A Comparative Study on Positional Encoding for Time-frequency Domain Dual-path Transformer-based Source Separation Models

Kohei saijo*, Tetsuji Ogawa*
*Department of Communications and Computer Engineering, Waseda University, Tokyo, Japan

***Abstract*—In this study, we investigate the impact of positional encoding (PE) on source separation performance and the generalization ability to long sequences (length extrapolation) in Transformer-based time-frequency (TF) domain dual-path models. The length extrapolation capability in TF-domain dual-path models is a crucial factor, as it affects not only their performance on long-duration inputs but also their generalizability to signals with unseen sampling rates. While PE is known to significantly impact length extrapolation, there has been limited research that explores the choice of PEs for TF-domain dual-path models from this perspective. To address this gap, we compare various PE methods using a recent state-of-the-art model, TF-Locoformer, as the base architecture. Our analysis yields the following key findings: (i) When handling sequences that are the same length as or shorter than those seen during training, models with PEs achieve better performance. (ii) However, models without PE exhibit superior length extrapolation. This trend is particularly pronounced when the model contains convolutional layers.**

***Index Terms*—positional encoding, source separation, time-frequency domain dual-path modeling, Transformer**

I. Introduction

With the advancements in neural networks, source separation has witnessed remarkable progress over the past decade. Early approaches, such as time-frequency (TF) masking [1], [2], have evolved into time-domain separation networks [3], [4], which have gained significant attention. More recently, TF-domain dual-path modeling has demonstrated superior performance, particularly in challenging noisy reverberant conditions [5], [6]. Simultaneously, a variety of network architectures, such as convolutional networks [4], recurrent neural networks [7], Transformers [8], and state-space models [9], have been explored for source separation. Among these, Transformer-based models exhibit unique characteristics that are either unattainable or yet to be fully validated in other architectures, such as prompting to specify the task [10], [11]. Consequently, Transformer-based models have attracted significant research interest, and recent TF-domain dual-path models have achieved state-of-the-art performance [12].

However, Transformer-based models often exhibit limited generalizability when applied to sequences longer than those seen during training. This *length extrapolation* problem is critical in source separation, as models are frequently required to process input sequences with a longer length than those seen during training. In the context of TF-domain dual-path models, length extrapolation ability also affects generalizability to signals with unseen sampling rates. Since short-time Fourier transform (STFT) with a fixed-duration window and hop size produces STFT spectra with a constant TF resolution regardless of the sampling rate, TF-domain dual-path models can handle inputs with various sampling rates [10]. However, as the number of frequency bins in the STFT representation changes with the input sampling rate, the model's ability to generalize to unseen sampling rates is inherently dependent on its length extrapolation capability.

In Transformer-based models, positional encoding (PE) plays a crucial role, and the length extrapolation ability is highly dependent on the choice of PE. For example, the absolute PE (APE) introduced in the original Transformer is known to exhibit poor generalization to longer sequences. To mitigate this limitation, various PE methods have been proposed. One such approach is rotary positional encoding (RoPE) [13], which is widely adopted in large language models (LLMs). RoPE enhances length extrapolation by embedding positional information exclusively in the key and query vectors. Some relative positional encoding (RPE) methods inspired by this concept have been proposed, further enhancing length extrapolation performance [14], [15].

Meanwhile, prior studies in computer vision have investigated the use of convolutional layers as PE, leveraging the fact that convolutional layers can implicitly encode positional information through zero-padding [16]. Since convolution-based PEs extract position information from local neighborhoods alone, they may demonstrate stronger generalization to longer sequences than those seen during training [17]. Given that many high-performing speech processing models incorporate both self-attention and convolution [12], [18], they may be able to utilize position information encoded by convolution to enhance length extrapolation. However, this approach remains unexplored in the context of source separation. Indeed, although previous research has compared PEs in speech enhancement, it was limited to APE and RPE, without considering other PEs, such as convolution or RoPE.

In this work, we investigate the impact of positional encoding (PE) in Transformer-based TF-domain dual-path source separation models. To this end, we examine several PE methods from different categories within the framework of TF-Locoformer [12], a state-of-the-art model. TF-Locoformer incorporates both self-attention and convolution, enabling us to analyze not only the effects of explicit PEs, such as RPE, but also the implicit positional encoding induced by convolutional

layers. We systematically compare four types of PEs, absolute, relative, rotary, and convolution, and evaluate their impact on source separation performance and length extrapolation.

