GLASS: Investigating Global and Local context Awareness in Speech Separation

Kuan-Hsun Ho *, En-Lun Yu †, Jeih-weih Hung ‡, Shih-Chieh Huang §, and Berlin Chen ¶
*†¶ Department of Computer Science and Information Engineering, National Taiwan Normal University, Taiwan
E-mail: {jasonho610, enlunyu, berlin}@ntnu.edu.tw
‡ Department of Electrical Engineering, National Chi Nan University, Taiwan
E-mail: jwhung@ncnu.edu.tw
§ Realtek Semiconductor Corp., Taiwan

***Abstract*—Previous speech separation systems commonly employ the Dual-Path (DP) mechanism. The DP mechanism addresses optimization challenges posed by considerable sequential input lengths, yet its compulsory interleaving pattern for local and global feature extraction raises concerns regarding optimal utilization of features across different layers. This study emphasizes the need for parallel processing of global and local information in speech separation, proposing the Global and Local context-Aware Speech Separation method (GLASS). GLASS integrates self-attention and convolutional layers into a parallel design, demonstrating state-of-the-art performance in both anechoic and noisy settings. The findings reveal patterns in the relevance of local and global information across layers, underscoring the significance of proper architecture in improving speech separation systems.**

***Index Terms*—Speech Separation, Global and Local Dependencies**

I. Introduction

In real-world environments, audio often contains parts where multiple speakers talk over each other. Therefore, accurately separating multiple speakers from a single-channel mixture would significantly facilitate many applications. Due to the outstanding modeling capability, deep-learning algorithms have been used as core models in state-of-the-art speech separation systems [1]–[9]. Moreover, most successful systems follow the encoder-decoder masking-based strategy [2], [3] and the Dual-Path (DP) mechanism [4].

The DP mechanism, proven effective in various speech-processing tasks such as separation [4], [7], [8] and enhancement [10], encompasses three stages: segmentation, block processing, and overlap-add. The segmentation stage splits a sequential input into overlapped chunks and concatenates all the chunks into a 3-D block. Afterward, the block is passed to local and global extractors in an interleaving fashion. Finally, the output from the last layer is transformed back to a sequential output with the overlap-add method. One of the objectives of the DP mechanism is to decrease the optimization difficulty that arises when the sequential input length is considerable.

However, the interleaving pattern of local and global feature extraction in the DP mechanism is compulsory. The fixed single-branch architecture translates a vague interpretation of how global and local relationships are utilized across different layers. Moreover, it hypothesizes that global and local information hold identical levels of importance since the numbers of global and local extractors are set the same. Without further exploration, one may naturally wonder if the arrangement is optimal, as questions like: Are global and local information equally crucial in every layer? If not, is it possible to identify a pattern?

Therefore, our intuition is to seek a scheme that retrieves global and local information in parallel and investigates the best possible way to utilize them while performing speech separation. Such an investigation is crucial yet not emphasized and discussed in previous works. Furthermore, the respective outcome can help researchers design better speech separation architectures. In light of this, we propose a novel Global and Local context-Aware Speech Separation method entitled GLASS. GLASS adapts Branchformer [11], which combines self-attention and convolutional layers into a parallel design, to the speech separation task. Self-attention is valuable for capturing long temporal relationships, while convolutional layers are competent to extract regional features within its receptive field.

Through extensive evaluation under anechoic and noisy settings, GLASS achieves state-of-the-art performance. Interestingly, we found a regular pattern of when local and global information matters and can even reform the model accordingly to gain improved results. This contribution underscores the significance of parallel processing and utilizing proper dependency for enhanced speech separation architectures, providing valuable insights for future research and design considerations.

