Speech Communication 125 (2020) 80–96

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

Masked multi-head self-attention for causal speech enhancement

Aaron Nicolson [*], Kuldip K. Paliwal

Signal Processing Laboratory, Griffith University, Brisbane, Queensland, 4111, Australia

ARTICLE INFO

Keywords:
Speech enhancement
Multi-head attention
MHANet
Deep Xi

ABSTRACT

Accurately modelling the long-term dependencies of noisy speech is critical to the performance of a speech enhancement system. Current deep learning approaches to speech enhancement employ either a recurrent neural network (RNN) or a temporal convolutional network (TCN). However, RNNs and TCNs both demonstrate deficiencies when modelling long-term dependencies. Enter multi-head attention (MHA) — a mechanism that has outperformed both RNNs and TCNs in tasks such as machine translation. By using sequence similarity, MHA possesses the ability to more efficiently model long-term dependencies. Moreover, masking can be employed to ensure that the MHA mechanism remains causal — an attribute critical for real-time processing. Motivated by these points, we investigate a deep neural network (DNN) that utilises masked MHA for causal speech enhancement. The conditions used to evaluate the proposed DNN include real-world non-stationary and coloured noise sources at multiple SNR levels. Our extensive experimental investigation demonstrates that the proposed DNN can produce enhanced speech at a higher quality and intelligibility than both RNNs and TCNs. We conclude that deep learning approaches employing masked MHA are more suited for causal speech enhancement than RNNs and TCNs. ***Availability***—MHANet is available at https://github.com/anicolson/DeepXi

1. Introduction

A speech enhancement algorithm aims to improve the perceived quality and intelligibility of noisy speech (Loizou, 2013). It accomplishes this by suppressing background noise without distorting the speech. Many speech processing systems, such as automatic speech recognition (ASR) systems, speaker verification systems, and mobile communication systems typically rely on the enhancement of noisy speech for robustness. For example, a leading method to increase the robustness of an ASR system is to pre-process noisy speech with a speech enhancement algorithm (Nicolson and Paliwal, 2019b).

Currently, deep learning approaches to speech enhancement are at the forefront of the field. Deep neural networks (DNNs) are utilised to provide a non-linear map from a given noisy speech representation to a target representation. They have been used to map the noisy speech magnitude spectrum to the clean speech magnitude spectrum (Xu et al., 2014), or noisy speech time-domain frames to clean speech time-domain frames (Tamura and Waibel, 1988). Utilising time-domain frames enables the reconstruction of the distorted phase. DNNs have also been used to map the noisy speech spectrum to a time-frequency mask (Narayanan and Wang, 2013), or the *a priori* SNR (Nicolson and Paliwal, 2019a). Time-frequency masks, such as the ideal-ratio mask (IRM), are applied as a suppression function to the noisy speech magnitude spectrum (Narayanan and Wang, 2013). It was found that the IRM is able to outperform the clean speech magnitude spectrum when used as the training target (Wang et al., 2014). *A priori* SNR estimates are used by minimum mean-square error (MMSE) estimators of the clean speech spectrum (Ephraim and Malah, 1985). It was found that the *a priori* SNR as the training target produces higher objective quality scores than the IRM (Nicolson and Paliwal, 2020a). DNNs have also been tasked with computational auditory scene analysis (CASA), whereby a noisy speech magnitude spectrum component is classified as either noise or speech dominant (Narayanan and Wang, 2013; Nicolson and Paliwal, 2018). This is accomplished by using the ideal binary mask (IBM) as the training target, where a value of one and zero corresponds to a speech and noise dominant component, respectively. However, it was found that the IRM is able to significantly outperform the IBM (Wang et al., 2014).

A multitude of deep learning approaches have been successfully applied to speech enhancement. One recent approach is the use of teacher-student learning (Subramanian et al., 2018). An already trained, large *teacher* DNN is used to guide the training of a smaller, *student* DNN. In Tu et al. (2019), a causal student DNN is trained to estimate the improved speech presence probability (ISPP) with guidance from a

[*] Corresponding author.
E-mail addresses: aaron.nicolson@griffithuni.edu.au (A. Nicolson), k.paliwal@griffith.edu.au (K.K. Paliwal).

https://doi.org/10.1016/j.specom.2020.10.004
Received 22 December 2019; Received in revised form 15 October 2020; Accepted 24 October 2020
Available online 29 October 2020

Table 1
Computational complexity per layer and maximum number of sequential operations required to connect any two input and output time-steps of a layer (Vaswani et al., 2017). The sequence length, the kernel size, and the layer size is denoted by L, k, and d, respectively.