II. Background

A. TF-Locoformer

TF-Locoformer is based on TF-domain dual-path modeling, where frequency and temporal modeling are done alternately [5], [6]. Let us denote a M-channel mixture of N sources in the STFT domain $\boldsymbol{S} \in \mathbb{R}^{2\times N\times M\times T\times F}$ as $\boldsymbol{X} = \sum_n \boldsymbol{S}_n \in \mathbb{R}^{2\times M\times T\times F}$, where T and F are the number of frames and frequency bins, and $n = 1, \ldots, N$ is the source index. 2 corresponds to the real and imaginary (RI) parts.

The input $\boldsymbol{X}$ is first reshaped into $2M \times T \times F$ and then encoded into an initial feature $\boldsymbol{Z}$ with feature dimension D:

$$\boldsymbol{Z} = \mathrm{gLN}(\mathrm{Conv2D}(\boldsymbol{X})) \in \mathbb{R}^{D\times T\times F}, \tag{1}$$

where gLN is global layer normalization [4]. For frequency modeling, we interpret the feature $\boldsymbol{Z}$ as a stack of T arrays, each with shape $D \times F$. This is implemented by permuting the dimensions of $\boldsymbol{Z}$ to the form $T \times D \times F$. Frequency modeling is then performed as follows:

$$\boldsymbol{Z} \leftarrow \boldsymbol{Z} + \mathrm{ConvSwiGLU}(\boldsymbol{Z})/2, \tag{2}$$

$$\boldsymbol{Z} \leftarrow \boldsymbol{Z} + \mathrm{MHSA}(\mathrm{Norm}(\boldsymbol{Z})), \tag{3}$$

$$\boldsymbol{Z} \leftarrow \boldsymbol{Z} + \mathrm{ConvSwiGLU}(\boldsymbol{Z})/2, \tag{4}$$

where MHSA stands for multi-head self-attention with H heads [19]. In the original TF-Locoformer, RoPE [13] is used to encode the positional information of each frequency bin. Root-mean-square group normalization (RMSGroupNorm), a derivative of RMSNorm [20], is adopted as the Norm layer. The $\mathrm{ConvSwiGLU}$ module is a feed-forward network based on the convolution layers with a kernel size of K and stride of S and the swish gated linear unit (SwiGLU) activation [21]:

$$\boldsymbol{Z} \leftarrow \mathrm{Norm}(\boldsymbol{Z}), \tag{5}$$

$$\boldsymbol{Z} \leftarrow \mathrm{Swish}(\mathrm{Conv1D}(\boldsymbol{Z})) \otimes \mathrm{Conv1D}(\boldsymbol{Z}), \tag{6}$$

$$\boldsymbol{Z} \leftarrow \mathrm{Deconv1D}(\boldsymbol{Z}). \tag{7}$$

Temporal modeling is done in the same way by viewing the feature $\boldsymbol{Z}$ as a stack of F arrays of shape $D \times T$, where T is the sequence length (i.e., we permute the dimension order of $\boldsymbol{Z}$ to $F \times D \times T$). After alternating between frequency and temporal modeling B times, the final feature $\boldsymbol{Z}$ is used to estimate the RI components of N sources:

$$\hat{\boldsymbol{S}} = \mathrm{DeConv2D}(\boldsymbol{Z}) \in \mathbb{R}^{2\times N\times M\times T\times F}. \tag{8}$$

B. Sampling-frequency independence (SFI) of TF-domain dual-path models

In general, source separation models may struggle when processing input signals sampled at frequencies different from those used during training. This performance degradation is primarily attributed to differences in input resolution. However, when using STFT, the resolution of each TF bin remains unchanged as long as the fixed-duration STFT window and hop sizes are maintained, regardless of the input sampling frequency. The STFT spectra of signals with different sampling rates but the same duration contain the same number of frames but a different number of frequency bins, while preserving a consistent resolution. Leveraging this property, a separation model that does not explicitly depend on the number of frequency bins can exhibit sampling frequency invariance (SFI). SFI allows, for example, a model trained at a lower sampling rate to be effectively applied to data with a higher sampling rate, thereby improving training efficiency.