II. GLASS

A. Overall Structure

The overall description of the proposed GLASS framework follows the encoder-decoder masking-based strategy, as depicted in Fig. 1. This strategy works as follows: Initially, the encoder transforms the mixture $x = \sum_{i=1}^{C} s_i + n \in \mathbb{R}^T$, which contains audio from C active speakers and additive noise n, into an STFT-like representation w that characterizes the signal:

$$w = \text{ReLU}(f_{\text{enc}} * x) \in \mathbb{R}^{N \times L}, \tag{1}$$

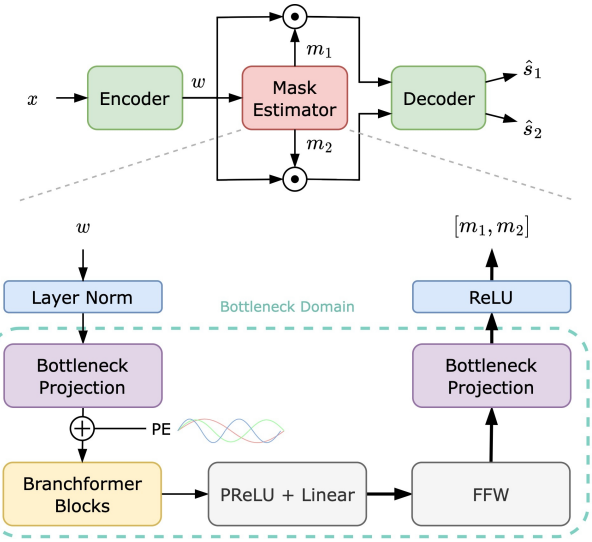

Fig. 1. The overall architecture of proposed method. This figure demonstrates an example of a mixture including two sources, and the bold arrows indicate that there are two vectors.

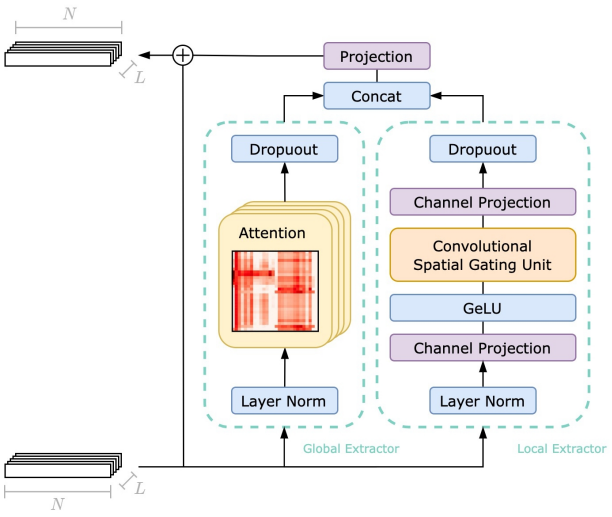

Fig. 2. A Branchformer block consists a global and a local context extractor.

where f_{enc}, $*$, and ReLU denote the encoding matrix, the convolution operation, and a rectified linear unit, respectively, and N and L are the feature size and the number of frames.

Then, by feeding w, the mask estimator produces C masks $\{m_i\}$ for each active speaker in the mixture. Finally, the decoder reconstructs each estimated source $\hat{s}_i$ by deconvolving the masked representation:

$$\hat{s}_i = f_{\text{dec}} * (w \odot m_i) \in \mathbb{R}^T, \tag{2}$$

where f_{dec} and $\odot$ denote the decoding matrix and element-wise multiplication. The analytic filters in f_{enc} and f_{dec} are learned from uni-dimensional convolution (Conv1d) and its transposition, respectively, both having N bases with a length of k. The objective is to minimize the distance between s_i and $\hat{s}_i$.

A more profound illustration of the mask estimator is also visualized in the lower part of Fig. 1. The mask estimator contains one pair of bottleneck projections, multiple Branchformer blocks, a linear layer, and feed-forward networks (FFWs). The representation w is first layer-normalized, and feature-wise downsized to a bottleneck dimension. Prior to the Branchformer blocks, positional encoding (PE) [25] is applied to inject information on the order of frame sequence. Afterward, the downsized representation feeds R Branchformer blocks to realize an enhanced vector space that alleviates the task difficulty for the subsequent linear layer. This linear layer is where the separation is carried out, transforming each of size N into a matrix of size $C \times N$. Ultimately, each of the C representations is transferred to FFWs, an upsizing projection, and a ReLU to generate a non-negative mask.