Layer	Complex. per layer	Max. seq. operations
Self-atten.	$O(L^2 \cdot d)$	$O(1)$
Convolut.	$O(k \cdot L \cdot d^2)$	$O(\log_k(L))$
Recurrent	$O(L \cdot d^2)$	$O(L)$

non-causal IRM estimator teacher DNN. An advantage of the teacher-student approach is that the student model can be compactly designed. Another approach is to optimise a DNN with respect to one or more objective measures (Fu et al., 2019; Koizumi et al., 2018). While it will perform well on the optimised objective measures, there is a risk that the DNN will not generalise to other measures. Another recent approach is the use of generative adversarial networks (GANs) for speech enhancement (SEGAN) (Pascual et al., 2017). One disadvantage of a SEGAN is that its generator does not have direct access to the clean speech during training, diminishing its performance at lower signal-to-noise ratio (SNR) levels. To mitigate this issue, a high-level GAN (HLGAN) was proposed, where a high-level loss function allows the generator to access both the clean and noisy speech during training (Zhao et al., 2019).

Many DNN types are used for deep learning approaches to speech enhancement. Feed-forward neural networks (FNNs) were amongst the first DNNs used for speech enhancement (Tamura, 1989; Fei Xie and Van Compernolle, 1994; Xu et al., 2014). They were adapted to input a window of several past, present, and future frames (Xu et al., 2015) and were able to significantly outperform previous speech enhancement algorithms such as the decision-directed (DD) approach (Ephraim and Malah, 1984). However, FNNs are only capable of modelling local dependencies. Speech exhibits non-linguistic long-term dependencies, such as gender, dialect, speaking rate, and emotional state (Bengio et al., 1994; Pisoni, 1993). Moreover, it has been demonstrated that modelling the non-linguistic information of speech can improve speech enhancement performance (Potamitis et al., 2002; Rao Naidu and Srinivasan, 2012). While coloured noise sources display only local dependencies (e. g. a fan), non-stationary noise sources inherently display long-term dependencies (e.g. music). The preceding points indicate that modelling the long-term dependencies of the target speech and background noise is important for speech enhancement.

With the ability to model long-term dependencies, recurrent neural networks (RNNs) employing long short-term memory (LSTM) have demonstrated a higher speech enhancement performance than FNNs (Chen and Wang, 2017; Liu et al., 2018). The cell state of LSTM grants it the ability to remember important information about the noise and speech (Gers et al., 1999). However, RNNs require a large number of parameters and lengthy training times. Moreover, the memory of LSTM is limited, making it prone to forgetting distant information (Li et al., 2018). Temporal convolutional networks (TCNs) were soon able to match the speech enhancement performance of RNNs while consuming significantly fewer parameters and requiring markedly less time to train (Rethage et al., 2018). TCNs utilise causal dilated kernels to garner a fixed-size receptive field over previous frames, allowing the modelling of long-term dependencies (Bai et al., 2018). However, the performance of a TCN is adversely affected when events occur out of expected order due to the positional nature of the kernels.

Recently, the Transformer network outperformed both RNN and TCN-based models on a machine translation task (Vaswani et al., 2017). Derivatives of the Transformer network have also been applied to tasks such as language modelling (Devlin et al., 2019), speech recognition (Sperber et al., 2018), and medical diagnoses (Wang et al., 2019). The key module of the Transformer network is *multi-head attention* (MHA). MHA utilises multiple heads, with each employing an attention mechanism. The sequence similarity between all time-steps is used by the attention mechanism to compute a new representation, granting it the ability to model long-term dependencies. Additionally, when the sequence length is less than the size of the layer, the attention mechanism boasts less complexity per layer than its RNN and TCN-based counterparts. This is shown in Table 1, where self-attention is the attention type used by the encoder of the Transformer network.

As described in Vaswani et al. (2017), a key factor affecting the ability of a DNN to learn long-term dependencies is the number of sequential operations required to connect input and output time-steps of a layer. The fewer the number of operations required to connect any combination of input and output time-step, the easier it is to learn long-term dependencies (Kolen and Kremer, 2001). The maximum number of sequential operations required to connect any two input and output time-steps of a layer is shown in Table 1. It can be seen that the maximum number of sequential operations for self-attention is constant, whereas the maximum number of sequential operations for recurrent and convolutional layers depends on L. This means that the distance between an input and an output time-step does not affect self-attention. This key advantage allows MHA to more efficiently model long-term dependencies than RNNs and TCNs (Vaswani et al., 2017).

