INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Lightweight Causal Transformer with Local Self-Attention for Real-Time Speech Enhancement

Koen Oostermeijer, Qing Wang, Jun Du[*]

University of Science and Technology of China, Hefei, China

keen@mail.ustc.edu.cn, qingwang2@ustc.edu.cn, jundu@ustc.edu.cn

Abstract

In this paper, we describe a novel speech enhancement transformer architecture. The model uses local causal self-attention, which makes it lightweight and therefore particularly well-suited for real-time speech enhancement in computation resource-limited environments. In addition, we provide several ablation studies that focus on different parts of the model and the loss function to figure out which modifications yield best improvements. Using this knowledge, we propose a final version of our architecture, that we sent in to the INTERSPEECH 2021 DNS Challenge, where it achieved competitive results, despite using only 2% of the maximally allowed computation. Furthermore, we performed experiments to compare it with with LSTM and CNN models, that had 127% and 257% more parameters, respectively. Despite this difference in model size, we achieved significant improvements on the considered speech quality and intelligibility measures.

Index Terms: speech enhancement, transformer, causal

1. Introduction

In recent years, transformers have seen increasing use in many fields of machine learning. Initially introduced for the purpose of natural language processing (NLP) [1], they have now been employed in many other areas such as image processing and speech recognition [2, 3]. However, applications in speech enhancement have been limited so far.

Speech enhancement has had a long history, starting from classical algorithms, such as spectral subtraction [4], minimum-mean square error (MMSE) based spectral amplitude estimators [5], Wiener filtering [6], Karhunen-Loéve transformations (KLT) [7] and non-negative matrix factorization (NMF) [8] to modern deep learning-based methods, which can be roughly subdivided into fully connected neural networks (FNNs) [9, 10], recurrent neural networks (RNNs) [11, 12] and convolutional neural networks (CNNs) [13, 14, 15].

Previous applications of transformers in speech enhancement include T-GSA [16], which uses Gaussian weighted self-attention and MHANet [17], a causal architecture that is trained using the deep xi learning approach [18]. Other approaches have merged transformers with other types of neural networks, two examples of these are [19], in which the authors combine multi-head self-attention with bidirectional long short-term memory (BLSTM) enhanced with speaker-aware features, and Self-Attention Speech Enhancement Generative Adverserial Network (SASEGAN) [20], which seeks to improve SEGAN [15] with multi-head self-attention.

Building on these previous works, we introduce a novel light-weight causal transformer model with local self-attention for real-time speech enhancement and apply it to the 2021 INTERSPEECH DNS Challenge [21]. Previous transformer-based speech enhancement models have either not been causal and are therefore not suited to the task of real-time speech enhancement, and/or used global attention, resulting in an $O(T^2)$ computational complexity for a sequence length of T, which also prohibits them from being used efficiently for real-time speech enhancement. Besides this improvement, we perform a number of ablation studies to determine which changes provide the largest improvements, divided into four categories, each focusing on a different part of the transformer architecture or loss function.

The structure of this paper is as follows: In the next section, the transformer architecture will be introduced in detail. Then in section 3, we will explain the local attention mechanism that is used for our model. Section 4 describes the experimental setup of the ablation studies, the results of which are described in section 5. Finally, section 6 is devoted to the ablation studies

To summarize, our contributions are as follows:

- We introduce a novel speech enhancement transformer with local self-attention. The model is light-weight and causal, making it ideal for real-time speech enhancement in low-resource environments.
- We perform a comparative study of different architectures to find the optimal one.
- We apply our method to the 2021 INTERSPEECH DNS Challenge.

2. Transformers

The defining feature of transformers is their multi-head self-attention modules (MHA) [1].

