, chapter 9, page 330–372. MIT Press, Cambridge, MA, USA.
- Alex Graves, Santiago Fernández, Faustino J. 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 (ICML)*, pages 369–376, Pittsburgh, Pennsylvania.

- Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. [Conformer: Convolution-augmented Transformer for Speech Recognition](#). In *Proc. Interspeech 2020*, pages 5036–5040.
- Hirofumi Inaguma, Yosuke Higuchi, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. 2021. Non-autoregressive end-to-end speech translation with parallel autoregressive rescoring. *arXiv preprint arXiv:2109.04411*.
- Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15*, page 448–456. JMLR.org.
- Philipp Koehn. 2004. [Statistical significance tests for machine translation evaluation](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.
- 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, Brussels, Belgium.
- Siddique Latif, Aun Zaidi, Heriberto Cuayahuitl, Fahad Shamshad, Moazzam Shoukat, and Junaid Qadir. 2023. Transformers in speech processing: A survey. *arXiv preprint arXiv:2303.11607*.
- James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. [FNet: Mixing tokens with Fourier transforms](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.
- Anne-Laure Ligozat, Julien Lefèvre, Aurélie Bugeau, and Jacques Combaz. 2021. Unraveling the hidden environmental impacts of ai solutions for environment. *arXiv preprint arXiv:2110.11822*.
- Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI Open*.
- Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. [Bridging the modality gap for speech-to-text translation](#).
- Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. [Understanding and improving transformer from a multi-particle dynamic system point of view](#).
- Alan V. Oppenheim and Ronald W. Schafer. 1975. *Digital Signal Processing*. Prentice Hall international editions. Prentice-Hall.
- Alan V. Oppenheim and Ronald W. Schafer. 2009. *Discrete-Time Signal Processing*, 3rd edition. Prentice Hall Press, USA.
- Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2021. [Speechformer: Reducing information loss in direct speech translation](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1698–1706, Online and Punta Cana, Dominican Republic.
- Sara Papi, Marco Gaido, Andrea Pilzer, and Matteo Negri. 2023. [When Good and Reproducible Results are a Giant with Feet of Clay: The Importance of Software Quality in NLP](#).
- Hao Peng, Qingqing Cao, Jesse Dodge, Matthew E. Peters, Jared Fernandez, Tom Sherborne, Kyle Lo, Sam Skjonsberg, Emma Strubell, Darrell Plessas, Iz Beltagy, Evan Pete Walsh, Noah A. Smith, and Hannaneh Hajishirzi. 2023a. [Efficiency Pentathlon: A Standardized Arena for Efficiency Evaluation](#).
- Yifan Peng, Jinchuan Tian, Brian Yan, Dan Berrebbi, Xuankai Chang, Xinjian Li, Jiatong Shi, Siddhant Arora, William Chen, Roshan Sharma, Wangyou Zhang, Yui Sudo, Muhammad Shakeel, Jee weon Jung, Soumi Maiti, and Shinji Watanabe. 2023b. [Reproducing whisper-style training using an open-source toolkit and publicly available data](#).
- Michael Poli, Stefano Massaroli, Eric Q. Nguyen, Daniel Y. Fu, Tri Dao, Stephen A. Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*.
- 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.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. [Robust Speech Recognition via Large-Scale Weak Supervision](#). In *Proceedings of the 40th International Conference on Machine*

- Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.
- Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. [Searching for activation functions](#).
- David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. 2022. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96.
- Elizabeth Salesky, Matthias Sperber, and Alan W. Black. 2019. [Exploring Phoneme-Level Speech Representations for End-to-End Speech Translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1835–1841, Florence, Italy. Association for Computational Linguistics.
- Ivan W Selesnick and C Sidney Burrus. 1997. *Fast Convolution and Filtering*. CRC Press.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.
- Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. [Efficient transformers: A survey](#). *ACM Comput. Surv.*, 55(6).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Benyou Wang, Donghao Zhao, Christina Lioma, Qiu-uchi Li, Peng Zhang, and Jakob Grue Simonsen. 2020a. [Encoding word order in complex embeddings](#). In *International Conference on Learning Representations*.
- Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020b. fairseq s2t: Fast speech-to-text modeling with fairseq. In *Proceedings of the 2020 Conference of the Asian Chapter of the Association for Computational Linguistics (ACL): System Demonstrations*.
- Aimee Wynsberghe. 2021. [Sustainable ai: Ai for sustainability and the sustainability of ai](#). *AI and Ethics*, 1.
- Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2020. [Adaptive feature selection for end-to-end speech translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2533–2544, Online. Association for Computational Linguistics.
- Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*.
- Jinming Zhao, Hao Yang, Gholamreza Haffari, and Ehsan Shareghi. 2022. [RedApt: An adaptor for wav2vec 2 encoding faster and smaller speech translation without quality compromise](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1960–1967, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.