Blending the attention mechanism in TasNet

Ismail Alaoui Abdellaoui, Nathan Souviraà-Labastie

BLENDING THE ATTENTION MECHANISM IN TASNET

Ismail Alaoui Abdellaoui[*]

Maastricht University
The Netherlands

Nathan Souviraà-Labastie[‡]

A-Volute
Lille, France

ABSTRACT

Audio source separation has seen a rapid progress due to the advances in data-driven approaches. Motivated by the success of the transformer architecture and the self-attention mechanism in many fields, we investigate in this paper how to include these recent advances in an audio source separation algorithm and in particular in the separator of a TasNet architecture. Our main result leads us to think that LSTM layers are required before the attention mechanism. Our other architecture design experiments focus on finding an optimal number of dense nodes within the point-wise neural networks of the transformer block as well as the right weights initialization in relationship with the multiphase gammatone filterbank. Finally, the non-causal version of our proposed algorithm shows promising results on the MUSDB18 test set (SiSEC challenge data).

Index Terms— Audio source separation, Self-attention mechanism, Transformer, TasNet

1. INTRODUCTION

Data-driven audio source separation greatly benefited from the rapid progress of deep learning techniques. While traditional approaches relied on various priors such as the harmonics of the signal or a linear correlation within the audio signal, recent techniques are data-driven and centered around deep neural networks [1, 2]. Within the field of audio source separation, we can distinguish two families of approaches, namely the frequency and time domain approaches. Early approaches to audio source separation were frequency domain approaches and the first data-driven approaches were also frequency based [1].

More specifically, it consists in using a time-frequency representation, *e.g.* Short-Time Fourier Transform (STFT), of the time domain audio signal as input and predicting the time-frequency representation of the unknown sources before converting back into the time domain. A frequency mask is usually predicted for each source and transcribes the fact that a particular frequency bin corresponds to a given source. Various techniques including ensemble learning as well as deep clustering among others used a time-frequency representation as input, yielding various levels of success [3–8].

On the other hand, time domain approaches quickly became prevalent as they are able to address both the magnitude and the phase of the signal in an end-to-end fashion [9]. These architectures typically incorporate three subparts: an encoder, a separator, and a decoder. Similarly to the time-frequency approaches, time domain approaches can also be mask-based, and it is the separator that learns to predict relevant masks for each source. Conversely to frequency approaches where the representation is fixed, the representation is learned by the encoder, which converts the raw audio data into a weight matrix. Finally, the decoder applies the masks produced by the separator on the weight matrix and reconstructs each time domain signal for each predicted source. One of the foundational time-domain approaches, TasNet, was first applied to address real time speech separation [10], and led to a great state-of-the-art improvement. A noncausal version of TasNet using bidirectional long short-term memory (LSTM) layers in the separator block was also proposed, yielding a better performance than the causal counterpart on the WSJ0-2mix dataset [11]. Other recent approaches include models using convolutional layers, recurrent layers, and U-Net based architectures [12–15].

An important subpart of the time domain approaches is the separator block. Since it is the one with the most learnable parameters, it is also most responsible for the performance of the separation. While the separator of the original TasNet is based on LSTMs, there has been an urge to use the multihead self-attention mechanism instead of the LSTM layers since the introduction of the transformer architecure [16]. This mechanism has proven to be effective for sequence transduction thanks to its parallelizable scaled dot product. Self-attention was successfully applied in sequence transduction for speech recognition [17], language modeling [18], or recommender systems [19]. Concerning the usage of the self-attention mechanism for audio separation, a few pioneer approaches have been proposed [20, 21]. Through depth-wise separable CNNs and multi-head self-attention, SamsNet [20] obtains results at the level of the state of the art on the MUSDB18 dataset, which is quite promising.

In this paper, we first investigate the incorporation of the self-attention mechanism inside the separator block of

[*]Work done during master internship at A-Volute
[‡]Contact: nathan.souviraa-labastie@a-volute.com

TasNet. More specifically, we demonstrate which kind of configuration works well for this particular task, as well as show the impact of the weight initialization with respect to a deterministic encoder. In addition, we propose a noncausal transformer-based separator which equals the original TasNet on MUSDB18 test set with additional benefits.