Many TF-domain dual-path models inherently satisfy this requirement, as they employ sequence modeling that is independent of sequence length in the frequency dimensions. Indeed, several models have been empirically shown to generalize well to sampling rates higher than those used during training [10]. TF-Locoformer also meets this criterion and is expected to exhibit SFI; however, its length extrapolation ability may strongly depend on the choice of PE. In this study, we conduct a case study on TF-Locoformer, systematically comparing different PE methods in Transformer-based TF-domain dual-path models and evaluating their effectiveness in improving length extrapolation performance.

III. Positional encodings

In this work, we investigate the following four PEs.

A. Absolute positional encoding (APE)

As the APE, we use the sinusoidal positional encoding used in the original Transformer [19]. The position embedding $\boldsymbol{P}_i \in \mathbb{R}^D$ at the position i is defined as

$$P_{d,i} = \begin{cases} \sin\left(10000^{-\frac{d}{D}} \cdot i\right), & \text{if } d \text{ is even} \\ \cos\left(10000^{-\frac{d-1}{D}} \cdot i\right), & \text{if } d \text{ is odd,} \end{cases}$$

where $d = 1, \ldots, D$. $\boldsymbol{P}$ is added to the encoded feature $\boldsymbol{Z}$ after the processing in Eq. (1). As the TF-Locoformer has sequence modeling for both time and frequency dimensions, we add two positional embeddings separately. We first make a positional embedding for time frames $\boldsymbol{P}^{\mathrm{time}} \in \mathbb{R}^{D\times T}$, broadcast it to $D \times T \times F$, and add it to $\boldsymbol{Z}$. We then make another embedding $\boldsymbol{P}^{\mathrm{freq}} \in \mathbb{R}^{D\times F}$, broadcast it to $D \times T \times F$, and add it to $\boldsymbol{Z}$.

B. Kernelized relative positional embedding (KERPLE)

Relative positional encoding (RPE) encodes the distances between each pair of elements in the input sequence. It is known to generalize to longer sequences better than the APE. Although there are many RPEs, we choose KERPLE [15], which has shown better length generalization ability in speech enhancement[1] [22]. KERPLE kernelizes the distance between i-th and j-th elements in a sequence with conditionally positive definite kernels:

$$\boldsymbol{P}_{i,j} = -r_1(\log(1 + r_2|i - j|)), \tag{9}$$

[1]Although KERPLE is designed for causal processing, we decided to use it since it has been reported to work in non-causal speech enhancement [22]

where $r_1 > 0$ and $r_2 > 0$ are learnable parameters. Unlike the APE, RPE is calculated in each MHSA layer, and $\boldsymbol{P}$ is added to the attention matrix.

C. Rotary positional encoding (RoPE)

RoPE is used in the original TF-Locoformer to encode the position information. RoPE encodes the position by applying rotation matrices to the key and query separately before computing the attention matrix. Although RoPE is one of the most widely used PE and is employed in some LLMs [23], recent works have shown that RoPE may not generalize to longer sequences than those seen during training [24].

D. No positional encoding (NoPE)

In prior work in the field of computer vision, it has been shown that convolutional layers inherently encode positional information [16], allowing Transformers with convolutional layers to function effectively even without explicit PE [25]. Removing PE not only reduces computational costs but also may facilitate generalization to sequences longer than those encountered during training [17]. Inspired by these findings, this study investigates TF-Locoformer with no positional encoding (NoPE). Since TF-Locoformer incorporates convolutional layers, it is likely to work even in the absence of PE. We investigate whether the model generalizes better to long sequences compared to its counterpart that includes PE.

IV. Experiments

A. Datasets

WHAMR! [26] contains noisy reverberant two-speaker mixtures sampled at 8 kHz. Only the first channel was used to evaluate performance in a monaural setup. Speech and noise signals are sourced from WSJ0 [27] and WHAM! [28], respectively. The dataset comprises 20000, 5000, and 3000 mixtures for training, validation, and testing, respectively. The model is trained to perform dereverberation, denoising, and separation simultaneously.

MUSDB18-HQ contains 100 and 50 stereo songs sampled at 44.1 kHz for training and testing, respectively. For data partitioning, we adopted a commonly used split, allocating 86 songs for training and 14 for validation. The goal is to separate the mixture into the individual stems (vocals, bass, drums, and other). The model takes stereo mixtures as input and estimates stereo-separated signals.

