TRANSFORMER-BASED VIRTUAL MICROPHONE ESTIMATOR

Zheng Qiu[1]*, Jiachen Wang*[2]*, Bo He*[1]*, Shiqi Zhang*[1]*, Shoji Makino*[1]

[1] Waseda University, Japan and [2]Nagoya University, Japan

ABSTRACT

The performance of microphone array signal processing depends on the number of microphones used. Generally, higher microphone counts correlate with improved performance. However, there are often some limitations on the number of microphones used in the practical scenery. In such cases, we can generate virtual microphone signals to mimic the real signal so that the number of pieces of equipment used can be reduced. However, previous traditional methods may not always perform optimally in real-world situations, and previous neural network models may not accurately estimate the virtual signal. Our research introduces a new method called the Transformer-based Virtual Microphone Estimator (TVME), which estimates virtual microphone signals in the time domain. TVME employs a fully supervised learning framework that utilizes real observations from virtual microphone locations as targets and is trained using multi-channel observations from recordings. We conducted experiments using the CHiME-4 corpus, which demonstrates that TVME is more accurate than previous NN-VME approaches in estimating virtual microphone signals.

***Index Terms*—** virtual microphone, time-domain network, array signal processing, supervised training

1. INTRODUCTION

Microphone array signal processing [1] is a technique that captures and analyzes acoustic signals using multiple microphones. Including blind source separation (BSS) [2, 3], speech enhancement [4, 5], and direction-of-arrival (DOA) estimation [6, 7]. Using more microphones results in better performance, but practical limitations often prevent the integration of many microphones. Researchers are developing techniques to synthesize virtual microphone channels from a limited set of microphones to enhance overall performance in audio applications.

Within the traditional approaches [8, 9] , certain researchers have based their methods [8] on underlying physical models, estimating virtual signals through complex interpolation techniques within the domain of complex logarithmic domains. Different phase and amplitude estimates have also been performed independently [9], and these estimates have subsequently been combined to utilize methods rooted in β-scatter estimation. This composite approach improves the processing performance of microphone arrays, especially in the presence of inherent uncertainties.

In light of the rising prominence of Deep Neural Networks (DNNs) within the domain of virtual microphone research [10, 11], certain investigators have introduced a time-domain model named NN-VME [11], which shares structural similarities with ConvTasNet [12]. However, a persistent challenge remains. The current model is facing limitations in generating realistic audio signals due to the traditional neural network's inability to process continuous data effectively. It is an area that requires further exploration and improvement.

To tackle this challenge, in this paper, we introduce an innovative methodology for virtual microphone signal estimation within a supervised learning framework. This method leverages the Transformer [13], a contemporary and widely adopted novel neural network architecture. Notably, Transformer does not explicitly rely on assumptions tied to physical modeling and has a remarkable potential for processing sequential data. In recent years, Transformer-based neural network models [14, 15, 16] have demonstrated exceptional performance in audio classification tasks. Our study takes inspiration from recent advancements to explore the use of the Transformer framework in the field of virtual microphones. We aim to investigate the potential for estimating virtual microphone signals by establishing mapping relationships between signals derived from observed data captured using real microphone arrays. We will be using transformer-based models for this new approach, which is called the Transformer-Based Virtual Microphone Estimator (TVME).

In our research, we conducted thorough assessments to compare our proposed approach with NN-VME. To ensure a fair evaluation, we used the same dataset as NN-VME, the CHiME-4 corpus [17], which is known for its realistic, noisy recordings from public environments. Our experimental results confirm that our proposed TVME achieves a commendable performance in virtual microphone estimation. Moreover, we found that incorporating TVME leads to significant improvements in the speech enhancement beamformer's performance, demonstrating the practical usefulness of our approach.

2. RELATED WORKS

2.1. Transformer

Transformer represents an innovative deep learning architecture initially tailored for natural language processing [18, 19], which has subsequently gained broad application across diverse domains. This architectural paradigm introduces a suite of multi-headed self-attention mechanisms that operate concurrently on all components within an input sequence, thereby adeptly extracting and transforming pertinent features.

