Multi-Scale Group Transformer for Long Sequence Modeling in Speech Separation

Yucheng Zhao[*1], **Chong Luo**[2], **Zheng-Jun Zha**[†1] and **Wenjun Zeng**[2]
[1]University of Science and Technology of China
[2]Microsoft Research Asia
lnc@mail.ustc.edu.cn, cluo@microsoft.com, zhazj@ustc.edu.cn, wezeng@microsoft.com

Abstract

In this paper, we introduce Transformer to the time-domain methods for single-channel speech separation. Transformer has the potential to boost speech separation performance because of its strong sequence modeling capability. However, its computational complexity, which grows quadratically with the sequence length, has made it largely inapplicable to speech applications. To tackle this issue, we propose a novel variation of Transformer, named multi-scale group Transformer (MSGT). The key ideas are group self-attention, which significantly reduces the complexity, and multi-scale fusion, which retains Transform's ability to capture long-term dependency. We implement two versions of MSGT with different complexities, and apply them to a well-known time-domain speech separation method called Conv-TasNet. By simply replacing the original temporal convolutional network (TCN) with MSGT, our approach called MSGT-TasNet achieves a large gain over Conv-TasNet on both WSJ0-2mix and WHAM! benchmarks. Without bells and whistles, the performance of MSGT-TasNet is already on par with the SOTA methods.

1 Introduction

Speech separation is a fundamental task in acoustic signal processing with a wide range of applications [Wang and Chen, 2018]. The goal of speech separation is to separate target speech from interfering speech, non-speech noise, or both. Thanks to the success of deep learning, the performance of single-channel speech separation system have been dramatically improved in recent years [Hershey *et al.*, 2016; Kolbæk *et al.*, 2017]. In particular, a new category of speech separation methods called time-domain methods [Luo and Mesgarani, 2018; Luo and Mesgarani, 2019] begin to emerge. These methods take the sampled data points from raw waveform as input, use a learnable neural network layer as encoder, and adopt a sequence modeling tool for feature separation before the separated feature is converted back to time domain by a learnable decoder. The time-domain methods have been shown to surpass ideal time-frequency (T-F) magnitude masking methods for speech separation.

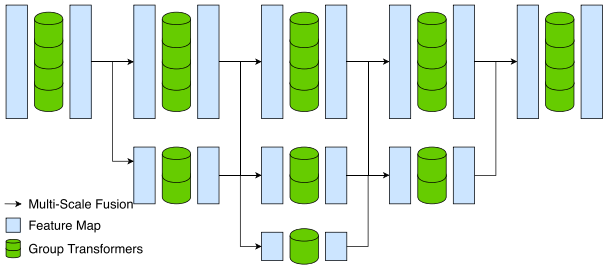


Figure 1: A schematic diagram of the proposed MSGT with three scales. The cylinders represent group Transformers.

Recently, several investigations [Heitkaemper *et al.*, 2019] reveal that the performance of time-domain methods is highly dependent on using short frame length in the encoder, which produces extremely long sequence. For example, the frame length used in Conv-TasNet [Luo and Mesgarani, 2019] is 2ms, producing 4000 frames for a 4-second audio segment with 1ms overlap. Such a long sequence is difficult to model using the conventional recurrent neural network (RNN) or temporal convolutional network (TCN)[Bai *et al.*, 2018] as both of them have a long path before connecting all positions [Vaswani *et al.*, 2017].

Transformer [Vaswani *et al.*, 2017] have recently shown its strong capability for sequence modeling in many natural language processing tasks [Devlin *et al.*, 2018]. Compared to RNN or TCN, Transformer uses self-attention (SA) to compute correlations between any input positions in parallel, which can effectively capture global dependencies in addition to the local dependencies. Introducing Transformer to speech separation has the potential to model the long-range dependencies displayed in speech signals. But the main obstacle is the complexity. The computational complexity of Transformer grows quadratically with the sequence length, which is not affordable when the sequence length is on the order of thousands.