B. Model configuration

When training on WHAMR!, we used the small version of the TF-Locoformer [12]. Specifically, unless otherwise stated, we set the parameters as follows: $D = 96$, $B = 4$, $K = 8$, $S = 1$, and $H = 4$ (notation follows Section II-A).

For training on MUSDB, to reduce computational cost, we replace the Conv2D and DeConv2d layers in Eq. (1) and Eq. (8), respectively, with the band-split encoder and band-wise decoding module [29]. The band-split encoder decomposes the input spectrogram $\boldsymbol{X} \in \mathbb{R}^{2M \times T \times F}$ with F frequency bins into Q non-overlapping subband spectrograms $\boldsymbol{X}_q \in \mathbb{C}^{T \times b_q}$ $(q = 1, \ldots, Q)$, where the pre-defined bandwidths b_q satisfy $\sum_q b_q = F$. $\boldsymbol{X}_q$ undergoes normalization and a linear transformation, producing a feature representation $\boldsymbol{Z}_q \in \mathbb{R}^{D \times T \times 1}$. The Q features are then concatenated and result in a feature $\boldsymbol{Z} \in \mathbb{R}^{D \times T \times Q}$, which is processed by TF-Locoformer blocks. The band-wise decoding module splits $\boldsymbol{Z}$ into Q sub-features and decodes them to obtain band-wise masks (see [29] for more details). We used the same band-split configuration as [30] with $Q = 62$ bands and refer to this model as the band-split Locoformer (BS-Locoformer).

C. Training and evaluation details

The training was conducted with a single Nvidia RTX A5000 GPU. For WHAMR!, we trained the TF-Locoformer for up to 150 epochs, with each epoch consisting of 5000 training steps. We used the AdamW optimizer [31] with a weight decay factor of 1e-2. The learning rate was linearly increased from 0 to 1e-3 over the first 4000 training steps, kept constant for the first 75 epochs, and subsequently decayed by a factor of 0.5 if the validation loss did not improve for three consecutive epochs. Early stopping was applied when the best validation score remained unchanged for ten epochs. The batch size was four. Gradient clipping was applied with a maximum gradient L_2-norm of five. The negative scale-invariant signal-to-distortion ratio (SI-SDR) [32] was used as the loss function. No data augmentation was applied. To enhance training efficiency, we employed the automatic mixed precision technique and flash attention [33].

For MUSDB, we trained the BS-Locoformer for 200 epochs without early stopping, with each epoch consisting of 2500 training steps. The optimizer and learning rate scheduling followed the WHAMR! configuration, except for the learning rate decay factor of 0.8. We applied an unsupervised source activity detection method from [29] with a segment duration of eight seconds to remove silent segments from the training data. During training, dynamic mixing was performed at each training step, where a segment from each stem was randomly selected, RMS-normalized, scaled by a gain uniformly sampled from [-10, 10] dB, and mixed. Following [29], each segment was dropped with a probability of 0.1 before mixing to simulate the mixture where the target source is inactive. The loss function was the negative SNR loss which accepts zero signals as ground truth [34]. During inference, each song was segmented into multiple T'-second chunks with half overlap, separated individually, and reconstructed using overlap-add to obtain song-level separation results. We use the utterance-level SDR (uSDR) as the evaluation metric [35]. All other configurations remained consistent with those used for the WHAMR! setup.

D. Results on WHAMR!

Table I presents the average SI-SDR improvement (SI-SDRi) on the WHAMR! test set. Both `A⋆` and `B⋆` used the 8kHz min version of training data; however, `A⋆` models were trained with one-second input segments, whereas `B⋆` models were trained with four-second input segments.

TABLE I

AVERAGE SI-SDRI ON WHAMR! TEST SET. 'MIN' VERSION IS USED FOR TRAINING, WHILE EVALUATION IS DONE ON BOTH 'MIN' AND 'MAX' VERSIONS, EACH WITH 8KHZ AND 16KHZ DATA.