2. PROPOSED LTL ARCHITECTURE

In this section, we present the proposed LTL architecture (LSTM-Transformer-LSTM) applied to TasNet's separator and formalize the self-attention mechanism incorporated in it.

2.1. Baseline system

The backbone of our model is TasNet [10]. Starting from the input signal, a segment is selected to be processed. This segment then passes through two 1D convolutional layers in parallel with different activation functions. One of the layer uses the rectified linear unit (ReLu) while the other one uses the sigmoid function. An element-wise multiplication (Hadamard product) is then performed between the outputs of each layer. This produces the mixture weight matrix $w \in \mathbb{R}^{L \times C}$, where C is the number of filters used in the 1D convolutional layer and L is the length of the kernel. The decoder block uses a dense layer on the result of the product from the output of the separator and w. This acts as a learned deconvolutional operation, mimicking an inverse operation of the 1D convolutional layers [10]. The last step is to concatenate the recovered signals across all segments to obtain each estimated source s_i and compute the loss function.

2.2. Multi-headed self-attention

Self-attention was introduced by Vaswani *et al.* [16] to capture long-range dependencies in sequences through an attention mechanism. Other key aspects of this mechanism are a reduction of the complexity per layer, as well as an increase in the amount of parallel processing during the training. Given a sequence of inputs $I \in \mathbb{R}^{S \times E}$ where S is the sequence length and E is the embedding dimension of each input feature, we first compute the query Q, key K, and value V matrices are computed as follows:

$$Q = IW_q, \quad K = IW_k, \quad and \quad V = IW_v, \tag{1}$$

where W_q, W_k, and W_v are learnable weights through a linear function. The attention output, also called the head output, is then computed as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{2}$$

where d_k is the dimension of the key vector $K \in \mathbb{R}^{1 \times d_k}$. The operation inside the softmax function is called a scaled dot product, and is implemented to refrain the dot products from having big values [16]. In addition to this attention mechanism, multiple heads using multiple projections W_q, W_k, and W_v were used to allow the network to attend various areas of the sequence. Given a number of heads H, multi-head attention is computed as:

$$MultiHead(Q, K, V) = Concat(OutHead_1, ..., OutHead_H)W^O \tag{3}$$

where $OutHead_i$ is computed through equation 2. The original transformer mentioned is based on an encoder-decoder structure. The encoder part of the transformer contains 2 sub-layers: the multi-head self-attention and the position-wise feed-forward neural network. While the multi-head self-attention part is explained above, the feed-forward neural network uses the output of the multi-head attention as an input. Given the output of the multi-head attention $MultiHeadOut$, we feed this output to a 2 layer fully-connected network with a ReLu activation function after the first layer:

$$FFN(MultiHeadOut) = ReLu(MultiHeadOutW_1 + b_1)W_2 + b_2 \tag{4}$$

Residual connections are also included around each of the 2 sub-layers followed by a layer normalization. In this work, we are only interested in the encoder part of the transformer since we focus into the latent space of TasNet's separator.

2.3. Proposed LTL architecture

As mentioned previously, we investigate the TasNet separator by incorporating both LSTM layers and encoders of the transformer architecture. More specifically, we try zero or more LSTM layers before and/or after the encoder block of the transformer. This transformer block is made of at least one transformer encoder. Fig. 1 shows the LTL model. For the sake of simplicity, we refer in the legend and later a transformer encoder as simply a transformer layer.

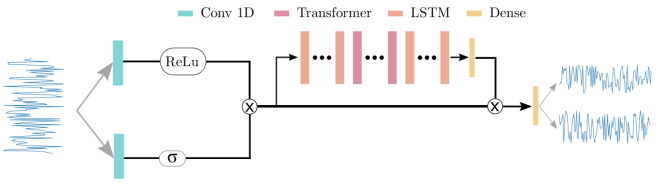

Fig. 1. Schema of the LTL architecture.

3. EXPERIMENTAL PROCEDURES

We present in this section a quick description of the music dataset as well as the modalities of the experiments. More specifically, we define the data used, show some hyperparameters used, and formalize the objective function.