In order to solve this issue, we propose a novel architecture called multi-scale group Transformer (MSGT). We divide the input sequence into groups and use group self-attention to calculate correlations within each group. The complexity re-

[*]This work is done when Yucheng Zhao is an intern in MSRA
[†]Corresponding author

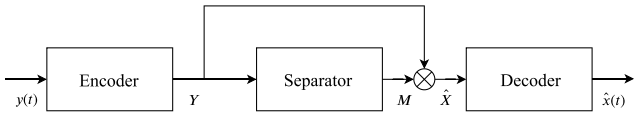


Figure 2: A schematic diagram of a deep-learning-based speech separation system.

mains tractable as long as the group size is not too large. As we down-sample the features, the group size in group Transformer is kept constant, so correlations in a longer range can be captured. On the lowest scale, all data points are contained in a single group. As such, both local and global dependencies are captured and retained. Besides, this structure pays more attention to local dependencies, which aligns with the physical characteristics of acoustic signals.

We implement two versions of MSGT with different complexities and apply them to the Conv-TasNet speech separation system. In particular, we replace TCN with MSGT for sequence modeling. Experimental results show that the MSGT-based system, which we call MSGT-TasNet, achieves significant performance gain over Conv-TasNet in both noise-free speech separation on WSJ0-2mix dataset [Hershey *et al.*, 2016] and noisy speech separation on WHAM! dataset [Wichern *et al.*, 2019].

To summarize, our work makes three main contributions: (1) To the best of our knowledge, we are the first to investigate Transformer for sequence modeling in speech separation. (2) We propose multi-scale group Transformer which reduces the complexity of the standard Transformer without losing its capability to model global dependencies. (3) We implement two versions of MSGT and use them to build a speech separation system MSGT-TasNet. Without bells and whistles, the performance of MSGT-TasNet is already on par with the state-of-the-art (SOTA) methods.

2 Related Work

In this section, we provide the background of the speech separation task and briefly review several variations of Transformer for long sequence modeling.

2.1 Speech Separation

The goal of speech separation is to separate target speech from interference including interfering speech, non-speech noise, or both. As shown in Fig.2, a deep learning-based speech separation system is composed of three modules. Before the pioneering work named TasNet [Luo and Mesgarani, 2018], researchers have been using fixed transformation, such as STFT and ISTFT, for the encoder and the decoder [Hershey *et al.*, 2016; Kolbæk *et al.*, 2017; Shi *et al.*, 2018]. The separation is conducted on the two-dimensional time-frequency features. However, the success of TasNet has ignited the interest of time-domain approaches, which directly take data samples from time-domain raw waveform as input, and conduct separation in a latent one-dimensional domain.

One typical feature of time-domain approaches is that their success rely on the effectiveness in modeling long sequences [Heitkaemper *et al.*, 2019; Luo and Mesgarani, 2019]. For Conv-TasNet, it is discovered that shorter frame length achieves better performance than longer ones. Therefore, a frame length of 2ms is suggested for high performance. This length is an order of magnitude smaller than the length used by conventional encoders. Given a fixed-length audio input, such a short frame length also produces an extremely long sequence, which is difficult to model.

In Conv-TasNet [Luo and Mesgarani, 2019], the authors used a temporal convolution network to model long-range dependencies. However, it has been shown [Vaswani *et al.*, 2017] that convolution is not as efficient as Transformer in sequence modeling, as the latter is capable of capturing both local and global dependencies. This has been the motivation of our work. More recently, FurcaNeXt [Zhang *et al.*, 2020] introduced gated activation and ensemble learning into the framework to improve performance. However, they are still using TCN for sequence modeling.