One aspect to consider when developing a deep-learning approach to speech enhancement is causality. Most speech processing applications require real-time processing (Zhao et al., 2019; McGraw et al., 2016; Prabhavalkar et al., 2016; Chan and Lane, 2016). The responsiveness of a real-time system is negatively affected when utilising non-causal modules, as future frames are required. Moreover, the demand for embedded real-time systems in mobile devices is increasing as more sophisticated processors are employed (Yun et al., 2018; Wang et al., 2020). Therefore, most speech processing applications require a causal deep learning approach to speech enhancement. All of the aforementioned DNN types possess the ability to be causal. An FNN is causal if its input includes only current and previous time-steps. For RNNs, only unidirectional RNNs can be used, as opposed to bidirectional RNNs (Schuster and Paliwal, 1997). For TCNs, kernels that consider only current and previous time-steps are permitted. For MHA, causality is upheld when similarities considering future time-steps are masked.

In this work, we investigate masked MHA for causal speech enhancement. This is motivated by the following points: 1) modelling the long-term dependencies of noisy speech is critical for speech enhancement, and 2) MHA possesses the ability to more efficiently model long-term dependencies than RNNs and TCNs. This indicates that MHA has the potential to outperform RNNs and TCNs at the task of speech enhancement. The blocks of the proposed MHA network (MHANet) are identical to those used in the encoder of the Transformer network, except that masking is applied to ensure causality. Unlike the Transformer network, the MHANet does not utilise positional encoding, as we find that there is sufficient information about the order of events encoded into the noisy speech. In this work, the MHANet is tasked with estimating the *a priori* SNR for an MMSE approach.

In research proposed simultaneously to this work, MHA with no masking was used for speech enhancement (Jin et al., 2020; Liao et al., 2019). In Kim et al. (2020), the encoder of the Transformer network was used to estimate the IRM, called the Transformer with Gaussian-weighted self-attention (T-GSA). A Gaussian weighting was applied to the attention weights to attenuate according to the distance between the current frame and past/future frames. T-GSA is jointly optimised for two objective measures, namely the perceptual evaluation of speech quality (PESQ) (Rix et al., 2001) and the signal-to-distortion ratio (SDR). In Koizumi et al. (2020), a network comprising of two-dimensional convolutional units, bidirectional LSTM cells, and MHA modules was used to estimate a complex-valued time-frequency (TF) mask. Speaker aware features were also employed, along with multi-task learning of speech enhancement and speaker identification. Furthermore, the network is jointly optimised for cross-entropy and

SDR. Both of these networks utilise non-causal MHA modules and were both able to outperform bidirectional RNN, convolutional neural network (CNN), and GAN-based speech enhancement systems.

In this paper, we first describe the employed speech enhancement framework in Section 2. The proposed MHANet is described in Section 3. In Section 4, the validation error of the MHANet is compared to that of an RNN and a TCN. In Section 5, the sequence similarity of the heads from the MHANet are observed, in order to understand what the heads attend to during speech enhancement. In Sections 6 and 7, the speech enhancement performance and the execution time of the MHANet is compared to both RNNs and TCNs. The MHANet is also compared to multiple recent causal and non-causal deep learning approaches to speech enhancement in Section 8. Conclusions are drawn in Section 9.

2. Background

2.1. Analysis, modification, & synthesis framework

The short-time Fourier analysis, modification, and synthesis (AMS) framework is used here for speech enhancement (Allen, 1977; Allen and Rabiner, 1977). The AMS framework consists of three stages: (1) the analysis stage, where noisy speech undergoes short-time Fourier transform (STFT) analysis; (2) the modification stage, where the noisy speech magnitude spectrum is modified; and (3) the synthesis stage, where the enhanced speech is synthesised by applying the inverse STFT (ISTFT).