Given an input $X \in \mathbb{R}^{T \times n}$, where T is the number of time steps and n is the hidden state dimension, a set of query, key and value matrices is generated using the weight matrices W_h^Q, W_h^K and $W_h^V \in \mathbb{R}^{n \times d_k}$, respectively, where d_k is the dimension of the heads of the attention module. There is one embedding per head, denoted by the subscript h.

$$Q_h = XW_h^Q, \tag{1}$$

$$K_h = XW_h^K, \tag{2}$$

$$V_h = XW_h^V. \tag{3}$$

The keys and queries are multiplied with each other to obtain a $T \times T$ attention matrix A. This matrix encodes the relative importance of each time step, i.e. how much attention each time step receives, by assigning each pair of time steps a scalar. A softmax function with temperature $\sqrt{d_k}$ is applied to turn this into a normalized distribution. Afterwards, the normalized attention matrix is multiplied with the value matrix. This results in a linear combination of value embeddings for each time step, where the most important embeddings receive the highest

[*] corresponding author

http://dx.doi.org/10.21437/Interspeech.2021-668

weights:

$$\mathrm{Att}_h = \mathrm{Softmax}\left(\frac{Q_h K_h^\top}{\sqrt{d_k}}\right) V_h. \tag{4}$$

The heads are then concatenated and transformed back to the original dimension n via the weight matrix $W^{\text{out}} \in \mathbb{R}^{d_k \cdot n_h \times n}$, where n_h is the number of heads. Moreover, a residual connection is added, that connects the output to the input:

$$X^{\text{out}} = \mathrm{Concat}_h(\mathrm{Att}_h) W^{\text{out}} + X. \tag{5}$$

Next, each of the time steps is standardized via layer normalization. For time step t, the overall mean of the feature dimension is subtracted from the input, which is then divided by the standard deviation. This is rescaled and shifted by the learnable parameters α and β:

$$X_t^{\text{norm}} = \frac{X_t^{\text{out}} - \mu_t}{\sigma_t} \cdot \alpha + \beta, \tag{6}$$

where

$$\mu_t = \frac{1}{n} \sum_i X_{ti}^{\text{out}}, \tag{7}$$

$$\sigma_t = \sqrt{\frac{1}{n} \sum_i (X_{ti}^{\text{out}} - \mu_t)^2}. \tag{8}$$

Then, a feedforward neural network is applied time stepwise. This part typically consists of two fully connected layers parameterized by the weight matrices $W_1 \in \mathbb{R}^{n \times \phi n}$, $W_2 \in \mathbb{R}^{\phi n \times n}$ and bias vectors $b_1 \in \mathbb{R}^{\phi n}$, $b_2 \in \mathbb{R}^n$ and a residual connection:

$$f(X_t^{\text{norm}} W_1 + b_1) W_2 + b_2 + X_t^{\text{norm}}, \tag{9}$$

where $f(\cdot)$ is an element-wise activation function, such as Rectified Linear Unit (ReLU) or Gaussian Error Linear Unit (GELU). Here, ϕ is a scaling factor for the inner dimension of the feedforward module. Finally, another layer normalization is applied.

3. Local Self-Attention

Transformers scale quadratically in the input length, this is prohibitive when scaling to longer sequences. We resolve this issue by introducing a novel speech enhancement transformer model that relies on local attention [22, 23]. Local attention is particularly suited for speech enhancement, since predictions do not rely on long-range correlations as is the case for NLP; often sufficient information is contained within a couple of seconds of the target time frame. This requirement is naturally incorporated with local attention.

Instead of taking inner products between all keys and queries, we restrict the context size to only a window of length W, which spans from the time step $t - W + 1$ to the current time step t.This also enforces causality, as the frame at time step t only has access to frames that lie in its past. However, it is possible to extend the context size to several future time steps at the cost of more latency. Formally, we can write that the keys matrix is restricted as

$$\tilde{K}_{htwi} = \begin{cases} K_{h,t-w,i}, & \text{if } t - w > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{10}$$

where t and i are the time index and feature index, respectively. The index w runs from 0 to $W - 1$. Note that in general, W can depend on both the layer number and the head number. Similarly, the value matrix is also restricted: $\tilde{V}_{htwi}$. The restricted attention matrix is then computed as

$$\tilde{A}_{htw} = \frac{\sum_i Q_{hti} \tilde{K}_{htwi}}{\sqrt{d_k}} \tag{11}$$

and the self-attention is

$$\mathrm{Att}_{htw} = \sum_w \mathrm{Softmax}\left(\tilde{A}_{htw}\right) \tilde{V}_{htwi}. \tag{12}$$

This reduces memory costs of the attention matrix from a quadratic time complexity $O(T^2)$ to a linear one $O(WT)$.