ID	Train SF	Input len[s]	PE	8kHz min	8kHz max	16kHz min	16kHz max
A1	8kHz	1	APE	15.9	16.4	6.4	5.4
A2	8kHz	1	KERPLE	16.1	16.7	15.3	15.9
A3	8kHz	1	RoPE	15.8	16.2	14.9	15.0
A4	8kHz	1	NoPE	16.1	16.7	15.4	16.0
B1	8kHz	4	APE	17.6	18.4	14.8	14.8
B2	8kHz	4	KERPLE	17.7	18.4	17.0	17.7
B3	8kHz	4	RoPE	**17.8**	18.5	16.9	17.6
B4	8kHz	4	NoPE	**17.8**	**18.6**	17.3	18.1
C1	16kHz	4	RoPE	17.4	18.2	17.6	18.5
C2	16kHz	4	NoPE	17.3	17.9	**17.7**	**18.7**

TABLE II

AVERAGE SI-SDRI WHEN $K = 1$ ON WHAMR! TEST SET. TRAINING IS DONE WITH 8KHZ 'MIN' VERSION, WHILE TESTING IS DONE USING BOTH 'MIN' AND 'MAX' VERSIONS. TRAINING CHUNK WAS 4-SECONDS.

ID	PE	8kHz min	8kHz max	16kHz min	16kHz max
D1	APE	13.4	12.6	4.3	1.5
D2	KERPLE	13.8	13.7	9.9	9.8
D3	RoPE	**13.9**	**14.4**	6.6	6.7
D4	NoPE	13.5	**14.4**	**11.2**	**12.0**

As shown in Table I, the NoPE approach achieves performance comparable to or better than other explicit PE methods under matched conditions (8k min). In contrast, Table II shows the results for models without convolution, where the convolutional kernel size K is set to one, demonstrating that some explicit PEs outperform NoPE on 8k min data. These results suggest that in models incorporating convolutional layers, such as TF-Locoformer, convolution implicitly encodes position information, allowing explicit PE to be omitted without compromising overall performance. Notably, in our experimental setup, training B4 (NoPE) was approximately 16% faster than training B3 (RoPE). To further assess whether NoPE maintains its effectiveness in larger models, we evaluated a medium-sized TF-Locoformer, with results summarized in Table III. The NoPE-based model outperformed the RoPE-based model and other state-of-the-art models, demonstrating its scalability.

Next, we analyzed the length extrapolation capability of time-frame modeling by focusing on the results of A⋆. A comparison between the 'min' and 'max' data reveals that KERPLE and NoPE achieve the best extrapolation performance. The strong extrapolation ability of KERPLE aligns with findings from previous studies [22]; however, our results also confirm that NoPE, when used in conjunction with convolutional layers, is equally effective.

We then examine the length extrapolation capability in frequency modeling by comparing results on the 8kHz and 16kHz test sets. Similarly, NoPE and KERPLE exhibit the best performance, with NoPE achieving slightly superior results. Compared to RoPE (B3), which was originally used in TF-Locoformer, NoPE (B4) performs comparably on the 8kHz min dataset while improving SI-SDR by approximately 0.4 dB on the 16kHz min dataset. These findings suggest that convolution-encoded positional information also enhances frequency extrapolation.

TABLE III

COMPARISON WITH PREVIOUS MODELS ON WHAMR!. DM* INDICATES DM WITH SPEED PERTURBATION. RESULTS IN [DB]. † DENOTES THAT THE RESULT IS REPRODUCED BY US.

System	SI-SDRi	SDRi
SepFormer+DM* [8]	14.0	13.0
MossFormer2+DM* [36]	17.0	-
TF-GridNet [6]	17.1	15.6
TF-Locoformer (S) [12]	17.4	15.9
TF-Locoformer (M) [12]	18.5	16.9
TF-Locoformer (S)†	17.8	16.1
TF-Locoformer-NoPE (S)	17.8	16.1
TF-Locoformer-NoPE (M)	**18.8**	**17.1**

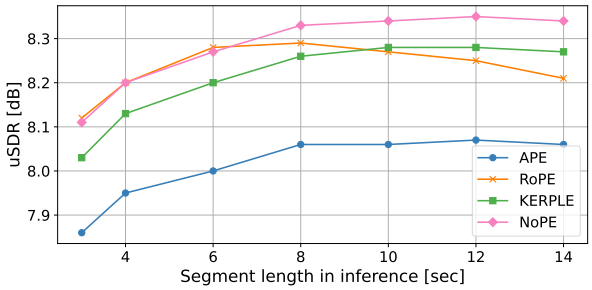

Fig. 1. uSDR scores of models trained on 3-second chunks when changing segment length in inference. Hop size was set to half of segment length.