In the time-domain, the noisy speech signal, $x[n]$, is given by:

$$x[n] = s[n] + d[n], \tag{1}$$

where $s[n]$, and $d[n]$ denote the clean speech and uncorrelated additive noise, respectively, and n denotes the discrete-time index. The noisy speech is analysed frame-wise using the running STFT (Vary and Martin, 2006):

$$X[l,k] = \sum_{n=0}^{N_d-1} x[n + lN_s]w[n]e^{-j2\pi nk/N_d}, \tag{2}$$

where l denotes the frame index, k denotes the discrete-frequency index, N_d denotes the frame duration in discrete-time samples, N_s denotes the frame shift in discrete-time samples, and $w[n]$ is an analysis window function.

In polar form, the noisy speech spectrum is expressed as

$$X[l,k] = |X[l,k]|e^{j\angle X[l,k]}, \tag{3}$$

where $|X[l,k]|$ and $\angle X[l,k]$ denote the noisy speech magnitude and phase spectrums, respectively. The modified magnitude spectrum is then formed by enhancing the noisy speech magnitude spectrum. The modified magnitude spectrum is an estimate of the clean speech magnitude spectrum, and is denoted by $|\widehat{S}[l,k]|$. The modified spectrum is constructed by combining the modified magnitude spectrum with the noisy speech phase spectrum:

$$Y[l,k] = |\widehat{S}[l,k]|e^{j\angle X[l,k]}. \tag{4}$$

The synthesis stage involves applying the ISTFT to the modified spectrum. First, the inverse discrete Fourier transform is applied to the modified spectrum:

$$y_f[l,n] = \frac{1}{N_d}\sum_{k=0}^{N_d-1} Y[l,k]e^{j2\pi nk/N_d}, \tag{5}$$

where $y_f[l,n]$ is the framed enhanced speech. The overlap-add method is subsequently applied to produce the final enhanced speech (Crochiere, 1980):

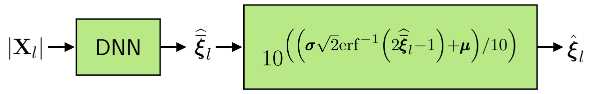
$|\mathbf{X}_l|$
DNN
$\widehat{\bar{\xi}}_l$
$10^{\left(\left(\boldsymbol{\sigma}\sqrt{2}\mathrm{erf}^{-1}\left(2\widehat{\bar{\xi}}_l-1\right)+\boldsymbol{\mu}\right)/10\right)}$
$\hat{\xi}_l$

Fig. 1. Deep Xi *a priori* SNR estimation framework.

$$y[n] = \frac{\sum_{l=-\infty}^{\infty} y_f[l, n - lN_s]}{\sum_{l=-\infty}^{\infty} w[n - lN_s]}, \tag{6}$$

where $w[n]$ is a synthesis window function.

In this work, the Hann window function is used for analysis and synthesis, with a frame-duration of 32 ms ($N_d = 512$) and a frame-shift of 16 ms ($N_s = 256$). The 257-point single-sided noisy speech magnitude spectrum, which includes both the DC frequency component and the Nyquist frequency component is used as the input to the MHANet.

2.2. Deep Xi framework

In this work, the proposed MHANet is compared to both RNNs and TCNs within the Deep Xi framework. Deep Xi is a deep learning approach to *a priori* SNR estimation (Nicolson and Paliwal, 2019a), and is depicted in Fig. 1. The Deep Xi framework consists of two stages. For the first stage, a DNN estimates a mapped version of the *a priori* SNR, $\widehat{\bar{\xi}}_l = \{\widehat{\bar{\xi}}[l,0], \widehat{\bar{\xi}}[l,1], \ldots, \widehat{\bar{\xi}}[l,K-1]\}$, from the noisy speech magnitude spectrum, $|\mathbf{X}_l| = \{|X[l,0]|, |X[l,1]|, \ldots, |X[l,K-1]|\}$, where K is the number of discrete-frequency bins for each frame ($K = N_d/2+1$). For the second stage, the *a priori* SNR estimate, $\widehat{\xi}_l$, is computed from the mapped *a priori* SNR estimate, $\widehat{\bar{\xi}}_l$. The mapped *a priori*SNR, $\bar{\xi}_l$, and the computation of the *a priori* SNR estimate during the second stage is described in Section 2.2.2. The *a priori* SNR estimated using the Deep Xi framework is used by an MMSE clean speech magnitude spectrum estimator, as described in Section 2.2.1.