3.1. Dataset and training procedure

Each input data is a mixture track $m_i \in \mathbb{R}^l$ where l is the length in samples of the track, while an output is the set of source tracks $\mathcal{S} = \{s_i\}_{i=1}^{C}$, where C is the number of target source signals. The entire dataset is a collection of songs divided into training and validation files (80%/20%) also described in [22]. Unless otherwise stated, 10% of all the dataset is used. This choice has been made to shorten experimental duration. Each data sample is made of 3 files: the input mixture song, the target vocal ground truth, and the target accompaniment ground truth. We use audio excerpts of 20 seconds as recommended in [22]. The sampling rate is 44.1 kHz.

Similarly to TasNet, we use the scale-invariant source-to-noise ratio (SI-SNR) as training objective because this loss function proved to be effective for this particular use case [12, 15, 22–24].

Adam algorithm [25] was used to optimize the SI-SNR, with a learning rate of $1e^{-4}$. Gradient clipping by norm was also used, with a threshold of 5. In addition, a basic learning rate scheduler was implemented with a halving of the learning rate if the validation loss does not improve after three consecutive epochs. An early stopping mechanism is also used after ten consecutive epochs. This is why the curves plotted in Fig. 2 and 3 are of different length. In order to make a fair comparison between the experiments, some hyperparameters were kept constant across all the trainings. These include the hidden size of the LSTMs (500), the number of filters in the 1D convolution of the encoder block (500), the length of these filters (220), as well as the number of the sources (2).

Concerning the used hardware, most experiments used a Nvidia GTX 1080 GPU, while all experiments involving more than 10% of the dataset used a Nvidia Titan X GPU.

3.2. MP-GTF

Another recent work in [26] proposed a deterministic encoder block instead of a learned one, which is based on the auditory system and which showed to yield an improvement of the scale-invariant source-to-noise ratio (SI-SNR) of 0.7 dB. We decided to incorporate such a deterministic encoder introduced in [26] in our experiment. This approach has been adapted to audio source separation, allowing real-time processing and enforcing the non-negative constraint.

3.3. Weight initialization

The initialization of the weights in the transformer block at the beginning of the training revealed to be a crucial hyperparameter, because sometimes allowing the gradient descent to converge faster. Here we present the 4 distribution functions used to initialize the weights. The first two functions are the normal and uniform distributions K_n and K_u of the Kaiming function defined in [27]. We also use the normal and uniform versions of the Glorot initialization [28] X_n and X_u.

4. RESULTS AND DISCUSSION

In this section, the findings of a set of experiments are depicted. For the sake of simplicity, we write X_trans_Y to denote X LSTM layers before the encoder part of the transformer, and Y LSTM layers after it.

4.1. The right LTL configuration

A first investigation was based on the right number of LSTM layers before and after the transformer block. Several configurations were tested (*i.e.*, LSTMs before only, after only, both before and after, no LSTMs), and our research showed that using LSTM layers only before the transformer block yielded the best results. More specifically, the best separator architectures were 1_trans_0 and 2_trans_0, performing similarly to our TasNet baseline model with less parameters, or outperfoming the baseline by 0.25 dB SI-SNR, respectively.

4.2. Low impact of the fully connected layer

The number of fully connected nodes within the position-wise feed-forward layer of the transformer layer was also subject to analysis. Based on the previous results, both the 1_trans_0 and 2_trans_0 architectures were examined. The number of fully connected nodes tested were 128, 256, 512, 1024, and 2048 for both of these configurations. From a general standing, decreasing the number of fully connected nodes is beneficial to the models because it allows simpler networks without being detrimental in terms of performance. Moreover, 256 dense nodes is the optimal number of dense nodes. Indeed, using 256 nodes allowed the 1_trans_0 model to perform comparably to the baseline model using 11% less parameters, and the 2_trans_0 model to slightly outperform the baseline model. Fig. 2 shows the impact of these experiments with respect to the average SI-SNR of both the loss of the vocals and the accompaniment. An interesting behavior is the suboptimal loss then a sudden drop when using 128 or 2048 dense nodes, while the other numbers of dense nodes make the network easily escape local minima. Therefore, it seems that the network architectures need neither too few nor too many dense nodes.