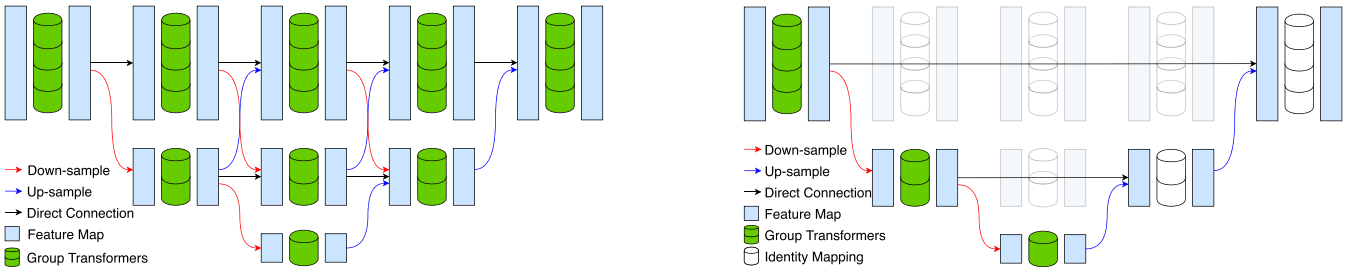
2.2 Transformer

Transformer is a strong sequence modeling tool but it is not readily applicable to long sequences due to its quadratically growing computational complexity with the sequence length. Several pieces of work [Dai *et al.*, 2019; Liu and Lapata, 2019; Shen *et al.*, 2018; Miculicich *et al.*, 2018] in natural language processing area have tried to tackle this limitation, so that Transformer can be applied to document-level tasks or having a higher efficiency. The main idea is to divide the sequence into a few conceptually meaningful sets or blocks, such as sentences and paragraphs, and then adopt a hierarchical architecture to explore the intra-set and inter-set correlations. To be more specific, local dependencies are captured within each set, and global dependencies are computed through cross-set attention.

[Shen *et al.*, 2018] presents a bi-directional block self-attention network (SAN) that divides a sequence into blocks and sequentially applies intra-block SAN to each block and inter-block SAN across blocks. [Miculicich *et al.*, 2018] uses a hierarchical attention network structure for document-level machine translation and [Liu and Lapata, 2019] uses a hierarchical Transformer for multi-document summarization. Transformer-XL [Dai *et al.*, 2019] introduces a segment-level recurrence mechanism that enables Transformer to learn dependencies beyond a fixed length.

The main difference between these hierarchical SAN and our work is that we do not calculate set-level dependencies. NLP tasks involve semantic units such as sentence, paragraph, and document. However, speech separation is a low-level task without multiple semantic units. For such input data, our model uses multi-scale fusion architecture to calculate element-level dependencies at different resolutions.

There is also a work called Sparse Transformer [Child *et al.*, 2019] for long sequence generation. They factorized full attention matrix by some sparse attention matrices to reduce the complexity. Compared to our model, Sparse Transformer relies on a highly optimized sparse matrix implementation and the complexity of it is $O(n\sqrt{n})$, which is higher than ours for long sequences.



(a) Dense Fusion (b) Light Fusion

Figure 3: Two implementations of MSGT. Dense-fusion MSGT employs multiple group Transformers (GT) on each scale while light-fusion MSGT only uses one GT on each scale.

3 Multi-Scale Group Transformer

We intend to design a long-sequence modeling tool based on Transformer. There are two design objectives. First, we shall reduce the computational complexity of Transformer so that it can scale with sequence length. Second, we shall maintain Transformer's ability to model both short-term and long-term dependencies.

3.1 The Proposed Architecture

Fig. 1 presents the architecture of the proposed multi-scale group Transformer (MSGT). The key innovations are the group self-attention and the multi-scale fusion.

In the original Transformer design, correlations are computed between any two positions in the input sequence. This brings quadratic complexity with respect to the sequence length. In contrast, the propose group self-attention restricts the correlation computation within local regions. When the group size is fixed at a constant, the number of groups grows linearly with the sequence length, so does the computational complexity of group self-attention.

However, group self-attention (GSA) does not consider correlations across groups, losing the capability to capture global dependencies. To retain this important capability and to balance the computation resources allocated between local and global dependencies, we propose the multi-scale architecture. On the high-resolution (large) scale, GSA captures local dependencies, while on the low-resolution (small) scale, GSA captures long-range dependencies. As sequences in small scales are several times shorter than sequences in large scales, more computation resources are thus allocated to capture local dependencies. This is consistent with the intuition that local correlations are stronger and more important than global correlations in audio signals.

We implement MSGT using two basic modules: operation module and transition module. The *operation module* conducts multi-scale feature transformation. It may conduct different transformations, including group Transformer and identity mapping, on different scales. The *transition module* handles feature resizing and feature fusion.