2.2.1. MMSE approaches to speech enhancement

The *a priori* SNR estimated using the Deep Xi framework is used by an MMSE approach to estimate the clean speech magnitude spectrum for Eq. (4). An example is the MMSE log-spectral amplitude (MMSE-LSA) estimator (Ephraim and Malah, 1985):

$$|\widehat{S}[l,k]| = |X[l,k]|\frac{\xi[l,k]}{\xi[l,k]+1}\exp\left\{\frac{1}{2}\int_{\nu[l,k]}^{\infty}\frac{e^{-t}}{t}dt\right\}, \tag{7}$$

where $\nu[l,k] = \frac{\xi[l,k]}{\xi[l,k]+1}\gamma[l,k]$, $\xi[l,k]$ is the *a priori* SNR and $\gamma[l,k]$ is the *a posteriori* SNR. The *a priori* SNR is defined as

$$\xi\left[l,k\right] = \frac{\lambda_s[l,k]}{\lambda_d[l,k]}, \tag{8}$$

where $\lambda_s[l,k] = \mathrm{E}\{S[l,k]^2\}$ is the variance of the clean speech spectral component, and $\lambda_d[l,k] = \mathrm{E}\{D[l,k]^2\}$ is the variance of the noise spectral component. The *a posteriori* SNR is defined as

$$\gamma[l,k] = \frac{|X[l,k]|^2}{\lambda_d[l,k]}. \tag{9}$$

For the Deep Xi framework, the maximum likelihood *a posteriori* SNR estimate is used: $\widehat{\gamma}[l,k] = \xi[l,k] + 1$.

2.2.2. Mapped a priori SNR training target

The training target for the DNN in Fig. 1 is the mapped *a priori* SNR (Nicolson and Paliwal, 2019a), which is a mapped version of the instantaneous *a priori* SNR. During the training phase, the clean speech and noise in Eq. (8) are known completely. This means that $\lambda_s[l,k]$ and

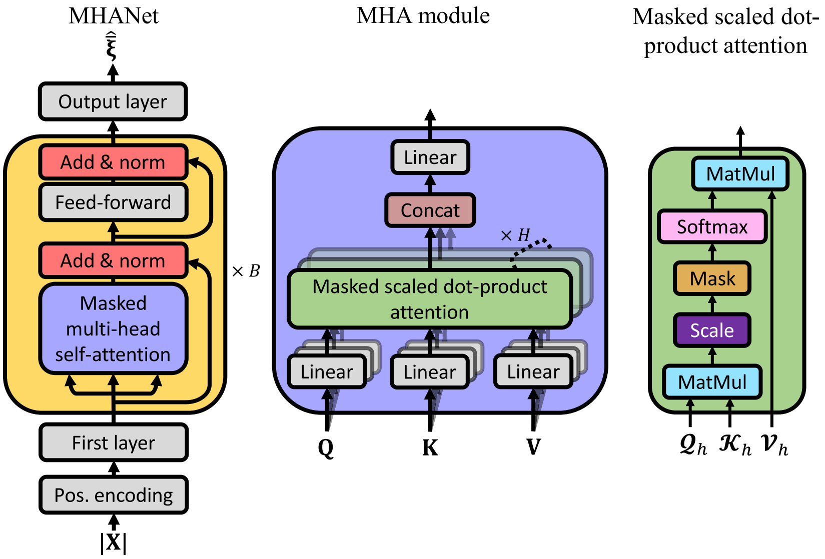

Fig. 2. (left) Proposed multi-head attention network (MHANet), (middle) multi-head attention module (MHA), and (right) masked scaled dot-product attention.

$\lambda_d[l,k]$ can be replaced with the squared magnitude of the clean speech and noise spectral components, respectively, giving the instantaneous *a priori* SNR.

In Nicolson and Paliwal (2019a), the instantaneous *a priori* SNR in dB, $\xi_{\text{dB}}[l,k] = 10\log_{10}(\xi[l,k])$, was mapped to the interval $[0,1]$ in order to improve the rate of convergence of the used stochastic gradient descent algorithm. The cumulative distribution function (CDF) of $\xi_{\text{dB}}[l,k]$ was used as the map. It can be seen (Nicolson and Paliwal, 2019a, Fig.2 (top)) that the distribution of $\xi_{\text{dB}}[l,k]$ for a given frequency component follows a normal distribution. It was thus assumed in Nicolson and Paliwal (2019a) that $\xi_{\text{dB}}[l,k]$ is distributed normally with mean μ_k and variance σ_k^2: $\xi_{\text{dB}}[l,k] \sim \mathcal{N}(\mu_k, \sigma_k^2)$. The map is given by

$$\bar{\xi}[l,k] = \frac{1}{2}\left[1 + \text{erf}\left(\frac{\xi_{\text{dB}}[l,k] - \mu_k}{\sigma_k\sqrt{2}}\right)\right], \tag{10}$$

where $\bar{\xi}[l,k]$ is the mapped *a priori* SNR and erf is the error function. Following Nicolson and Paliwal (2019a), the statistics of $\xi_{\text{dB}}[l,k]$ for each noisy speech spectral component were found over 1 000 samples of the training set.

