RESOURCE-EFFICIENT SEPARATION TRANSFORMER

Luca Della Libera∗1, Cem Subakan∗2,1,3, Mirco Ravanelli1,3, Samuele Cornell4, Frédéric Lepoutre5, François Grondin6

1Concordia University, 2Université Laval, 3Mila-Quebec AI Institute, 4Università Politecnica delle Marche, 5Soundskirt Inc., 6Université de Sherbrooke

ABSTRACT

Transformers have recently achieved state-of-the-art performance in speech separation. These models, however, are computationally demanding and require a lot of learnable parameters. This paper explores Transformer-based speech separation with a reduced computational cost. Our main contribution is the development of the Resource-Efficient Separation Transformer (RE-SepFormer), a self-attention-based architecture that reduces the computational burden in two ways. First, it uses non-overlapping blocks in the latent space. Second, it operates on compact latent summaries calculated from each chunk. The RE-SepFormer reaches a competitive performance on the popular WSJ0-2Mix and WHAM! datasets in both causal and non-causal settings. Remarkably, it scales significantly better than the previous Transformer-based architectures in terms of memory and inference time, making it more suitable for processing long mixtures.

Index Terms—Efficient speech separation, Transformer, self-attention, deep learning.

1. INTRODUCTION

In recent years, deep learning has become more and more computationally demanding. The current trend consists of improving the performance of Deep Neural Networks (DNNs) through ever-larger models crunching ever-larger amounts of data. In natural language processing (NLP), this tendency led to large language models such as GPT3 [1], PaLM [2], Megatron [3] and many others. Similarly, large neural networks like wav2vec2.0 [4] and HuBERT [5] have gained popularity for speech processing. Large neural models, however, are energy-demanding, causing high inference costs (with considerable CO2 emissions [6]) that limit their widespread adoption in production systems. Moreover, such models cannot process users’ data on the device, thus raising privacy concerns [7]. Efficient deep learning has been the object of recent research efforts. Approaches such as distillation [8], neural network pruning [9,10], and binarization/quantization [11,12] have been explored. In the context of speech processing, various efficient methods for speech enhancement/separation [13–19], keyword spotting, language identification [20], emotion recognition [21], and automatic speech recognition [22] have been proposed.

Among all the popular neural models, Transformers [23] are particularly difficult to make computationally efficient due to their quadratic memory bottleneck and the high number of parameters that they typically require. Revised Transformer-based architectures, which relax the quadratic memory requirement, [24] such as the Linformer [25], Longformer [26], and Reformer [27] have been proposed recently. Efficient Transformers have been studied for speech processing tasks as well, such as speech recognition [28], enhancement, and separation [18]. These previous works, however, do not consider causal models suitable for on-device real-time applications.

In this paper, we propose a novel small-footprint speech separation model built upon the SkiM framework [19], called Resource-Efficient Separation Transformer (RE-SepFormer), that represents a lightweight alternative to the recently-proposed SepFormer. Unlike the SepFormer, the RE-SepFormer uses non-overlapping chunks in the latent space and, therefore, it reduces by half the number of chunks to process (if we consider the default overlap rate of 50%). Moreover, we further reduce the computations by using a special mechanism called Memory Transformer. The Memory Transformer operates over a summary representation calculated from whole chunks rather than attending every single element of the chunk independently.

We conducted the experimental validation on the popular WSJ0-2Mix dataset. To assess our model in more realistic conditions, we also considered the WHAM! [29] dataset, which contains mixtures corrupted by non-stationary environmental noise. In addition to the non-causal offline scenario we provide experimental evidence in a real-time low-latency scenario by considering causal versions of the RE-SepFormer. In detail, our contributions are the following:

• We show that the RE-SepFormer reaches a competitive performance on the popular WSJ0-2Mix and WHAM! datasets in both causal and non-causal settings.

• With the RE-SepFormer, we achieve a 3x reduction of the parameters and 11x reduction of the multiply-accumulate operations (MACs) per second over the standard SepFormer.

• The RE-SepFormer is extremely parallelizable, and therefore highly suitable for GPU inference. It scales significantly better than the previous Transformer-based architectures in terms of memory and inference time, making it more suitable for processing long mixtures.