Group Self-Attention

Given an input sequence $X \in \mathbf{R}^{n\times d}$ where d is the feature dimension and n is the sequence length, GSA first divides the sequence into s equal-sized non-overlapped groups with group size g. Then, standard self-attention is calculated within each group and the output of each group is concatenated. We formulate the above process as follows:

$$\text{Group}(X) = \{X_1, X_2, ..., X_s\}, X_i \in \mathbf{R}^{g\times d} \quad (1)$$

$$\text{GSA}(X) = \{\text{SA}(X_1), \text{SA}(X_2), ..., \text{SA}(X_s)\} \quad (2)$$

The standard self-attention, which proposed in [Vaswani *et al.*, 2017] is computed as follows:

$$Q = XW^Q, K = XW^K, V = XW^V \quad (3)$$

$$\text{SA}(X) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V \quad (4)$$

where $\sqrt{d}$ is the scaling factor, $W^Q, W^K, W^V \in \mathbf{R}^{d\times d}$ are parameter matrices of different linear transformations.

For simplicity, here we only describe the formula of single-head group self-attention. The extension to multi-head group self-attention is straightforward.

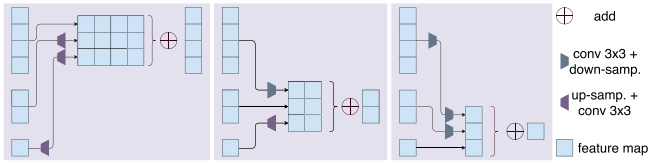


Figure 4: Illustration of feature fusion by transition modules. The large, medium, and small scales are shown from left to right.

Multi-Scale Fusion

Multi-scale fusion is conducted by the transition module. It exchanges information across multi-scale features. Let us take 3 scales as a example, which is illustrated in Figure 4. For the largest scale, we first conduct up-sampling on two smaller scales. Note we gradually up-sample the smallest scale by using two sequential up-sampling networks. Then, we have three features with the same size and use addition to fuse them. For the medium scale, we up-sample the smallest scale, down-sample the largest scale and then add them together. The operation on the smallest scale is similar. Note that we do not have to fuse all scales in the transition module. It is a design choice how many scales are fused. We implement two versions of MSGT with different complexities.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
MSG Self-Attention (Light)	$O(g \cdot n \cdot d)$	$O(1)$	$O(\log_2(n/g))$
MSG Self-Attention (Dense)	$O(g \cdot n \cdot d \cdot \log(n/g))$	$O(1)$	$O(\log_2(n/g))$

Table 1: Per-layer complexity, minimum number of sequential operations and maximum path lengths for different layer types. The definition of n, d and g follows Section 3.1 and k is the convolution kernel size. The complexity per layer for MSG self-attention is the total complexity divided by the number of layers as different scales have different complexities. Part of this table is adopted from [Vaswani *et al.*, 2017].

3.2 Two Implementations of MSGT

We present two concrete design instances of MSGT, namely dense-fusion MSGT and light-fusion MSGT, in Fig. 3. The fusion structures of these two implementations are inspired by the HRNet [Sun *et al.*, 2019] and the U-Net [Ronneberger *et al.*, 2015] in the computer vision area.

Dense-Fusion MSGT

The architecture of dense-fusion MSGT is illustrated in Fig. 3a. With dense fusion, multiple group Transformers are applied on each scale. For simplicity, we fuse only the adjacent scale features in the transition module. In other words, an input to the group Transformer is fused from no more than three scales. The dense fusion has log-linear complexity with sequence length.

Light-Fusion MSGT

As illustrated in Fig. 3b, the light fusion version only use one GT on each scale, although each GT may contain multiple layers of transformation. The transition module does not do fusion as the features scale down. When the features scale up, only features from two adjacent scales are fused together. The light fusion has linear complexity with sequence length.

3.3 Analysis

The sequence modeling problem can be formulated as mapping one sequence $\{x_1, x_2, ..., x_n\}$ to another sequence of equal length $\{z_1, z_2, ..., z_n\}$, where $x_i, z_i \in \mathbf{R}^d$ denote the feature vector of a symbol in the sequence.

There are three desired properties for a sequence modeling tool: 1) Parallel computing; 2) Capability to capture long-range dependencies; 3) Low-order complexity which allows it to scale to long sequences. The original Transformer paper [Vaswani *et al.*, 2017] introduced three metrics to measure these desired properties. They are complexity per layer, minimum number of sequential operation required and the maximum path length between any two input and output position.