Branchformer was initially designed to tackle automatic speech recognition (ASR) and spoken language understanding (SLU) tasks. However, their objectives are quite different from ours. For example, the output dimension to which each task wishes to map differs. Speech separation or enhancement, in particular, remains the same dimension as the input audio, whereas ASR or SLU maps to tokens with fewer quantities. Therefore, we carefully modified Branchformer to meet our objectives.

B. Branchformer Blocks

Two parallel branches mainly constitute a Branchformer block, as depicted in Fig. 2. Both branches share the same input but focus on different relation ranges that complement each other. For an input w, the global branch outputs a globally-viewed tensor $u \in \mathbb{R}^{N \times L}$ produced by the global context extractor, whereas the local branch outputs a locally-tuned tensor $v \in \mathbb{R}^{N \times L}$ produced by the local context extractor. The outputs of two branches are then merged, with the original input added as a residual connection to smooth the output when sources are absent. At the beginning and end of each branch, layer normalization [12] and dropout [13] are employed, respectively. For brevity, we omit them in the following formulation.

Global Context Extractor. The global extractor exploits Multi-Head Self-Attention (MHSA) [25], which can be expressed as follows:

$$\begin{aligned} &Q_i = (\mathbf{W}_i^Q w)^T, K_i = (\mathbf{W}_i^K)^T, V_i = (\mathbf{W}_i^V)^T; \\ &\text{head}_i = \text{SA}(Q_i, K_i, V_i) = \mathcal{S}(\frac{Q_i K_i^T}{\sqrt{N}}) V_i; \\ &u = \text{MHSA}(w) = \mathbf{W}_{\text{head}}[\text{head}_0, ..., \text{head}_{h-1}]_{\mathbf{1}}^T, \end{aligned} \tag{3}$$

where $\mathbf{W}_i^{\{Q,K,V\}} \in \mathbb{R}^{N/h \times N}$ and $\mathbf{W}_{\text{head}} \in \mathbb{R}^{N \times N}$ are learnable linear layers, $\mathcal{S}(\cdot)$ denotes a softmax activation function, and h denotes the number of attention heads. The concatenation along the $(\mathbf{i}+\mathbf{1})$'th dimension is denoted by $[\cdots]_{\mathbf{i}}$.

Local Context Extractor. The local extractor exploits a convolutional-gated Multi-Layer Perceptron (cgMLP) [14], which contains one pair of feature-wise projections, a Gaussian

Error Linear Unit (GELU) [15], and a Convolutional Spatial Gating Unit (CSGU). The sequential procedure can be formulated as follows:

$$\begin{aligned}
&\tilde{w} = \text{GELU}(\mathbf{W}^{\text{up}}_{\text{mlp}} w), \quad \tilde{w} \in \mathbb{R}^{F \times L};\\
&[\tilde{w}_1, \tilde{w}_2]_{\mathbf{0}} = \tilde{w}, \quad \tilde{w}_i \in \mathbb{R}^{F/2 \times L};\\
&\tilde{w}'_2 = \text{CSGU}(\tilde{w}_2) = \text{DWConv}(\text{LayerNorm}(\tilde{w}_2));\\
&\tilde{w}'_1 = \tilde{w}_1 \odot \tilde{w}'_2;\\
&v = \mathbf{W}^{\text{down}}_{\text{mlp}} \tilde{w}'_1,
\end{aligned} \tag{4}$$

where F denotes the hidden dimension that is usually larger than that of input, DWConv denotes depth-wise convolution, and $\mathbf{W}^{\text{up}}_{\text{mlp}} \in \mathbb{R}^{F \times N}$ and $\mathbf{W}^{\text{down}}_{\text{mlp}} \in \mathbb{R}^{N \times F/2}$ are learnable linear layers as well.