2. RE-SEPFORMER

2.1. Time-Domain Masking Architecture

The overall architecture of the RE-SepFormer is shown in Fig. 1. Similar to other popular architectures such as Conv-TasNet [15], Dual-Path RNN [16], and SepFormer [30], our model is based on learned-domain masking. The input mixture \( x \in \mathbb{R}^T \) is first passed through an encoder, which outputs a latent representation \( h \in \mathbb{R}^{T' \times F} \) using a strided convolutional layer:

\[
h = \text{ReLU} (\text{conv1d}(x)).
\] (1)

The masking network takes in the representation \( h \), and produces the masks \( m_1, m_2 \in \mathbb{R}^{T' \times F} \). We consider here a model with two
2.2. The Masking Network

The architecture of the masking network is depicted in Fig. 2 (top). Firstly, the input representation is split into temporal chunks. We denote this tensor with $h' \in \mathbb{R}^{C \times N_c \times F}$, where $C$ is the size of each chunk, and $N_c$ is the resulting number of chunks. As opposed to the standard SepFormer, we use non-overlapping chunks to reduce the amount of computations. Then, the RE-SepFormer processes the chunks $h'$ and provides a latent representation $h'' \in \mathbb{R}^{C' \times N_c \times F}$ (where $N_c$ denotes the number of sources). The tensor $h''$ is further transformed by a PReLU activation function and a linear layer. Finally, the chunking is undone by simply concatenating the chunks on the original time axis. We denote the resulting tensor with $h''' \in \mathbb{R}^{T/F \times N_c \times F}$. The final masks are estimated by passing this tensor through a ReLU non-linearity.

2.3. Resource Efficient SepFormer

The RE-SepFormer block is detailed in Fig. 2 (bottom). This module includes three main components: the IntraTransformer1, the MemoryTransformer, and the IntraTransformer2. The IntraTransformer1 is applied to the time axis of all of the chunks and generates a tensor $e_1 \in \mathbb{R}^{C \times N_c \times F}$. The goal of this block is to process short-term temporal information. We then compute a summary representation $e_2 \in \mathbb{R}^{N_c \times F}$ by averaging this tensor over the time axis. The intuition behind this operation is that the average over the time-axis of a latent representation can provide enough high-level contextual information to embed longer-term dependencies. Working with a summary vector is much more computationally convenient than operating on the full tensor $e_1$ as done in the original SepFormer. This saves significant amounts of computations.

This summary representation $e_2$ is then fed into the Memory Transformer. The latter is applied to the chunk axis and produces a representation $e_3 \in \mathbb{R}^{N_c \times F}$ that models long-term dependencies across chunks. The $e_3$ tensor is then added element-wise to $e_1$ (with broadcasting over the time axis). The resulting $e_4$ tensor is particularly rich as it incorporates both short and long-term dependencies. Note also that this operation implicitly adds a gradient shortcut in the architecture, contributing to making the architecture easier to train and more robust against vanishing gradient issues.

Finally, we provide more capacity to the model by feeding $e_4$ into another IntraTransformer2 operating on the time axis. This generates the tensor $h''' \in \mathbb{R}^{C'' \times N_c \times F}$, that is the output of the RE-SepFormer block.
The other proposed intervention is the summary representation (time average) processed by a Memory Transformer. This operation drastically reduces the number of parameters (3.2x reduction) and the MACs (11x reduction). Interestingly, this modification not only does not deteriorate the separation performance but even yields a slight improvement, possibly due to the fact that the averaging operation better promotes continuity between the latent chunks. We believe that the RE-SepFormer is particularly interesting for small-footprint devices. Even though the performance drop is not negligible compared to the standard SepFormer, the model still provides very high-quality speech separation (SDRi up to 18.9 dB) with a drastic reduction of computational resources.

### 4.2. Comparison with SkiM

Table 2 compares the performance of the RE-SepFormer with SkiM [19] on WSJ0-2Mix and WHAM! in causal and non-causal settings. SkiM [19] is a recently proposed model for efficient speech separation. It uses RNNs with non-overlapping chunks, making it a natural benchmark. For a fair comparison, we use dynamic mixing and the same kernel size in the convolutional layers (kernelsize = 16) for both models.

As shown in Table 2, the RE-SepFormer outperforms SkiM in three of the four tested conditions: it provides better performance on the WSJ0-2Mix dataset (both causal and non-causal settings) and on the WHAM! corpus (causal setting). SkiM, on the other hand, slightly outperforms the RE-SepFormer in the non-causal modality only. SkiM also uses fewer MACs/s than the RE-SepFormer (3.7G versus 6.3G). However, as we will show in the next section, this has no impact on latency. Remarkably, even when handling long sequences, RE-SepFormer matches SkiM in terms of memory usage and inference speed.

### 4.3. Speed and Memory Utilization

In Fig. 3, we compare the memory usage (left) and the inference time (right) of the RE-SepFormer, SkiM, and SepFormer. For a more meaningful comparison, we use a SepFormer (denoted as SepFormer-Light) with a reduced number of parameters (i.e., 6.4 M). This experiment has been conducted on an NVIDIA A100 GPU considering different input lengths (ranging from 1 to 256 seconds).