Table 1 provides a comparison of the tree metrics between the proposed MSGT and other widely used sequence modeling tools. A standard self-attention layer has complexity of $O(n^2 \cdot d)$, which is too high for long sequences. The multi-scale group self-attention have maximum path length $O(\log_2(n/g))$ and complexity $O(g \cdot n \cdot d)$ or $O(g \cdot n \cdot d \cdot \log(n/g))$ depending on the fusion choices. We will empirically show this improvement in Section 5.4. For completeness, we also include recurrent and convolutional layers in this table, but they are inferior to self-attention which has been detailed in [Vaswani *et al.*, 2017].

4 MSGT for Speech Separation

The goal of speech separation is to separate target speech from a mixture signal. Following the convention, all the speech signals appeared in the mixture are treated as target speeches. The mixture can be denoted by a sequence $y(t)$ in the time domain. It can be decomposed as the summation of K speech signals $x_k(t)$, and an additive noise $n(t)$,

$$y(t) = \sum_{k=1}^{K} x_k(t) + n(t), \tag{5}$$

where t is the time index.

Recent deep learning-based speech separation systems consist of three main components, namely the encoder, the decoder, and the separator. Fig.2 illustrates this framework. The encoder processes time domain mixture signal $y(t)$ by first dividing it into n overlapping frames $y_j \in \mathbf{R}^{1\times L}$ ($j = 1...n$), where L denotes frame length, and j is the frame index. Then, each frame is transformed into d-dimensional features $Y_j \in \mathbf{R}^{1\times d}$. Features of all the n frames are concatenated to form $Y \in \mathbf{R}^{n\times d}$. The separator use some sequence modeling tool to estimate multiplicative masks $M_k \in \mathbf{R}^{n\times d}$ for each signal and then multiply it on encoded features Y, producing $\hat{X}_k$ as separated features. Finally, the decoder transform $\hat{X}_k$ back to time domain and output the separated signals $\hat{x}_k(t)$.

It is worth noting that it is not necessary for the separated signals to have the same permutation as the ground-truth label. Utterance-level permutation invariant training (uPIT) [Kolbæk *et al.*, 2017] can be applied to address this problem.

We adopt the well-known speech separation framework Conv-TasNet [Luo and Mesgarani, 2019] and replace the original TCN with the proposed MSGT in its separator. The encoder and the decoder are kept the same. Our system is still a time-domain method, so we name it MSGT-TasNet. We train MSTG-TasNet using the time domain scale-invariant signal-to-distortion ratio (SI-SDR) as the training target, which is defined as:

$$\text{SI-SDR} = 10\log_{10}\left(\frac{|\alpha s|^2}{|\hat{s} - \alpha s|^2}\right). \tag{6}$$

where $\hat{s} \in \mathbf{R}^T$ and $s \in \mathbf{R}^T$ are the estimated and original clean signals, respectively. $\alpha = \hat{s}^T s/|s|^2$ is the scaling factor.

Separation Method	SI-SDRi (dB)	SDRi (dB)	PESQ
PSM	16.4	16.7	3.98
DPCL++ (2016)	10.8	-	-
uPIT (2017)	-	10.0	-
chimera++ (2018)	12.6	13.1	-
Sign Prediction Net (2019)	15.3	15.6	3.36
Deep CASA (2019)	17.7	18.0	3.51
Conv-TasNet (2019)	15.3	15.6	3.24
FurcaNeXt (2020)	-	18.4	-
MSGT-TasNet (Light)	16.8	17.1	3.35
MSGT-TasNet (Dense)	17.0	17.3	3.30

Table 2: Performance comparison on WSJ0-2mix. Methods are grouped into ideal masks, T-F domain and time-domain methods.

5 Experiments

5.1 Dataset and Evaluation Metric

We use two datasets for evaluation. The first is the widely-used WSJ0-2mix dataset [Hershey *et al.*, 2016]. WSJ0-2mix contains 30 hours of training data, 10 hours of validation data and 5 hours of testing data. The validation set contains utterances of the same speakers as in the training set, while the testing set contains utterances of different speakers. Each mixture is artificially generated at a random signal-to-noise ratio (SNR) between -5 and 5 dB. The second is the recently proposed WHAM! dataset [Wichern *et al.*, 2019], which is an extension of the original WSJ0-2mix. The WHAM! dataset consists of two speaker mixtures from the WSJ0-2mix dataset combined with real ambient noise samples. This is a more challenging dataset compared to the noise-free WSJ0-2mix.