The cornerstone of the Transformer framework lies in the Multi-head Attention module, which plays a pivotal role in the model's effectiveness. The multi-head attention module takes as input a matrix $\boldsymbol{E} \in \mathbb{R}^{N\times k}$ consisting of multiple embedded vectors $\boldsymbol{e} \in \mathbb{R}^{1\times k}$ and outputs features. Where N represents the total number of vectors and k represents the length of embed vectors. Multi-head Attention involves a three-step process. First, it learns weights, the module implicitly learns weights $\boldsymbol{W_Q}$, $\boldsymbol{W_K}$, and $\boldsymbol{W_V}$, responsible for generating query matrix $\boldsymbol{Q}$, key matrix $\boldsymbol{K}$, and value matrix $\boldsymbol{V}$ for each input position. Secondly, the procedure begins with computing the dot product between the query matrix $\boldsymbol{Q}$ and the key matrix $\boldsymbol{K}$. This result is then scaled by a normalization factor, represented as the reciprocal of the square root of the vector's dimension d. A softmax operation is applied across all inputs, and each value vector is weighted accordingly to yield the output of the attention module. The attention mechanism can be expressed as follows:

$$\text{Attention}\left(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\right)=\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathsf{T}}}{\sqrt{d}}\right)\boldsymbol{V} \quad (1)$$

Finally, to encompass multiple attention heads, each head denoted as h, we concatenate their results to form the final output:

$$\text{MHAttention}\left(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\right)=\text{concat}\left(h_1,...h_n\right)\boldsymbol{W_o} \quad (2)$$

Where MHAttention donated multi-head attention, $\boldsymbol{W_o}$ is the matrix acquired through the learning process and n is the number of heads.

The Transformer architecture has proven to be highly effective in natural language processing tasks. It has also shown great potential in the audio field due to its inherent attributes.

3. PROPOSAL METHOD

The architectural framework of the proposed model, TVME, is illustrated in Figure 1. The network is designed to accommodate two input channels, which correspond to the observed signals from real microphones, and it subsequently produces an output channel representing the estimated virtual microphone signal. In the subsequent sections, we delineate the method employed for forecasting the virtual microphone signal.

3.1. Embedding

First, we need to embed the time domain input as $\boldsymbol{E}$. In TVME, the process is divided into two steps - waveform embedding and position embedding.

3.1.1. Waveform embedding

When a Transformer-based model deals with natural language processing task, the input consists of words and can be easily transformed into vectors using algorithms like Word2Vec [20]. However, when TVME deals with this task, the input is a time-domain signal which requires special processing.

Consider $\boldsymbol{r}_c \in \mathbb{R}^{1\times T}$ as the time domain waveform signal of length T from channel c, which is observed by the microphone array, and let $\hat{c}$ represent the channel need to estimate. Subsequently, the input $\boldsymbol{r}^{\text{in}} = \{\boldsymbol{r}_{c_1}, \boldsymbol{r}_{c_2}\}$. Initially, these two input tensors are concatenated along the channel dimension, i.e., the first dimension, yielding the tensor $\boldsymbol{r}^{\text{in}} \in \mathbb{R}^{2\times T}$. Then $\boldsymbol{r}^{\text{in}}$ is introduced into the waveform encoder. The encoder first divides the input tensor into N segments $\boldsymbol{s}$, then $\boldsymbol{r}^{\text{in}} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, ...\boldsymbol{s}_N\}, \boldsymbol{s} \in \mathbb{R}^{2\times \frac{T}{N}}$; each segment $\boldsymbol{s}$ is then embedded as a vector $\boldsymbol{e}_t^{\text{wav}}$ by using 1D convolution layer such as:

$$\boldsymbol{e}_t^{\text{wav}} = \text{Conv1d}(\boldsymbol{s}_t) \quad (3)$$

The variable t denotes the position of the segment in the input signal sequence. The $\boldsymbol{e}_t^{\text{wav}} \in \mathbb{R}^{k\times 1}$, then transpose the vector to get the final $\boldsymbol{e}_t^{\text{wav}}$.

3.1.2. Position embedding

Furthermore, a dedicated position encoding is devised for each feature segment, with the position encoding equation taking the following form:

$$\boldsymbol{e}_t^{\text{pos}} = \begin{cases} \text{PE}\left(t, 2i\right) = \sin\left(\dfrac{t}{10000^{\frac{2i}{d}}}\right) \\ \text{PE}\left(t, 2i+1\right) = \cos\left(\dfrac{t}{10000^{\frac{2i+1}{d}}}\right) \end{cases} \quad (4)$$

Here, position embedding is denoted by PE, with $\boldsymbol{e}_t^{\text{pos}}$ representing the vector associated with position t. The dimension of the vector is denoted as d, $2i$ denotes even dimension and $2i+1$ denotes odd dimension. Eventually, we will be able to get the embedded vector $\boldsymbol{e}_t$:

$$\boldsymbol{e}_t = \boldsymbol{e}_t^{\text{wav}} + \boldsymbol{e}_t^{\text{pos}} \quad (5)$$

Combining the vectors $\boldsymbol{e}_t$ at different positions t gives the Embedded matrix $\boldsymbol{E} \in \mathbb{R}^{N\times k}$.