Merging Branches. After obtaining u and v as in Eqs. (3) and (4), two merging methods are designed to gain an intermediate representation w' that captures global and local dependencies. The first is simply concatenating them and performing projection back to the original dimension by a linear layer $\mathbf{W}_{\text{cat}} \in \mathbb{R}^{N \times 2N}$:

$$w' = \mathbf{W}_{\text{cat}}[u, v]_0 \in \mathbb{R}^{N \times L}. \tag{5}$$

Concatenation-based merging is straightforward. However, due to its lack of flexibility, the alternative may be a weighted sum of two branches:

$$w' = \alpha_u u + \alpha_v v \tag{6}$$

The weights, α_u and α_v, indicate how global and local relationships are incorporated. This can be particularly useful in cases where the relative importance of the global and local information may vary depending on different layers. While there are multiple ways to learn α_u and α_v, attention-based pooling [16] is employed here as follows:

$$\begin{aligned}
\alpha'_u &= \mathcal{S}(\frac{\mathbf{W}^{u,1}_{\text{pool}} u}{\sqrt{N}})(\mathbf{W}^{u,2}_{\text{pool}} u)^T \in \mathbb{R};\\
\alpha'_v &= \mathcal{S}(\frac{\mathbf{W}^{v,1}_{\text{pool}} v}{\sqrt{N}})(\mathbf{W}^{v,2}_{\text{pool}} v)^T \in \mathbb{R};\\
[\alpha_u, \alpha_v]_{\mathbf{0}} &= \mathcal{S}([\alpha_u, \alpha_v]_{\mathbf{0}}) \in \mathbb{R}^2,
\end{aligned} \tag{7}$$

where $\mathbf{W}^{\{u,v\},\{1,2\}}_{\text{pool}} \in \mathbb{R}^{1 \times N}$ are learnable linear layers.

C. Comprehending Branches

If we regard every linear layer $\mathbf{W}$ as interpolation or subsampling, then the understanding of Branchformer's architecture can be comprehensible. For the global branch, each attention map, $\mathcal{S}(Q_i K_i^T/\sqrt{N}) \in \mathbb{R}^{L \times L}$, explains how the network adjusts its focus based on long-temporal information. In contrast, the local branch modifies features within a single frame. Specifically, CSGU processes the neighboring values via convolution kernels, resembling sub-band processing. At the same time, the gating design and up- and down-sizing projection layers can be interpreted as full-band processing.

III. Experiments

A. Datasets

We validate GLASS on the popular WSJ0-2mix dataset [3] under the anechoic setting and use the improvement of SI-SDR [17] and SDR as the evaluation metrics. WSJ0-2mix is generated from the Wall Street Journal (WSJ) dataset [18] and consists of mixed speech utterances from two distinct speakers with random SDR between 0 dB and 5 dB. The training, validation, and test sets contain 30, 10, and 5 hours of speech data. Furthermore, we perform experiments in noisy settings. We rely on WHAM! [19], where each two-speaker utterance from the WSJ0-2mix dataset is mixed with a unique noise sample recorded in ambient environments such as coffee shops, restaurants, and bars. The models are supposed to simultaneously perform speech separation and denoising to extract clean signals from such mixed data. Finally, all speech data are sampled at 8 kHz.

B. Model Configurations

GLASS is implemented using PyTorch [20], and the experiments are conducted using the Speechbrain toolkit [21]. For the encoder and decoder, we set the number of bases $N = 256$ and the kernel size $k = 16$ with a stride factor of 8. Regarding the mask estimator, the repetition of the Branchformer block R is 8, 12, and 16. We use $h = 8$ parallel attention heads inside each global branch and the hidden dimension $F = 2048$ for each local branch. The kernel size used in CSGU is 17. All the dropout layers are applied with a probability of 0.1. For model training, we optimize the model using the Adam optimizer [22] with a learning rate of $1.5e-4$, which is halved after 5 epochs without improvement. Finally, the models are trained over 150 epochs with Permutation Invariant Training [23] and SI-SDR losses [3].