The local context size W can be of the order of 32 frames, corresponding to a second of audio. In speech enhancement, where typical sample lengths range up to hundreds of thousands of tokens, i.e. hours of speech, this yields significant improvements.

4. Experimental Setup

For our experiments, we used the DNS-Challenge dataset[1] [21], containing a total of 760 hours of clean speech and 181 hours of noise, as well as about 118,000 room impulse responses to construct reverberant speech. This speech corpus covers English, French, German, Mandarin Chinese, Russian and Spanish, as well as emotional (English) utterances from speakers of different ethnic backgrounds.

The audio was sampled at 16kHz and subsequently transformed using a short-time Fourier transform with a window size of 512 samples and a step size of 256 samples. This resulted in each time frame having 257 frequency bins, ranging from 0 to 8 kHz. These were converted to log-power spectrum features [24]. Additionally, we appended the average energy at each time step.

As a base model, we used a four-module transformer model, with causal local self-attention with context size 24. The first layer was a causal convolutional layer with kernel size 3 responsible for projecting from the initial $257 + 1$ dimensions to the hidden state dimension $n = 256$. Similarly, the final convolutional layer projected the output of the last transformer module back to 257 dimensions, the number of frequency bins. Further details are given in the next section.

The models were trained using an Adam optimizer [25] with a cosine learning rate decay, going from an initial learning rate of $1e-4$ to a final learning rate of $1e-5$. The speech quality and intelligibility measures that were used to judge the models were Perceptual Evaluation of Speech Quality (PESQ) [26] and [Extended] Short-Time Objective Intelligibility ([E]STOI) [27, 28], respectively.

5. Ablations

In addition to introducing local attention for speech enhancement, we perform a series of ablation studies to find an architecture that makes optimal use of this and further improves performance.

[1] https://github.com/microsoft/DNS-Challenge

5.1. Positional encodings

The self-attention mechanism is invariant under permutations of the inputs. To break this invariance and thereby give the model the ability to differentiate between different time steps, the inputs are imbued with positional encodings. This can be done in several ways. For the original version of the transformer model, the authors used additive sinusoidal positional embeddings [1]. Later versions [29, 23] employed learnable relative positional embeddings. In [30], the authors showed that it is beneficial to only add embeddings to the key and query matrices. In the field of speech enhancement, Kim et al. [16] proposed a method that used multiplicative weights that follow a Gaussian distribution with learnable variance σ_h^2. This corresponds to the intuitive notion that time steps closest by are most relevant to the current time step. The authors also proposed absolute attention, here the absolute value of the inner products between the keys and queries is taken. The idea behind this is that anti-alignment between frames is of equal importance as alignment. In general the methods described above can be written as

$$A'_{hij} = e^{-|i-j|^2/2\sigma_h^2} \left| \tilde{A}_{hij} + P_{ij} \right|, \tag{13}$$

where the absolute operation is element-wise and P_{ij} are the learnable relative positional embeddings.

We performed experiments with different combinations of relative positional embeddings, Gaussian weights and absolute attention. The results of which are shown in Table 1.

Table 1: *Results ablation study positional embeddings*

Pos.	Gauss.	Abs.	ESTOI(%)	PESQ	STOI(%)
Yes	Yes	Yes	**79.6**	**3.27**	**89.0**
No	Yes	Yes	79.4	3.25	88.9
No	Yes	No	79.2	3.26	88.7
Yes	No	Yes	79.5	**3.27**	88.7
Yes	No	No	79.4	**3.27**	88.1
No	No	No	79.1	3.25	88.5

The common denominator for the best performing models on PESQ is that they all use relative positional embeddings. Comparing the second and the third, and the fourth and the fifth model, respectively, it can be seen that absolute attention gave better ESTOI and STOI values. A similar conclusion can be drawn for Gaussian weights, when comparing the first with the fourth model and third with the sixth. Overall, the combination of relative positional embeddings with Gaussian weighting and absolute attention provided the best results.