We use the scale-invariant signal-to-distortion ratio (SI-SDR) improvement [Le Roux *et al.*, 2019] and signal-to-distortion ratio (SDR) [Vincent *et al.*, 2006] improvement as the main evaluation metrics. SI-SDR is also referred to as SI-SNR in some work [Luo and Mesgarani, 2019]. We also report perceptual evaluation of subjective quality (PESQ) [Rix *et al.*, 2001] to evaluate the quality of the separated mixtures.

5.2 Experiment Configuration

We train all models for 1M steps on 4-second segments with sample rate of 8K Hz. We use Adam optimizer with warm-up. The learning rate is initialized to 0.0003 and is adjusted according to the following formula:

$$\text{lr} = \text{init\_lr} \cdot \min(\text{step}^{-0.3}, \text{step} \cdot \text{warmup\_steps}^{-1.3}) \quad (7)$$

We choose warmup_steps $= 10000$. We also use dropout to relieve over-fitting.

In all experiments, we use frame length of 2 ms. We choose group size of 1000 for noise-free speech separation and group size of 500 for noisy speech separation. Following the notations in [Vaswani *et al.*, 2017], the Transformer parameters are $d_{\text{ff}} = 1024$, $d_{\text{model}} = 512$, and $h = 8$. In the light fusion, we use 8 layers of transformation for the GT in the smallest scale and 2 layers for the GT in the other scales. In the dense fusion, we use 3 layers of transformation for GT in all the scales. The output feature dimension of the encoder is 1024.

Separation Method	SI-SDRi (dB)
IRM	12.8
IBM	13.4
PSM	16.8
chimera++	9.9
Conv-TasNet*	12.0
MSGT-TasNet (Light)	12.3
MSGT-TasNet (Dense)	13.1

Table 3: Performance comparison on WHAM!. The three results on top are performance of different ideal masks. Conv-TasNet* is our re-implementation.

5.3 Comparison with Other Methods

Previously, research on speech separation is more focused on separating target speech from interfering speech [Luo and Mesgarani, 2019; Liu and Wang, 2019]. In this paper, we use the term *noise-free speech separation* to denote this conventional setting while we use another term *noisy speech separation* to refer to the task of separating target speech from both interfering speech and non-speech noise. The WSJ0-2mix dataset is used for the former task while the WHAM! dataset is used for the latter task.

Speech Separation Results on WSJ0-2mix

Table 2 shows speech separation results on WSJ0-2mix. DPCL++ [Isik *et al.*, 2016], uPIT [Kolbæk *et al.*, 2017], chimera++ [Wang *et al.*, 2018] and Sing Prediction Net [Wang *et al.*, 2019] are time-frequency (T-F) domain methods. Conv-TasNet [Luo and Mesgarani, 2019], FurcaNeXt [Zhang *et al.*, 2020] and our proposed MSG-TasNet are time-domain methods. The fist row gives the performance of ideal phase-sensitive mask (PSM), which is a reasonable upper bound for all T-F domain methods.

Our proposed two versions of MSGT-TasNet outperform Conv-TasNet by a large margin only by replacing TCN with MSGT, showing that MSGT is a better sequence modeling method for speech separation. The SI-SDRi performance of the light version is 0.2dB lower than that of the dense version, but it uses 2x fewer memory and runs 2.8x faster.

FurcaNeXt and Deep CASA achieve slightly higher SDRi or SI-SDRi than MSGT-TasNet. But the innovations in these two methods can also be applied to our framework and can potentially improve the performance of MSGT-TasNet. Specifically, FurcaNext adopts gated activation and Deep CASA optimizes frame-level separation and speaker tracking in turn. We plan to integrate these two innovations into our model in the future.

Noisy Speech Separation Results on WHAM!

The WHAM! dataset is more realistic and challenging since it contains noise. As this is a new dataset and only baseline result is available, we add several idea masks to the comparison. The ideal masks provide performance upper bounds for most T-F domain methods. Results are presented in Table 3.