IV. Results

A. Difference in Context Extractor

To demonstrate its effectiveness in modeling long- and short-term dependencies, we first compare GLASS with two context extractors: Transformer [25] and Conformer [24]. While Transformer is good at modeling long-range global context, it is less capable of extracting fine-grained local feature patterns. Conformer, on the other hand, employs a single-branch architecture, making it challenging to analyze how local and global interactions are utilized in different layers. Furthermore, similar to the DP mechanism, the fixed interleaving pattern between self-attention and convolution in Conformer might not be optimal.

The result is shown in Tab. I. We can observe that if the extractor readily possesses both local and global views, it is less favored by incorporating the DP mechanism. Positive examples are Conformer without DP mechanism and GLASS, while the negative example is Transformer without DP mechanism. Furthermore, while the DP mechanism is beneficial

TABLE I
THE SI-SDR AND SDR IMPROVEMENT SCORES OBTAINED BY GLASS AND DIFFERENT EXTRACTOR IN WSJ0-2MIX. GLASS USING CONCATENATION-BASED MERGING WITH 8 LAYERS IS ABBREVIATED AS "GLASS-C8", AND THE WEIGHTED-SUM ONE AS "GLASS-S8."

Extractor	Total layers	DP	SI-SDRi ↑	SDRi ↑
Transformer	8	×	17.6	18.0
		✓	18.8	19.1
Conformer	8	×	18.6	18.8
		✓	18.3	18.5
GLASS-c8	8	×	18.6	18.9
GLASS-s8	8	×	19.0	19.1

when integrated with Transformer (equivalent to a smaller-sized SepFormer [8]), probably due to the division of labor, GLASS using weighted-sum merging attains better outcomes.

B. Comparison with Previous Works

Here, we increase the number of the Branchformer blocks R in GLASS from 8 to 12 and 16 to gain a higher result. Tab. II compares the performance achieved by GLASS with the best results obtained by well-known methods tested on the WSJ0-2mix and WHAM!. As seen, GLASS families stand competitively; notably, "GLASS-s16" achieves state of the art. Although our method is slightly inferior to Wavesplit [9] for anechoic data, it still manages to discern noises and separate the clean sources better than Wavesplit on WHAM!, where source separation and speech enhancement are practiced simultaneously and thus closer to the real environment. The same explanation can also be given to DPTNet [7], which seems to be more parameter-wise efficient but fails for realistic data. It is noteworthy that Wavesplit and VSUNOS [6] leverage additional speaker information, either trained from scratch or with pre-trained embeddings.

Furthermore, the results from Tabs. I and II suggest that GLASS performs better with weighted-sum merging than with concatenation, which contradicts our initial expectations. We reckon the underlying cause is that concatenating the embeddings and projecting them back to the original dimension may lead to a certain loss of information. When performing one-to-more mapping as speech separation, as opposed to classification tasks like ASR and SLU, the loss of information is treated more severely and has to be addressed more carefully. The weighted-sum merging can mitigate this issue by providing a more controlled interpolation between the two embeddings.

C. Receptive Field in Local Branch

The choice of the sub-band range is crucial yet hard to determine in practice. As metaphorized in Sec II-C, the range of the sub-band translates to the receptive field in the context of Conv1d. We are curious about the impact of varying kernel sizes in Conv1d, and the findings are presented in Tab III. It shows diminishing rewards when the kernel size increases from 17 to 33, while performance is marginally worse when decreasing the kernel size from 17 to 9. A similar outcome can be found in [27].