5.2. Feedforward layers

The role of the feedforward layer in the transformer model is to perform local operations, whereas the self-attention layers are responsible for non-local operations. We opted to make the feed-forward layers also consider nearby past time steps, since relevant information is localized here, similar to Conformer [3]. This has the added benefit of increasing overall context size, allowing information to propagate from time steps further back in time. For example, in our model each layer uses a kernel width of three, combined with an average context window size of 24, this results in a total of about 3.1 seconds of context size for a four-module network.

We generalized the feedforward layers further by enhancing them with gating like the one used for Generalized Linear Units (GLU) [31]. These add an extra linear layer to dynamically generate weights to rescale the connections of the first feedforward layer.

$$X_{l+1} = ((X_l W_1 + b_1) \circ f(X_l W_2 + b_2))\, W_3 + b_3, \tag{14}$$

where $f(\cdot)$ is an activation function such as Sigmoid, ReLU, GELU or Swish. When $W_1 = 0$ and $b_1 = 1$, this reduces to the standard feed-forward layer. The product between the input X_l and the weight matrix W_2 is understood as a convolution with kernel width 3. The other convolutions are standard matrix multiplications, i.e. they correspond to kernels of width 1. This is done to reduce the computational demand of these layers. As before the inner dimension is scaled up by a factor of ϕ. Moreover, we considered variations in which the convolutions W_1 and W_2 are depthwise separable. It is important that at least one of the layers is fully connected to allow information between the channels of the feature dimension to travel. We defined a grouping factor γ , where $\gamma = 0$ and $\gamma = 1$ correspond to ungrouped and fully grouped convolutions respectively.

The final modification we considered is the recent Macaron [32], which has shown to significantly improve performance in NLP. It works by adding an extra feedforward layers before the self-attention mechanism and rescaling them by a factor of $1/2$.

Therefore, we considered the following five models, where ϕ was chosen such that all the models had an approximately equal number of parameters:

1. ReLU activation, $\gamma = 0$, $\phi = 1$
2. GELU activation, $\gamma = 0$, $\phi = 1$
3. ReLU activation, $\gamma = 1$, $\phi = 4$
4. ReLU activation, $\gamma = 1$, $\phi = 2$, GLU
5. ReLU activation, $\gamma = 0$, $\phi = 0.5$, Macaron

The results of our experiments are shown in Table 2.

Table 2: *Results ablation study feedforward layers*

Model	ESTOI(%)	PESQ	STOI(%)
1	80.1	3.31	**89.4**
2	**80.3**	**3.33**	88.9
3	80.0	3.30	89.2
4	79.4	3.26	88.8
5	NaN	NaN	NaN

It can be seen that the GELU activation function performed better than ReLU, as it provided better results on two of the three metrics. This is in line with findings from NLP. Comparing the separated model (3) and the GLU model (4) with the baseline (1), we see that neither yielded improvements on any of the speech quality and intelligibility metrics. Finally, despite tweaking learning rates and batch sizes, the Macaron model (5) proved to be unstable.

5.3. Window sizes

In [33], the authors found that for NLP it is beneficial to start with short context sizes in the lower layers and increase them toward the higher layers. We replicated this experiment in the context of speech enhancement. We considered three different variants: increasing context length: from size 12 to 36 in steps of 8, constant context length: all context size equal to 24 and decreasing context length: 36 to 12 in steps of 8. The results are shown in Table 3 below.

Table 3: *Results ablation study context sizes*

Model	ESTOI(%)	PESQ	STOI(%)
Ascending	**80.1**	**3.31**	**89.5**
Equal	**80.1**	**3.31**	89.4
Descending	79.9	3.29	88.6

We found that unlike in NLP, there was no significant performance improvement when using ascending context sizes. However, we did observe that descending context sizes resulted in reduced performance.

5.4. Loss functions

As a base loss function we used a mean squared error (MSE) loss function based on log-power spectrum (LPS) features [24]. Since this loss function does not capture perceptually relevant effects such as correlations, differences in loudness and threshold effects, we enhanced it by adding an auxiliary Perceptual Metric for Speech Quality Evaluation (PMSQE) [34] loss function, based on PESQ. We compared this with the E^{2}STOI loss function [35], which is an ESTOI-based loss function. For both we used a scaling factor of 0.1. The results are shown below in Table 4.