3.2. Encoding

To obtain the code matrix $\boldsymbol{C} \in \mathbb{R}^{N\times k}$, the Transformer encoder introduces $\boldsymbol{E}$, as shown in Figure 1. The encoder is

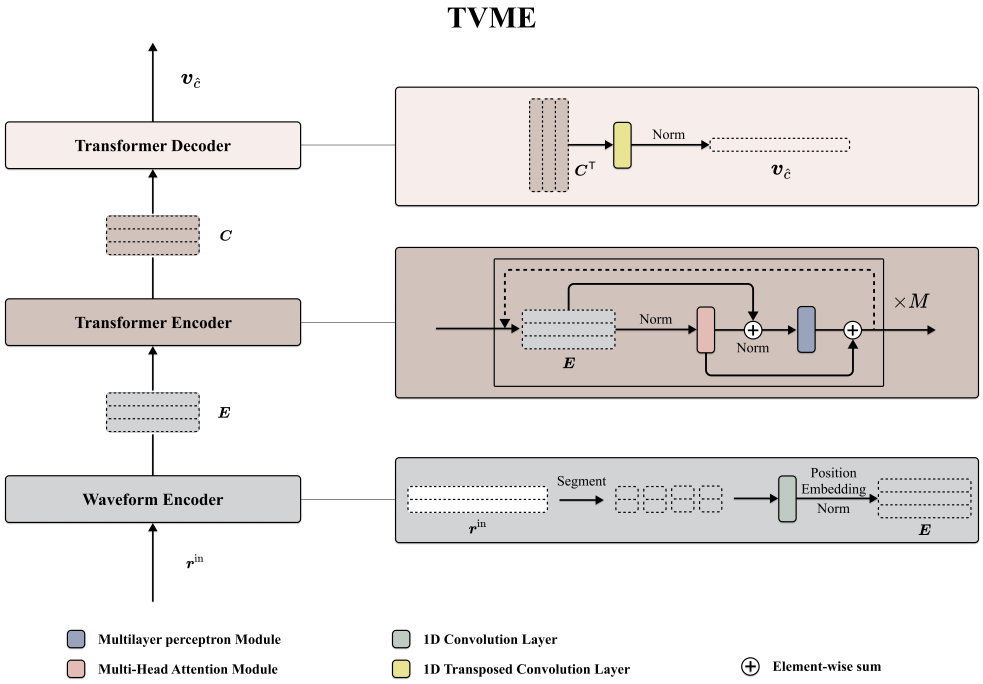

Fig. 1. Structure of transformer-based virtual microphone estimator (TVME)

made up of a stack of M blocks, each block containing a multi-layer perceptron (MLP) module and a multi-head attention module. Before processing, the inputs of each module are layer normalized to ensure stability. Additionally, both inputs and module outputs are aggregated by element-wise summation to form a residual structure.

3.3. Decoding

In order to convert the encoded matrix $\boldsymbol{C}$ into a time domain output, we need to do the embedding inverse operation. Therefore, in the transformer encoder, the input coding matrix $\boldsymbol{C}$ needs to be transposed first and then converted to a single-channel time-domain virtual signal $\boldsymbol{v}_{\hat{c}} \in \mathbb{R}^{1\times T}$ using a one-dimensional transposed convolution as follows:

$$\boldsymbol{v}_{\hat{c}} = \text{ConvTranspose1d}(\boldsymbol{C}^{\mathsf{T}}) \tag{6}$$

Finally, the output is normalized to output a virtual signal.