D. Visualizing Branch Weights

As formulated in Eq. (7), the branch weights are determined by the given context and thus may be dynamic accordingly. This raises an interesting question of whether weights in the same layer are consistent or divergent regarding different inputs. To address such concern, we visualize each branch weights gained from 3000 testing trials via box plots, as plotted in Fig. 3.

We can observe four phenomena:

1) The first quarter of layers primarily retains local information while discarding global information (see data points cropped in the orange box).
2) Weights with high dynamic ranges are found in two to three levels preceding the last two layers, where many outliers exist (see data points cropped in the green box). It reveals that these layers adjust their priority to global or local embeddings when facing different inputs.
3) Aside from the layers mentioned above, the weights in the remaining layers are low-dispersive, indicating that the division of labor is distinct in each layer.
4) On average, the local information is utilized more in the overall percentage than the global information (see the dashed lines), which opposes to the hypothesis of equal importance.

We also conducted an experiment in which the global branches of the first four layers in GLASS-s12 were dropped since their weights were close to zero, according to Fig. 3. Relative to the original GLASS-s12, this arrangement yields comparable SI-SDRi performance (20.4 for WSJ0-2mix and 16.1 for WHAM), but converges faster. Due to the specific vision we have assigned for those layers, the subsequent layers can be more focused on their priorities and have a better grasp of what kind of embedding to employ. This highlights the importance of using global and local dependencies in an appropriate order, not to mention the reduction of parameters that leads to faster convergence. Furthermore, these findings suggest that an interleaving pattern of local and global extractors is suboptimal.

E. Ablation Study

Inspired by the visualization of the branch weights, a question naturally emerges: What if only one branch is used while the other is dropped? We conduct an ablation study, shown in Tab. IV, verifying whether both the global context extractor and the local context extractor are useful. It turns out that the performance of the model using only the local extractor is superior to the one using only the global extractor, even though Transformer is unarguably a more powerful design. The result gives hints on an essential and supplementary role for local and global features respectively, yet they complement each other when using both branches. The same finding can also be indicated from Fig. 3, where the overall weight for the local branch is greater than that of the global branch.

TABLE II
THE SI-SDR AND SDR IMPROVEMENT SCORES OBTAINED BY GLASS AND PREVIOUS WORKS IN WSJ0-2MIX AND WHAM!.

Method	Size	WSJ0-2mix		WHAM!	
		SI-SDRi ↑	SDRi ↑	SI-SDRi ↑	SDRi ↑
Chimera++ [26]	32.9M	11.5	12.0	9.9	-
Deep CASA [1]	12.8M	17.7	18.0	-	-
BiLSTM-TasNet [2]	23.6M	13.2	13.6	12.0	-
Conv-TasNet [3]	5.1M	15.3	15.6	12.7	-
DPRNN [4]	2.6M	18.8	19.0	13.9	-
SuDoRM-RF [5]	2.6M	18.9	-	-	-
VSUNOS [6]	7.5M	20.1	-	15.2	-
DPTNet [7]	2.7M	20.2	20.6	14.9	15.3
SepFormer [8]	26M	20.4	20.5	16.3	16.7
Wavesplit [9]	29M	21.0	21.2	15.4	15.8
GLASS-c12	14.8M	20.0	20.3	15.7	16.1
GLASS-s12	14.1M	20.3	20.5	16.1	16.2
GLASS-c16	19.9M	20.4	20.6	16.2	16.4
GLASS-s16	18.6M	20.8	21.0	16.5	16.8