Table 4: *Results ablation study loss functions*

Model	ESTOI(%)	PESQ	STOI(%)	SDR
PMSQE	80.1	**3.31**	89.5	10.63
E^{2}STOI	**83.0**	3.01	**92.0**	**13.87**

The PMSQE-enhanced loss function provided better results for the PESQ measure and the E^{2}STOI-enhanced loss function gave the best results for STOI and ESTOI. This is likely due to the fact that PMSQE is based on PESQ and therefore optimizes better for this. On the other hand, E^{2}STOI for ESTOI, which is closely related to STOI, was better able to improve STOI and ESTOI. To give a more complete image, we also reported the signal-to-distortion ratio (SDR) [36]. When taking only this into account, the E^{2}STOI-enhanced loss function performs best.

6. Full Model

The finding of our ablation studies were used to guide us in constructing the final model, that was sent into the INTERSPEECH 2021 DNS Challenge. Based on the first ablation study, we chose to use positional embeddings, Gaussian weighting and absolute attention. From the experiments of the feedforward layers, we used GELU activation functions, unseparated convolutions and an expansion factor of $\phi = 1$. Window sizes were kept constant throughout the layers at $W = 16$. Furthermore, we combined the PMSQE and E^{2}STOI loss functions to optimize for both PESQ and STOI simultaneously.

6.1. Computational complexity

The number of multiply-add (MAdd) operations [37] for a feedforward layer as used in our model with d_k input channels and kernel size 3 applied to an input of length T is $T(3n+1)n + T(n+1)n = T(4n+2)$. The the number of Madd operations of the layer normalization is $T \cdot n$ and that of the key, query and value projections is $T \cdot n_h \cdot d_k \cdot n$, respectively. As mentioned before, the use of local attention gives us a complexity for the attention mechanism of $T \cdot n_h \cdot W \cdot d_k$ Madd operations instead of for full $T^2 \cdot n_h \cdot d_k$. Softmax, multiplication with the value matrix and the transformation combining the heads have a complexity of $T \cdot W \cdot n_h$, $T \cdot n_h \cdot d_k \cdot W$ and $T \cdot n_h \cdot d_k \cdot n$, respectively. Finally, the input and output layers add $T \cdot (3(n_f + 1) + 1) \cdot n$ and $T \cdot (n_f + 1) \cdot n$ Madd operations, respectively, where n_f is the number of frequency bins. For a model of n_l layers, this leads to a total complexity of

$$T(n_l(6n + 2 + 4n_h d_k n + 2W n_h (d_k + 1)) + n(4n_f + 3))$$

To simplify this further, we set $n = n_h d_k$. With this, the complexity can be approximated as

$$Tn(n_l(6 + 4n + 2W) + 4n_f))$$

The final model used a head dimension $d_k = 48$, and the number of heads $n_h = 8$. The inner dimension $n = 384$. With these settings, we found the total number of multiply-add operations to be approximately $3.4 \cdot 10^6$. The total number of parameters was $6.2 \cdot 10^6$. On an Intel I5 quad core clocked at 1.6 GHz the computational time per second was about 19 ms, a mere 1.9% of the maximum allowed computation time.

6.2. Comparison models

We compared our model with similar LSTM and CNN architectures. The LSTM model was comprised of three LSTM layers, each with 1024 hidden units, for a total of $2.2 \cdot 10^7$ parameters. The CNN model used four convolutional layers with causal kernels of width 3, batch normalization and ReLU activation functions. The number of channels was 1024. This model had $1.4 \cdot 10^7$ parameters. Therefore, the proposed model had only 28% and 44% the number of parameters of the LSTM and CNN, respectively.

6.3. Results

In Table 5 below, the results of our final model and the comparison models are listed.