Although NoPE demonstrates the best length extrapolation capability, an interesting trend emerges in Table I: for input sequences shorter than those used during training, explicit PE methods outperform NoPE. This effect is evident in the performance of C1 and C2, which were trained on 16 kHz data and evaluated on 8 kHz data, where C1 achieves superior results. Based on these findings, we conclude that NoPE is preferable for handling sequences longer than those seen during training, whereas explicit PE methods, such as RoPE, are more suitable for shorter sequences.

E. Results on MUSDB

Figure 1 presents the uSDR scores of BS-Locoformer trained with three-second segments. The horizontal axis represents the segment length T' used during inference, which was set to 3, 4, 6, 8, 10, 12, and 14 seconds for evaluation. As shown in the figure, when $T' = 3$, matching the training condition, NoPE and RoPE achieve the best performance. As T' increases, performance improves; however, while RoPE saturates at $T' = 8$, NoPE continues to improve beyond $T' > 8$. Although KERPLE exhibits lower overall performance, it follows a similar trend to NoPE, further demonstrating the strong length extrapolation capabilities of NoPE and KERPLE.

Figure 2 presents the evaluation results for models trained with segment lengths of 3, 6, or 8 seconds, using either RoPE or NoPE. Regardless of the PE method, models trained with longer input segments tend to achieve better performance. However, even under these conditions, NoPE exhibits superior length extrapolation compared to RoPE. In contrast, when T' is shorter than the training segment length, RoPE outper-

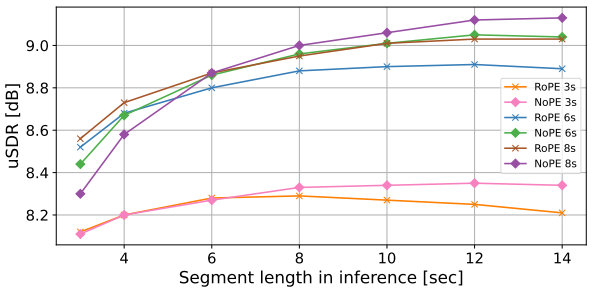

Fig. 2. uSDR scores of models trained on 3-, 6-, or 8-second chunks. uSDRs are evaluated changing segment length in inference, where hop size was set to half of segment length.

forms NoPE. This observation is consistent with findings from WHAMR! experiments, further suggesting that explicit PE methods such as RoPE are preferable for handling sequences shorter than those seen during training.

V. Conclusion

In this work, we investigated the impact of positional encoding (PE) on length extrapolation in Transformer-based TF-domain dual-path models. To this end, we compared four PE methods, APE, KERPLE, RoPE, and NoPE, in a framework of the TF-Locoformer, a recent state-of-the-art model. Through experiments on both speech separation and MSS, we observed the following trends: (i) When handling sequences that are the same length as or shorter than those seen during training, models with explicit PEs tend to perform better. (ii) However, models without PE exhibit superior length extrapolation, a trend that is particularly pronounced when the model includes convolutional layers.

VI. Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP24KJ2096.

References

[1] J. R. Hershey, Z. Chen, J. Le Roux *et al.*, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. ICASSP*, 2016.

[2] D. Yu, M. Kolbæk, Z. H. Tan *et al.*, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. ICASSP*, 2017.

[3] Y. Luo and N. Mesgarani, "TasNet: Time-domain audio separation network for real-time, single-channel speech separation," in *Proc. ICASSP*, 2018.

[4] ——, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.

[5] L. Yang, W. Liu, and W. Wang, "TFPSNet: Time-frequency domain path scanning network for speech separation," in *Proc. ICASSP*, 2022.

[6] Z.-Q. Wang, S. Cornell, S. Choi *et al.*, "TF-GridNet: Integrating full- and sub-band modeling for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 3221–3236, 2023.

[7] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-Path RNN: Efficient long sequence modeling for time-domain single-channel speech separation," in *Proc. ICASSP*, 2020.

[8] C. Subakan, M. Ravanelli, S. Cornell *et al.*, "Attention is all you need in speech separation," in *Proc. ICASSP*, 2021.

[9] X. Jiang, C. Han, and N. Mesgarani, "Dual-path mamba: Short and long-term bidirectional selective structured state space models for speech separation," in *Proc. ICASSP*, 2025, pp. 1–5.