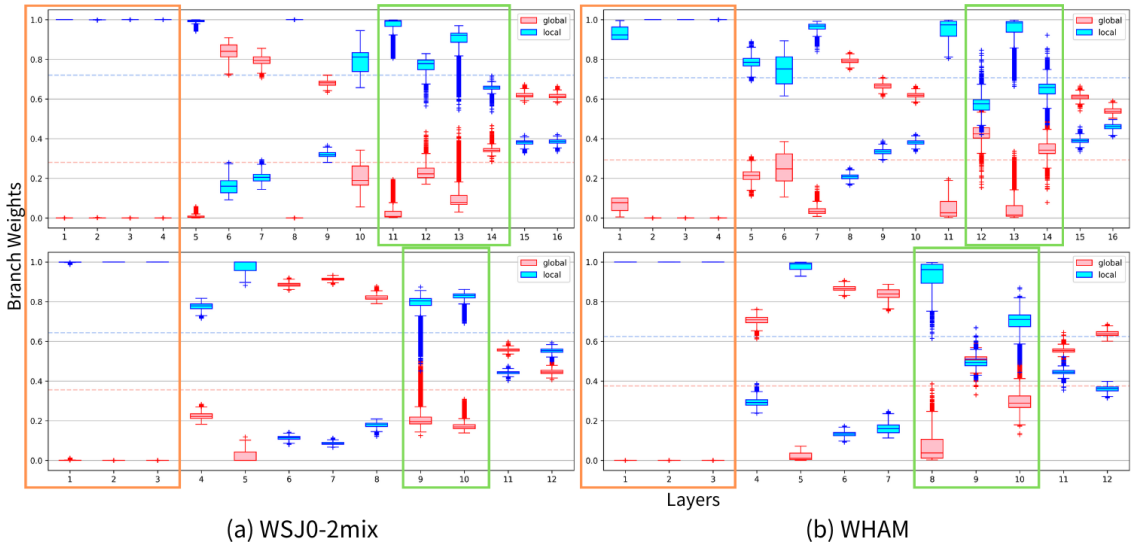

Fig. 3. The box plot of branch weights for model with 12 and 16 layers. '+' symbols denote outliers, and dashed lines plot the mean of all layer weights.

TABLE III
THE EFFECT ON DIFFERENT RESPECTIVE FIELD IN LOCAL BRANCH.

Model	kernel size			
	$k = 9$	$k = 17$	$k = 25$	$k = 33$
GLASS-s16	20.82	20.84	20.76	20.57

TABLE IV
ABLATION STUDY ON BRANCH DROPOUT.

local	global	SI-SDRi ↑	SDRi ↑
✓		18.75	18.91
	✓	18.48	18.72
✓	✓	20.84	21.01

F. Time complexity

In Fig. 4, we plot the execution time of each model. The first thing that comes into sight is the linear function against the step function. Being also categorized as a frame-online system, Conv-TasNet [3] has the same linear time complexity as GLASS. Regardless of the latency gap caused by their respective parameter size, GLASS holds a large margin of better separation quality than Conv-TasNet. On the other hand, the step function describes the chunked modeling of the DP-based method. Albeit this stepping characteristic helps reduce the time complexity if a long recording is readily in hand [28], the fact that it takes more time to separate short frames disqualifies the DP mechanism from being a frame-

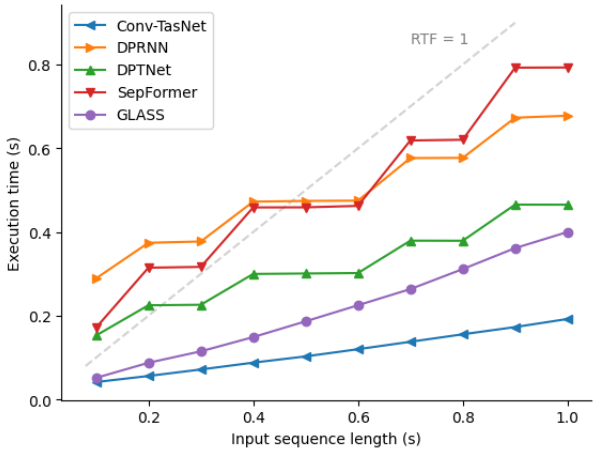

Fig. 4. Execution time, ran by GTX1080Ti, plotted as a function of the input length. RTF stands for real-time factor.

online application. Using GLASS, instead, reserves more spare time for downstream speech applications, say online speaker diarization [29], to operate.