3.4. Loss function

The proposed TVME utilizes a supervised learning framework to estimate virtual microphone signals. During the training phase, the actual microphone signal, denoted as $\boldsymbol{r}_{\hat{c}}$, is used as the training target at the virtual microphone location. The network is trained using the time-domain loss between the estimated and actual signals.we have chosen signal-to-distortion ratio (SDR) [21] as the loss function. SDR is the ratio of the input signal's power to the difference between the input signal and the reconstructed signal. This loss function can give us a clear understanding of the reconstruction performance. So the SDR is used as the training loss as follows:

$$\textbf{SDR}\left(\boldsymbol{r}_{\hat{c}}, \boldsymbol{v}_{\hat{c}}\right) = 10\log_{10}\left\|\frac{\boldsymbol{r}_{\hat{c}}}{\boldsymbol{v}_{\hat{c}} - \boldsymbol{r}_{\hat{c}}}\right\|^2 \tag{7}$$

4. EXPERIMENT

The proposed TVME will be evaluated through two types of assessments: 1) to evaluate its virtual microphone estimation performance; and 2) to evaluate the enhancement performance of the beamformer using the estimated virtual microphone signals.

4.1. Experiment condition

We have used the CHiME-4 corpus to train and test our TVME. This dataset comprises both simulated and real recordings and includes 3 hours of actual speech from 4 speakers and 15 hours of simulated speech from 83 speakers. We have excluded the microphone signals with low inter-correlation scores in channels 4, 5 and 6, resulting in 1149 discourses for evaluation. During the training process, we take the signals of channels 4 and 6 of the training set as inputs and use the signal of channel 5 as the target to train TVME to generate the signal.

We evaluated the effectiveness of virtual microphone estimation by using SDR as a metric. Our approach involved computing the SDR using actual microphone signals of the corresponding channels and TVME estimated signals. We then compared the results with NN-VME. To examine the impact of the virtual signal on the beamformer's enhancement effect, we created noisy speech by mixing clean speech

Table 1. SDR(higher is better) of the virtual signal that generated by estimator with channel 4 and 6 real signals on the test set

Model	Eval CH	Ref CH	Simu	Real
no process	4	5	12.1	8.8
NN-VME	5	5	16.6	13.8
TVME(proposed)	5	5	**20.4**	**19.1**

from the dataset with the provided noise. Using TVME, we generated the virtual signal from this mixture and used the MVDR (Minimum Variance Distortionless Response) beamformer for enhancement [22]. We also calculated the SDR scores of the clean and enhanced speech to evaluate if the virtual signal could improve the beamformer's performance.

The proposed TVME model was trained on both simulated and real data from the training set. The Adam [23] algorithm was employed during the training process, with an initial learning rate of 0.0001. The training program was stopped after 200 epochs.

For the configuration of the STFT, we utilized a Blackman window with a length of 64 ms and a shift of 16 ms, which same as the NN-VME.

4.2. Experiment result

4.2.1. Evaluation of estimation performance

Table 1 shows the SDR scores of the evaluated models on both simulated and real records in the test set. Here, note that the reference signal for the SDR calculation is the noisy observed signal at the channel corresponding to the virtual microphone, not the clean signal so that the actual microphone estimation performance can be evaluated even for real records.

In the table, the first column ("Eval Channel") shows the channel index of the virtual or real microphone signal used as the estimated signal in the SDR calculation. The second column ("Ref Channel") shows the channel index of the real microphone signal used as the reference signal. As a baseline, we compare the scores with the SDR obtained by NN-VME.

The results in Table 1 demonstrate that the signals estimated by the proposed TVME method achieve significantly higher SDR scores compared to those obtained by NN-VME, for both virtual and real data. These findings suggest that TVME outperforms NN-VME in estimating virtual microphones.

4.2.2. Evaluation of beamformer enhancement performance

Table 2 displays the SDR scores of the beamformer assessed on the simulated data in the test set. In this table, VM BF stands for the beamformer that incorporates the estimated virtual microphone and RM BF refers to the beamformer that utilizes only the real microphone. The columns "real" and "virtual" in "used ch" indicate the channel numbers that correspond to the real and virtual microphones, respectively, which are utilized for developing the beamformer.

Table 2. Speech enhancement performance of MVDR beamformer on the noisy test dataset, which using SDR as metric

Method	Used Channel		SDR
	Real	Virtual	
no process	5	-	3.19
RM BF	4,6	-	6.42
VM BF	4,6	5	**7.01**
RM BF	4,5,6	-	8.49

According to the table, the VM BF achieves higher SDR scores compared to RM BF, which is constructed using the same number of real microphone signals. However, the VM BF has slightly lower SDR scores when compared to the RM BF which uses the same number of microphone signals. This indicates that the virtual microphone signals generated by TVME can enhance the performance of the beamformer to some extent but have limitations compared to real signals.