In noisy speech separation, MSGT-TasNet (Light) and MSGT-TasNet (Dense) get 0.3 dB and 1.1 dB SI-SDRi gain over Conv-TasNet, respectively. This gain is smaller than

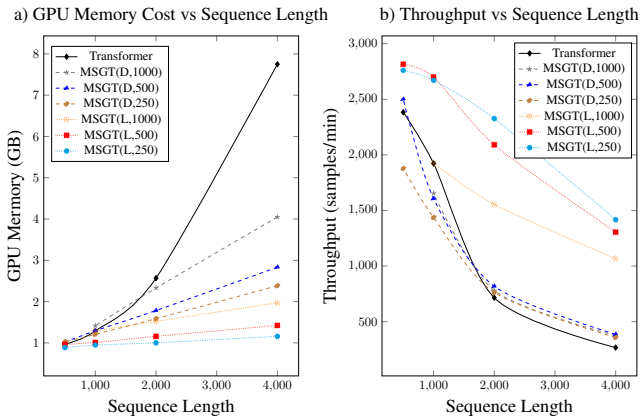


Figure 5: Empirical computation cost of Transformer and MSGT. D and L stand for dense and light, respectively. The number follows indicates group size. One sample is a 4s audio segment.

what we get in noise-free speech separation, but the advantage of the dense version to the light version becomes significant. Both results are due to the fact that noisy speech separation is a more difficult task than its noise-free counterpart. Note that, in this difficult task, MSGT-TasNet (Dense) is the first method that surpasses the ideal T-F mask IRM.

5.4 Efficiency for Long Sequence

MSGT reduces the time/space complexity from $O(n^2 \cdot d)$ to $O(g \cdot n \cdot d)$ or $O(g \cdot n \cdot d \cdot \log(n/g))$ in theory. We would like to compare the empirical GPU memory cost and throughput for speech separation under the same configuration. We use 4s audio segment and choose different frame lengths to control the actual sequence length used in group Transformer. We choose group size g as a hyper-parameter. For different sequence lengths, we keep the total number of group Transformers the same for light fusion and we keep the number of group Transformer in the largest scale the same for dense fusion. This constraint yields models with similar size for different sequence length.

As Fig.5a shows, the GPU memory usage in MSGT is significantly smaller than in Transformer, and it grows slower with the increase of sequence length. Under the same group size, the light version requires less GPU memory than the dense version. Within the same fusion model, the GPU memory cost grows with the group sizes g.

Fig.5b presents the throughput of different models. The numbers are acquired on a single P100 GPU. MSGT (Dense) has slightly lower throughput than Transformer for short sequence because more scales are involved in computation. MSGT (Light) consistently achieves higher throughput than Transformer.

5.5 Ablation Study

All ablation studies are carried out on the WSJ0-2mix speech separation dataset.

Comparison of different sequence modeling tools. As shown in Table 4, using group Transformer without multi-scale fusion significantly degrades the performance. The performance of single-scale GT, which has only one scale and uses the same number of layers as MSGT (Light), is even worse than TCN. Using light fusion, we can achieve almost doubled throughput and 3.3dB SI-SDRi gain over the single-scale version. We would also like to mention that MSGT (Light) achieve 1.5 dB gain over TCN with 3.9X speedup.

Type	SI-SDRi (dB)	Throughput (samples/min)
TCN	15.3	276
Single-scale GT	13.5	528
MSGT (Light)	16.8	1066
MSGT (Dense)	17.0	375

Table 4: Comparison of different sequence modeling tools when group size is 1000.

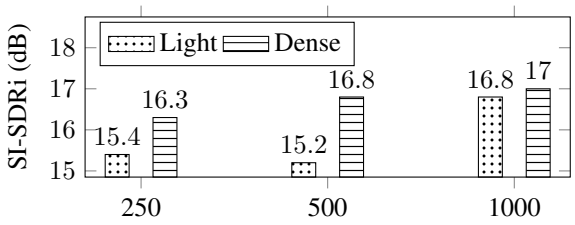


Figure 6: Influence of group size for MSGT-TasNet (Light) and MSGT-TasNet (Dense).