During inference, $\hat{\xi}[l,k]$ is found from $\hat{\bar{\xi}}[l,k]$ as follows:

$$\hat{\xi}[l,k] = 10^{\left(\left(\sigma_k\sqrt{2}\text{erf}^{-1}\left(2\hat{\bar{\xi}}[l,k]-1\right)+\mu_k\right)/10\right)}. \tag{11}$$

Eq. (11) is shown in Fig. 1, where $\boldsymbol{\mu} = \{\mu_0, \mu_1, \ldots, \mu_{K-1}\}$ are the means for each discrete-frequency bin and $\boldsymbol{\sigma} = \{\sigma_0, \sigma_1, \ldots, \sigma_{K-1}\}$ are the corresponding standard deviations.

3. MHANet for speech enhancement

In this section, we describe the proposed MHANet for speech enhancement, as illustrated in Fig. 2 (left). For speech enhancement, the sequence length of the input and target are equal. This means that the encoder-decoder structure of the Transformer network is not required. Thus, a network can be formed simply by stacking blocks of computational units from the input to the output. The blocks of the MHANet are identical to those used by the encoder of the Transformer network, except that masking is employed for causality. Each block includes the fundamental component of the Transformer network — multi-head attention (MHA). As the MHANet is derived from the Transformer network, the notation in this section is based on that used in Vaswani et al. (2017).

The proposed MHANet is described from input to output as follows. The noisy speech magnitude spectra, $|\mathbf{X}| \in \mathbb{R}^{L\times K}$, is the input to the MHANet. An attention mechanism is unaware of the order of events in a sequence. Hence, it must have positional information encoded into its input. Here, the trigonometric positional encoding (PE) used for the Transformer network is employed (Vaswani et al., 2017). We investigate three different ways of including the positional encoding with the input: the addition of the positional encoding (Vaswani et al., 2017), the concatenation of the positional encoding (Sperber et al., 2018), and no positional encoding. Including no positional encoding assumes that there is enough information encoded in the noisy speech about the position of each event. After the positional encoding is included, the first layer is used to project the input to a size of d_{model}. The first layer from Nicolson and Paliwal (2019a) is used here: $\max(0, \text{LN}(|\mathbf{X}|\mathbf{W}^I + \mathbf{b}^I))$, where LN is frame-wise layer normalisation Ba et al. (2016), $\mathbf{W}^I \in \mathbb{R}^{K\times d_{model}}$, and $\mathbf{b}^I \in \mathbb{R}^{d_{model}}$.

Next, B blocks (called "layers" in Vaswani et al. (2017)) are cascaded, where each block includes an MHA module, a two-layer FNN, residual connections (He et al., 2016), and frame-wise layer normalisation. The blocks are further described in Section 3.1. After the blocks is the output layer, which is a sigmoidal feed-forward layer as in Nicolson and Paliwal (2019a). The MHANet is trained to estimate the mapped *a priori* SNR, $\bar{\boldsymbol{\xi}} \in \mathbb{R}^{L\times K}$, as described in Section 2.2.2. The optimisation method used to train the MHANet is described in Section 3.3.

3.1. MHANet blocks

The MHA module is the first component of the block, and is illustrated in Fig. 2 (middle). The MHA module takes as input a set of L queries ($\mathbf{Q} \in \mathbb{R}^{L\times d_{model}}$), keys ($\mathbf{K} \in \mathbb{R}^{L\times d_{model}}$), and values ($\mathbf{V} \in \mathbb{R}^{L\times d_{model}}$), where L is the number of frames, and d_{model} is the size of each query, key, and value. For a detailed explanation about what $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent, we refer the reader to the work in Vaswani et al. (2017). $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ in this case are all replicas of the input to the block. Thus, the specific form of MHA that is used is *multi-head self-attention*.