[10] W. Zhang, K. Saijo, Z.-Q. Wang *et al.*, "Toward universal speech enhancement for diverse input conditions," in *Proc. ASRU*, 2023.

[11] K. Saijo, J. Ebbers, F. G. Germain *et al.*, "Task-aware unified source separation," in *Proc. ICASSP*, 2025, pp. 1–5.

[12] K. Saijo, G. Wichern, F. G. Germain *et al.*, "Tf-locoformer: Transformer with local modeling by convolution for speech separation and enhancement," in *Proc. IWAENC*, 2024, pp. 205–209.

[13] J. Su, Y. Lu, S. Pan *et al.*, "RoFormer: Enhanced transformer with rotary position embedding," *arXiv preprint arXiv:2104.09864*, 2021.

[14] O. Press, N. A. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," *arXiv preprint arXiv:2108.12409*, 2021.

[15] T.-C. Chi, T.-H. Fan, P. J. Ramadge *et al.*, "Kerple: Kernelized relative positional embedding for length extrapolation," *Proc. NeurIPS*, vol. 35, pp. 8386–8399, 2022.

[16] M. A. Islam, S. Jia, and N. D. Bruce, "How much position information do convolutional neural networks encode?" *arXiv preprint arXiv:2001.08248*, 2020.

[17] X. Chu, Z. Tian, B. Zhang *et al.*, "Conditional positional encodings for vision transformers," *arXiv preprint arXiv:2102.10882*, 2021.

[18] A. Gulati, J. Qin, C.-C. Chiu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[19] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," *Proc. NeurIPS*, 2017.

[20] B. Zhang and R. Sennrich, "Root mean square layer normalization," *Proc. NeurIPS*, 2019.

[21] N. Shazeer, "GLU variants improve Transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[22] Q. Zhang, H. Zhu, X. Qian *et al.*, "An exploration of length generalization in transformer-based speech enhancement," in *Proc. Interspeech*, 2024, pp. 1725–1729.

[23] H. Touvron, L. Martin, K. Stone *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[24] A. Kazemnejad, I. Padhi, K. Natesan Ramamurthy *et al.*, "The impact of positional encoding on length generalization in transformers," *Proc. NeurIPS*, vol. 36, pp. 24 892–24 928, 2023.

[25] W. Wang, E. Xie, X. Li *et al.*, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational visual media*, vol. 8, no. 3, pp. 415–424, 2022.

[26] M. Maciejewski, G. Wichern, E. McQuinn *et al.*, "WHAMR!: Noisy and reverberant single-channel speech separation," in *Proc. ICASSP*, 2020.

[27] J. S. Garofolo *et al.*, *CSR-I (WSJ0) Complete LDC93S6A*, Linguistic Data Consortium, Philadelphia, 1993, web Download.

[28] G. Wichern, J. Antognini, M. Flynn *et al.*, "WHAM!: Extending speech separation to noisy environments," in *Proc. Interspeech*, 2019.

[29] Y. Luo and J. Yu, "Music source separation with band-split rnn," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1893–1901, 2023.

[30] W.-T. Lu, J.-C. Wang, Q. Kong *et al.*, "Music source separation with band-split rope transformer," in *Proc. ICASSP*, 2024, pp. 481–485.

[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2018.

[32] J. Le Roux, S. Wisdom, H. Erdogan *et al.*, "SDR — half-baked or well done?" in *Proc. ICASSP*, 2019.

[33] T. Dao, D. Fu, S. Ermon *et al.*, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *Proc. NeurIPS*, vol. 35, pp. 16 344–16 359, 2022.

[34] S. Wisdom, H. Erdogan, D. P. Ellis *et al.*, "What's all the fuss about free universal sound separation data?" in *Proc. ICASSP*, 2021, pp. 186–190.

[35] Y. Mitsufuji, G. Fabbro, S. Uhlich *et al.*, "Music demixing challenge 2021," *Frontiers in Signal Processing*, vol. 1, p. 808395, 2022.

[36] S. Zhao, Y. Ma, C. Ni *et al.*, "MossFormer2: Combining transformer and RNN-free recurrent network for enhanced time-domain monaural speech separation," in *Proc. ICASSP*, 2024.