5. CONCLUSION

This paper presents a novel time-domain virtual microphone estimator, termed TVME. The proposed approach employs the Transformer architecture that utilizes a two-channel real signal as input and produces a single-channel virtual signal as output. TVME is trained using a supervised learning methodology to accurately estimate the virtual microphone signal. The proposed model's efficacy is demonstrated through experimental evaluations that show promising results.

The findings of the experiments conducted by our team indicate that the TVME approach surpasses the NN-VME approach in terms of predicting virtual microphone signals with greater accuracy. The results also suggest that the signals generated through the TVME approach have a positive effect on the beamformer's performance, albeit to a limited extent when compared to the actual signals. These results emphasize the potential of TVME as a dependable method for enhancing the precision of virtual microphone signals, thereby improving the beamformer's performance. Furthermore, the experiments validate the feasibility of the Transformer architecture in the realm of virtual microphone technology, underscoring the potential of the Transformer architecture in the field of virtual microphone technology.

6. REFERENCES

[1] Jacob Benesty, Jingdong Chen, and Yiteng Huang, *Microphone array signal processing*, vol. 1, Springer Science & Business Media, 2008.

[2] Adel Belouchrani, Karim Abed-Meraim, J-F Cardoso, and Eric Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. SP*, vol. 45, no. 2, pp. 434–444, 1997.

[3] Sharon Gannot, Emmanuel Vincent, Shmulik Markovich-Golan, and Alexey Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE Trans. ASLP*, vol. 25, no. 4, pp. 692–730, 2017.

[4] Jitendra D Rayala and Krishna Vemireddy, "Real-time microphone array with robust beamformer and postfilter for speech enhancement and method of operation thereof," Jan. 3 2017, US Patent 9,538,285.

[5] Kristina Tesch and Timo Gerkmann, "Nonlinear spatial filtering in multichannel speech enhancement," *IEEE Trans. ASLP*, vol. 29, pp. 1795–1805, 2021.

[6] Xiaohuan Wu, Wei-Ping Zhu, and Jun Yan, "A toeplitz covariance matrix reconstruction approach for direction-of-arrival estimation," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8223–8237, 2017.

[7] Serkan Tokgöz, Anton Kovalyov, and Issa Panahi, "Real-time estimation of direction of arrival of speech source using three microphones," in *IEEE Workshop SiPS*, 2020, pp. 1–5.

[8] Hiroki Katahira, Nobutaka Ono, Shigeki Miyabe, Takeshi Yamada, and Shoji Makino, "Virtually increasing microphone array elements by interpolation in complex-logarithmic domain," in *Proc. EUSIPCO*, 2013, pp. 1–5.

[9] Hiroki Katahira, Nobutaka Ono, Shigeki Miyabe, Takeshi Yamada, and Shoji Makino, "Nonlinear speech enhancement by virtual increase of channels and maximum snr beamformer," *EURASIP J. Advances Signal Process.*, vol. 2016, no. 1, pp. 1–8, 2016.

[10] Kouei Yamaoka, Li Li, Nobutaka Ono, Shoji Makino, and Takeshi Yamada, "Cnn-based virtual microphone signal estimation for mpdr beamforming in underdetermined situations," in *Proc. EUSIPCO*, 2019, pp. 1–5.

[11] Tsubasa Ochiai, Marc Delcroix, Tomohiro Nakatani, Rintaro Ikeshita, Keisuke Kinoshita, and Shoko Araki, "Neural network-based virtual microphone estimator," in *Proc. ICASSP*, 2021, pp. 6114–6118.

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

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ***L***ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances Neural Inf. Process. Syst.*, vol. 30, 2017.

[14] Yuan Gong, Yu-An Chung, and James Glass, "Ast: Audio spectrogram transformer," in *Proc. Interspeech*, 2021, pp. 571–575.

[15] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *Proc. ICASSP*, 2022, pp. 646–650.

[16] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei, "Beats: Audio pre-training with acoustic tokenizers," in *Proc. PMLR*, 2023, pp. 5178–5193.

[17] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The third 'chime'speech separation and recognition challenge: Dataset, task and baselines," in *IEEE Workshop ASRU*. IEEE, 2015, pp. 504–511.

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

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv Preprint arXiv:1301.3781*, 2013.

[21] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. ASLP*, vol. 14, no. 4, pp. 1462–1469, 2006.

[22] Mehrez Souden, Jacob Benesty, and Sofiene Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Trans. ASLP*, vol. 18, no. 2, pp. 260–276, 2009.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015.