V. Conclusions

In this study, we introduce GLASS, a novel approach that explores the use of global and local information in speech separation. GLASS employs a parallel design with branches extracting and merging features based on contextual importance. Evaluation results demonstrate the state-of-the-art performance of GLASS. The learned branch weights provide insights into an optimal pattern for integrating global and local information, enhancing interpretability and understanding in speech separation. Our work contributes to advance future endeavors in optimizing network architectures for speech separation.

VI. Acknowledgement

This work was supported in part by Realtek Semiconductor Corporation under Grant Number 112KK010. Any findings and implications in the paper do not necessarily reflect those of the sponsors.

References

[1] Y. Liu and D. Wang, "Divide and conquer: A deep casa approach to talker-independent monaural speaker separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 27, no. 12, pp. 2092–2102, 2019.

[2] Y. Luo and N. Mesgarani, "Real-time single-channel dereverberation and separation with time-domain audio separation network." in *Interspeech*, 2018.

[3] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal timefrequency magnitude masking for speech separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.

[4] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP*, 2020.

[5] E. Tzinis, Z. Wang, and P. Smaragdis, "Sudo rm-rf: Efficient networks for universal audio source separation," in *MLSP*, 2020.

[6] E. Nachmani, Y. Adi, and L. Wolf, "Voice separation with an unknown number of multiple speakers," in *ICML*, 2020.

[7] J. Chen, Q. Mao, and D. Liu, "Dual-path transformer network: Direct context-aware modeling for end-to-end monaural speech separation," in *Interspeech*, 2020.

[8] C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention Is All You Need In Speech Separation," in *ICASSP*, 2021.

[9] N. Zeghidour and D. Grangier, "Wavesplit: End-to-End speech separation by speaker clustering," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 2840–2849, 2021.

[10] Fu. Chao, J. Hung, T. Sheu, B. Chen, "A time-reversal enhancement network with cross-domain information for noise-robust speech recognition," *IEEE Multimedia*, vol. 29, no. 1, pp. 114-124, 2022.

[11] Y. Peng, S. Dalmia, I. R. Lane and S. Watanabe, "Branchformer: Parallel MLP-Attention Architectures to Capture Local and Global Context for Speech Recognition and Understanding," in *ICML*, 2022.

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

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[14] J. Sakuma, T. Komatsu, and R. Scheibler, "MLP-based architecture with variable length input for automatic speech recognition," in *ICLR*, 2022.

[15] D. Hendrycks, and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[16] C. Wu, F. Wu, T. Qi, Y. Huang, and X. Xie, "Fastformer: Additive attention can be all you need," *arXiv preprint arXiv:2108.09084*, 2021.

[17] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – Half-baked or Well Done?" in *ICASSP*, 2019.

[18] J. S. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) Complete," *LDC93S6A. Web Download. Philadelphia: Linguistic Data Consortium*, 1993.

[19] G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux, "Wham!: Extending speech separation to noisy environments," in *Interspeech*, 2019.

[20] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] M. Ravanelli et al., "SpeechBrain: A General-Purpose Speech Toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015.

[23] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, 2017.

[24] A. Gulati et al., "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech*, 2020.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[26] Z.-Q. Wang, J. L. Roux, and J. R Hershey, "Alternative objective functions for deep clustering," in *ICASSP*, 2018.

[27] Y. Tsao, K. Ho, J. Hung, and B. Chen, "Adaptive-FSN: Integrating Full-Band Extraction and Adaptive Sub-Band Encoding for Monaural Speech Enhancement," in *SLT*, 2023.

[28] C. Li et al., "Dual-Path Modeling for Long Recording Speech Separation in Meetings," in *ICASSP*, 2021.

[29] E. Gruttadauria, M. Fontaine, S. Essid, "Online speaker diarization of meetings guided by speech separation," in *ICASSP*, 2024.