Influence of group size. The hyper-parameter g in group Transformer controls how many frames should be consider as a group. As shown in Fig.6, the largest group size achieves the highest SI-SDRi for both fusion versions. In particular, large group size is essential for the light version to get good performance, which may due to its inadequacy in exploring long-range dependencies.

6 Conclusion and Future Work

In this paper, we present the design and implementation of multi-scale group Transformer for long sequence modeling. Through group self-attention and multi-scale fusion, MSGT significantly reduces the computational complexity of Transformer without affecting its performance. Two versions of MSGT with different complexities are implemented and applied in a well-known speech separation framework called Conv-TasNet. Experiment results show the proposed MSGT-TasNet achieves a large gain over Conv-TasNet on both WSJ0-2min and WHAM! benchmarks. For the noisy speech separation task, MSGT-TasNet is the first approach surpassing the performance of the ideal T-F mask.

In the future, we plan to apply neural architecture search (NAS) [Liu *et al.*, 2018] to find better fusion choices of MSGT. Besides, we plan to integrate the innovations of Deep CASA into MSGT-TasNet. We believe that such combination can further advance the state-of-the-art of speech separation.

References

[Bai *et al.*, 2018] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[Child *et al.*, 2019] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[Dai *et al.*, 2019] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[Devlin *et al.*, 2018] 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.

[Heitkaemper *et al.*, 2019] Jens Heitkaemper, Darius Jakobeit, Christoph Boeddeker, Lukas Drude, and Reinhold Haeb-Umbach. Demystifying tasnet: A dissecting approach. *arXiv preprint arXiv:1911.08895*, 2019.

[Hershey *et al.*, 2016] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *ICASSP*, pages 31–35, 2016.

[Isik *et al.*, 2016] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey. Single-channel multi-speaker separation using deep clustering. *Interspeech*, pages 545–549, 2016.

[Kolbæk *et al.*, 2017] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, Jesper Jensen, Morten Kolbaek, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM TASLP*, 25(10):1901–1913, 2017.

[Le Roux *et al.*, 2019] Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *ICASSP*, pages 626–630, 2019.

[Liu and Lapata, 2019] Yang Liu and Mirella Lapata. Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*, 2019.

[Liu and Wang, 2019] Yuzhou Liu and DeLiang Wang. Divide and conquer: A deep casa approach to talker-independent monaural speaker separation. *arXiv preprint arXiv:1904.11148*, 2019.

[Liu *et al.*, 2018] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[Luo and Mesgarani, 2018] Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *ICASSP*, pages 696–700, 2018.

[Luo and Mesgarani, 2019] Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM TASLP*, 27(8):1256–1266, 2019.

[Miculicich *et al.*, 2018] Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. Document-level neural machine translation with hierarchical attention networks. In *EMNLP*, pages 2947–2954, 2018.

[Rix *et al.*, 2001] Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *ICASSP*, pages 749–752, 2001.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[Shen *et al.*, 2018] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Bi-directional block self-attention for fast and memory-efficient sequence modeling. In *ICLR*, 2018.

[Shi *et al.*, 2018] Jing Shi, Jiaming Xu, Guangcan Liu, and Bo Xu. Listen, think and listen again: Capturing top-down auditory attention for speaker-independent speech separation. In *IJCAI*, pages 4353–4360, 2018.

[Sun *et al.*, 2019] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Vincent *et al.*, 2006] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *TASLP*, 14(4):1462–1469, 2006.

[Wang and Chen, 2018] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM TASLP*, 26(10):1702–1726, 2018.

[Wang *et al.*, 2018] Zhong-Qiu Wang, Jonathan Le Roux, DeLiang Wang, and John R Hershey. End-to-end speech separation with unfolded iterative phase reconstruction. *Interspeech*, 2018.

[Wang *et al.*, 2019] Zhong-Qiu Wang, Ke Tan, and DeLiang Wang. Deep learning based phase reconstruction for speaker separation: A trigonometric perspective. In *ICASSP*, pages 71–75, 2019.

[Wichern *et al.*, 2019] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. Wham!: Extending speech separation to noisy environments. *Interspeech*, pages 1368–1372, 2019.

[Zhang *et al.*, 2020] Liwen Zhang, Ziqiang Shi, Jiqing Han, Anyan Shi, and Ding Ma. Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks. In *MMM*, pages 653–665, 2020.