4.3. Impact of the transformer weight initialization

As mentioned in section 3.3, the initial weight distribution plays a critical role in the trainings performed. Therefore, these experiments used different function distributions to initialize the weights of the transformer. Additionally, we also use in some of the experiments the deterministic encoder presented in section 3.2 (*i.e.* MP-GTF). The results are shown in Fig. 3.

Concerning the trainings involving the 1_trans_0 model, if we analyze the results by pairs of experiments where each pair is made of an experiment using a specific distribution

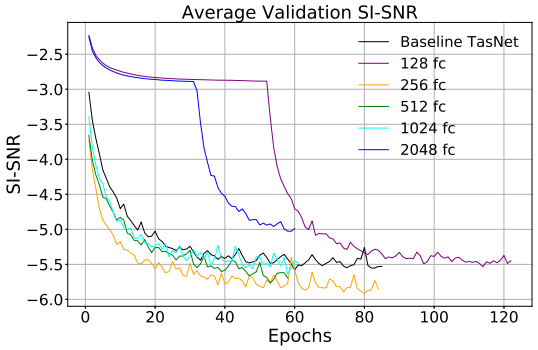

Fig. 2. The impact of the number of fully connected nodes in the transformer part of the *2_trans_0* architecture.

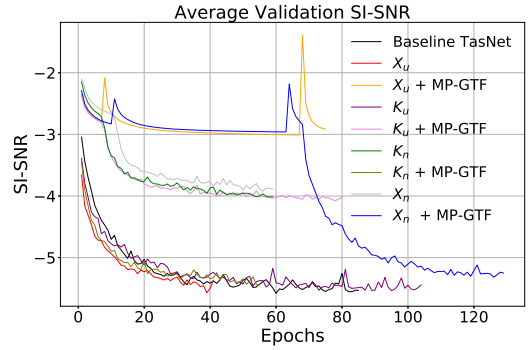

Fig. 3. The impact of MP-GTF and weight initialization on the *1_trans_0* architecture.

and the other experiment is made of the same distribution jointly used with MP-GTF, we can observe an interesting pattern. The experiment "X_u + MP-GTF" gives worse results than using X_u only. Similarly, the experiment "K_u + MP-GTF" gives worse results than using K_u only. In contrast to this observation, using "K_n + MP-GTF" increases the performance by approximately 1.5 dB compared to the experiment using the distribution K_n only. In addition, the experiment "X_n + MP-GTF" leads to an improvement of around 1.25 dB compared to using X_n only, even if the training seemed stuck in a local minimum during the first 70 epochs.

To sum up the results observed for 1_trans_0, we can see there is a good synergy between the normal distribution on the transformer and the MP-GTF on the encoder of TasNet when used on the 1_trans_0 architecture since the filterbank consistently improved the performance of the experiments where the normal distribution is used. It would be hard to come up with an interpretation of such result mainly because of the position of the LSTM layer between the encoder (potentially set to MP-GTF) and the transformer block. Furthermore, the opposite effect can be seen when using a uniform distribution. Similar experiments for the 2_trans_0 architecture leads us to choose the Xavier uniform distribution without using the MP-GTF for later experiments.

4.4. Bidirectionnal version scores on MUSDB18 test set

Motivated by the superior results of the bidirectional version of TasNet, a LTL architecture using bidirectional LSTMs was also investigated. In particular, a 3_trans_0 model using 2048 dense nodes within the transformer yielded comparable separation performances to our TasNet baseline model (here with bidirectional LSTMs) when using 10% or 100% of the dataset, hence showing good scaling capability. Despite the 2048 dense nodes being unfavorable for the previous causal models, using this number of dense nodes for the bidirectional LTL leads to stable and efficient learning of the separation task. We assume that using three bidirectional LSTM layers induces a transformer block with more paramaters in order to learn the appropriate masks. Moreover, a Xavier uniform was used to instantiate the weights of the transformer block, and no MP-GTF was used.