Table 5: *Results final model*

Model	ESTOI(%)	PESQ	STOI(%)	Params.
Noisy	78.0	2.51	89.0	N/A
Proposed	**83.2**	**3.35**	**92.3**	$6.2 \cdot 10^6$
LSTM	82.4	3.16	92.0	$2.2 \cdot 10^7$
CNN	81.5	3.09	91.3	$1.4 \cdot 10^7$

Our model outperformed the LSTM and CNN models on all speech quality and intelligibility measures for a fraction of the model complexity. In particular, on ESTOI it yielded a value that was 0.8 percentage points and 1.7 percentage points above the LSTM and CNN models, respectively. Similarly, its PESQ score was 0.19 higher compared to the LSTM model and 0.26 compared to the CNN. And for STOI it improved the score by 0.3 percentage points and 1.0 percentage points.

7. Acknowledgements

This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDC08050200, as well as Huawei Noah's Ark Lab and the China Scholarship Council.

8. References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020.

[3] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[4] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[6] P. Scalart *et al.*, "Speech enhancement based on a priori signal to noise estimation," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2. IEEE, 1996, pp. 629–632.

[7] Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 4, pp. 334–341, 2003.

[8] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4029–4032.

[9] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 553–554.

[10] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, pp. 65–68, 2013.

[11] S. Parveen and P. Green, "Speech enhancement with missing data techniques using recurrent neural networks," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2004, pp. I–733.

[12] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 91–99.

[13] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," *arXiv preprint arXiv:1609.07132*, 2016.

[14] T. Kounovsky and J. Malek, "Single channel speech enhancement using convolutional neural network," in *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*. IEEE, 2017, pp. 1–5.

[15] S. Pascual, A. Bonafonte, and J. Serra, "Segan: Speech enhancement generative adversarial network," *arXiv preprint arXiv:1703.09452*, 2017.

[16] J. Kim, M. El-Khamy, and J. Lee, "T-gsa: Transformer with gaussian-weighted self-attention for speech enhancement," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6649–6653.

[17] A. Nicolson and K. K. Paliwal, "Masked multi-head self-attention for causal speech enhancement," *Speech Communication*, vol. 125, pp. 80–96, 2020.

[18] ——, "Deep learning for minimum mean-square error approaches to speech enhancement," *Speech Communication*, vol. 111, pp. 44–55, 2019.

[19] Y. Koizumi, K. Yaiabe, M. Delcroix, Y. Maxuxama, and D. Takeuchi, "Speech enhancement using self-adaptation and multi-head self-attention," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 181–185.

[20] H. Phan, H. L. Nguyen, O. Y. Chén, P. Koch, N. Q. Duong, I. McLoughlin, and A. Mertins, "Self-attention generative adversarial network for speech enhancement," *arXiv preprint arXiv:2010.09132*, 2020.

[21] C. K. Reddy, H. Dubey, K. Koishida, A. Nair, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, "Interspeech 2021 deep noise suppression challenge," *arXiv preprint arXiv:2101.01902*.

[22] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating Wikipedia by summarizing long sequences," *arXiv preprint arXiv:1801.10198*, 2018.

[23] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," 2018.

[24] J. Du and Q. Huo, "A speech enhancement approach using piecewise linear approximation of an explicit model of environmental distortions," in *Ninth annual conference of the international speech communication association*, 2008.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[26] I.-T. Recommendation, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[27] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[28] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2009–2022, 2016.

[29] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:1910.10683*, 2019.

[31] N. Shazeer, "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[32] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu, "Understanding and improving transformer from a multi-particle dynamic system point of view," *arXiv preprint arXiv:1906.02762*, 2019.

[33] J. W. Rae and A. Razavi, "Do transformers need deep long-range memory," *arXiv preprint arXiv:2007.03356*, 2020.

[34] J. M. Martin-Donas, A. M. Gomez, J. A. Gonzalez, and A. M. Peinado, "A deep learning loss function based on the perceptual evaluation of the speech quality," *IEEE Signal processing letters*, vol. 25, no. 11, pp. 1680–1684, 2018.

[35] K. Oostermeijer, Q. Wang, and J. Du, "Frequency gating: Improved convolutional neural networks for speech enhancement in the time-frequency domain," *Annual Summit and Conference 2020, Asia-Pacific Signal and Information Processing Association*, 2020.

[36] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.