The RE-SepFormer turned out to scale significantly better than the SepFormer-Light due to the summary representation. The most impressive result is the inference time observed when feeding the RE-SepFormer with long sequences. For an input of 256 seconds, the RE-SepFormer is 7x faster than SepFormer-Light. Furthermore, it results in a memory usage reduction of up to 28% for long sequences.

It is worth noting that the RE-SepFormer is composed of self-attention blocks that consist of feed-forward layers. This feature makes the overall architecture highly parallelizable. In contrast, SkiM is mainly composed of RNN (LSTM) layers, which require sequential processing. This potentially explains why it is not faster than RE-SepFormer, despite using only 60% of the MACs.

### 4.4. Comparison with Other Models

In Table 3, we compare the performance of RE-SepFormer with a wide range of models from the literature (WSJ0-2Mix, non-causal setting). Despite its efficiency, we observe that RE-SepFormer achieves competitive performance. For example, RE-SepFormer performs comparably to Dual-Path RNN, while being significantly more efficient in terms of MACs. The RE-SepFormer outperforms the popular Conv-TasNet, and SuDoRM-RF models. We also compare against...
popular efficient Transformer architectures (i.e., Refformer and Longformer) applied without chunking [18]. The RE-SepFormer outperforms aforementioned efficient Transformers as well.

4.5. Ablation Studies

To assess the relative importance of each component to the overall performance, we conduct the following ablation studies: (1) we decrease to 4 the number of layers in the IntraTransformer and Memory Transformer modules; (2) we reduce to 512 the dimension of the positional feed-forward layers in the IntraTransformer and Memory Transformer modules; (3) we combine all the previous ablations.

Table 4 shows the results. Modifications to the IntraTransformers lead to the most significant performance drop together with the most substantial reduction in the number of parameters and MACs. In particular, halving the number of layers has the largest impact. On the contrary, when we perform ablations on the Memory Transformer, we observe minimal effects on both the performance and MACs, despite a moderate reduction in the number of parameters.

It is worth noting that when we combine all the ablations, we still achieve an acceptable SDRi, surpassing methods like TasNet, SignPredictionNet and Conv-TasNet, while utilizing significantly fewer parameters and MACs. This further highlights the suitability of RE-SepFormer for small-footprint devices.

5. CONCLUSIONS

In this paper, we proposed the RE-SepFormer, which is a contribution towards more efficient speech separation with Transformers. The RE-SepFormer uses non-overlapping blocks and relies on compact latent summaries calculated from each chunk rather than attending all the time steps. Our experiments, conducted on the WSJ0-2Mix and WHAM! datasets in both causal and non-causal settings, show that the RE-SepFormer achieves an SDRi of 18.9 dB on WSJ0-2Mix and 14.4 dB on WHAM!. Compared to the SepFormer, it employs more than 3x fewer parameters with a 11x reduction of the MACs. The model is mainly composed of feed-forward layers, and it is thus highly parallelizable. As a result, it scales significantly better than the previous Transformer-based architectures in terms of memory and inference time, making it more suitable for processing long mixtures. This feature makes the RE-SepFormer particularly suitable for real-time low-latency speech separation on small-footprint devices such as GPU-equipped smartphones or laptops.

## 6. REFERENCES

[1] T. Brown et al., “Language models are few-shot learners,” in NeurIPS, 2020, pp. 1877–1901.
[2] A. Chowdhery et al., “PaLM: Scaling language modeling with pathways,” arXiv preprint arXiv:2204.02311, 2022.
[3] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, “Megatron-LM: Training multi-billion parameter language models using model parallelism,” arXiv preprint arXiv:1909.08053, 2020.
[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in NeurIPS, 2020, pp. 12449–12460.
[5] W.-N. Hsu et al., “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” IEEE/ACM Trans. Audio, Speech and Lang. Proc., pp. 3451–3460, 2021.
[6] T. Parcollet and M. Ravanelli, “The energy and carbon footprint of training end-to-end speech recognizers,” in Interspeech, 2021.
[7] N. Tomashenko et al., “Introducing the voiceprivacy initiative,” in Interspeech, 2020.
[8] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in NeurIPS Deep Learning and Representation Learning Workshop, 2015.
[9] R. Reed, “Pruning algorithms-a survey,” IEEE Transactions on Neural Networks, pp. 740–747, 1993.
[10] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, “Pruning convolutional neural networks for resource efficient inference,” in ICLR, 2017.