In terms of SDR on the MUSDB18 test set, the proposed 3_trans_0 achieved an encouraging 7.30 SDR (dB) on the `vocals` source and 13.46 SDR (dB) on the `accompaniment` source which is also closed but slightly inferior to the results obtained by our TasNet baseline [22]. However, this proposed 3_trans_0 architecture brings additional benefits compared to our TasNet baseline: firstly, a faster convergence (58% of the time necessary to TasNet to end its training, *i.e., 29 days instead of 50*) with around 3 times less epochs but with a double epoch duration and secondly a decrease by 53 % of the inference time compared to TasNet, but the number of learnable parameters increased (+43%, *i.e.*, $33M$). One can argue that the faster convergence is due to a higher number of parameters but similar behaviors are observed during experiments of Section 4.2 without the bias of different number of parameters.

5. CONCLUSION

This paper explores the use of the self-attention mechanism for the task of music source separation within the state of the art framework of TasNet. In particular, we described a series of experiments that lead us to our proposed new architecture scheme for the separator module of the TasNet framework/architecture: 1 to 3 LSTM layers followed by layers composed of the encoder part of a transformer. Experiments on the combination of a non-learned TasNet encoder (using Gammatone) and different transformer weight initialization are also described with intriguing behaviours. Finally, a bidirectional version of the proposed model is evaluated on the MUSDB18 test set (SiSEC challenge data) and reaches comparable separation performances as the state of the art while significantly reducing training, convergence and in-use time. Future experiments will focus on augmenting the length of the audio excerpts to conclude on the capacity of our architecture to better capture long dependencies.

6. REFERENCES

[1] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, Derry FitzGerald, and Bryan Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 8, pp. 1307–1335, 2018.

[2] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito, "The 2018 signal separation evaluation campaign," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2018, pp. 293–305.

[3] Xiao-Lei Zhang and DeLiang Wang, "A deep ensemble learning method for monaural speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 24, no. 5, pp. 967–977, 2016.

[4] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey, "Single-channel multi-speaker separation using deep clustering," *arXiv preprint arXiv:1607.02173*, 2016.

[5] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[6] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 246–250.

[7] Yi Luo, Zhuo Chen, and Nima Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

[8] A. Jansson, Eric J. Humphrey, N. Montecchio, Rachel M. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *ISMIR*, 2017.

[9] Szu-Wei Fu, Tao-Wei Wang, Yu Tsao, Xugang Lu, and Hisashi Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1570–1584, 2018.

[10] Yi Luo and Nima Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 696–700.

[11] John Garofalo, David Graff, Doug Paul, and David Pallett, "Csr-i (wsj0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[12] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[13] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *arXiv preprint arXiv:1806.03185*, 2018.

[14] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[15] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 46–50.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[17] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[20] Tingle Li, Jiawei Chen, Haowen Hou, and Ming Li, "Sams-net: A sliced attention-based neural network for music source separation," 2019.

[21] Tingle Li, Qingjian Lin, Yuanyuan Bao, and Ming Li, "Atss-net: Target speaker separation via attention-based neural network," *arXiv preprint arXiv:2005.09200*, 2020.

[22] Emery Pierson Lancaster and Nathan Souviraa-Labastie, "A frugal approach to music source separation," 2020.

[23] Eliya Nachmani, Yossi Adi, and Lior Wolf, "Voice separation with an unknown number of multiple speakers," *arXiv preprint arXiv:2003.01531*, 2020.

[24] Manuel Pariente, Samuele Cornell, Joris Cosentino, Sunit Sivasankaran, Efthymios Tzinis, Jens Heitkaemper, Michel Olvera, Fabian-Robert Stöter, Mathieu Hu, Juan M. Martín-Doñas, David Ditter, Ariel Frank, Antoine Deleforge, and Emmanuel Vincent, "Asteroid: the PyTorch-based audio source separation toolkit for researchers," *arXiv preprint arXiv:2005.04132*, 2020.

[25] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] David Ditter and Timo Gerkmann, "A multi-phase gammatone filterbank for speech separation via tasnet," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 36–40.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[28] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.