A total of H heads of *masked scaled dot-product attention* are used in each MHA module, where $h = \{1, 2, \ldots, H\}$ is the head index. For head h, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are linearly projected: $\mathcal{Q}_h = \mathbf{Q}\mathbf{W}_h^Q$, $\mathcal{K}_h = \mathbf{K}\mathbf{W}_h^K$, and $\mathcal{V}_h = \mathbf{V}\mathbf{W}_h^V$, where $\mathbf{W}_h^Q \in \mathbb{R}^{d_{model}\times d_k}$, $\mathbf{W}_h^K \in \mathbb{R}^{d_{model}\times d_k}$, and $\mathbf{W}_h^V \in \mathbb{R}^{d_{model}\times d_v}$ are learned weight matrices. The projected queries and keys are of size d_k, and the projected values are of size d_v. This allows each head to operate on different, learned linear projections of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$. To control the size of each head, the following policy from Vaswani et al. (2017) is used: $d_k = d_v = d_{model}/H$. The MHA module is defined as

$$\mathrm{MHA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{concat}(\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_H)\mathbf{W}^O, \tag{12}$$

where $\mathbf{A}_h = \mathrm{Attention}(\mathcal{Q}_h,\mathcal{K}_h,\mathcal{V}_h)$ is the output for head $h(\mathbf{A}_h \in \mathbb{R}^{L\times d_k})$. The masked scaled dot-product attention mechanism used for each head, $\mathrm{Attention}(\mathcal{Q}_h,\mathcal{K}_h,\mathcal{V}_h)$, is described in Section 3.2. It can be seen that the outputs from all of the heads are concatenated and linearly projected using the learned weight matrix $\mathbf{W}_h^O \in \mathbb{R}^{Hd_v\times d_{model}}$, forming the final output of the MHA module. A residual connection is applied from the input to the output of the MHA module, which is followed by frame-wise layer normalisation.

The second half of the block includes a two-layer feed-forward neural network (FNN):

$$\mathrm{FNN}(\mathbf{Z}) = \max\left(0,\mathbf{Z}\mathbf{W}^1 + \mathbf{b}^1\right)\mathbf{W}^2 + \mathbf{b}^2, \tag{13}$$

where $\mathbf{Z} \in \mathbb{R}^{L\times d_{model}}$ is the input, $\mathbf{W}^1 \in \mathbb{R}^{d_{model}\times d_{ff}}$, $\mathbf{b}^1 \in \mathbb{R}^{d_{ff}}$, $\mathbf{W}^2 \in \mathbb{R}^{d_{ff}\times d_{model}}$, and $\mathbf{b}^2 \in \mathbb{R}^{d_{model}}$. Hence, the inner layer has a size of d_{ff}. A residual connection is applied from the input to the output of the FNN, which is followed by frame-wise layer normalisation. As in Vaswani et al. (2017), a dropout rate of P_{drop} is applied before each residual connection during training.

3.2. Masked scaled dot-product attention

The masked scaled dot-product attention mechanism for head h, as shown in Fig. 2 (right), takes as input a set of L queries ($\mathcal{Q}_h \in \mathbb{R}^{L\times d_k}$), keys ($\mathcal{K}_h \in \mathbb{R}^{L\times d_k}$), and values ($\mathcal{V}_h \in \mathbb{R}^{L\times d_v}$). Masked scaled dot-product attention is computed as

$$\mathrm{Attention}(\mathcal{Q}_h,\mathcal{K}_h,\mathcal{V}_h) = \mathrm{softmax}\left(\mathbf{M} + \frac{\mathcal{Q}_h\mathcal{K}_h^\top}{\sqrt{d_k}}\right)\mathcal{V}_h. \tag{14}$$

The similarity matrix computed from the dot product of $\mathcal{Q}_h$ and $\mathcal{K}_h^\top$ forms the unnormalised weights of the attention mechanism. After scaling by $1/\sqrt{d_k}$, $\mathbf{M} \in \mathbb{R}^{L,L}$ is used to mask out similarities that include future frames, ensuring causality. As the ensuing operation is the softmax function, masking is performed by adding $-\infty$. Following masking, each row of the sequence similarity matrix is normalised into a probability distribution using the softmax activation function. Finally, a new representation is computed via the dot product of the normalised similarity matrix and $\mathcal{V}_h$.

3.3. Optimisation method

As in Vaswani et al. (2017), the *Adam* optimiser (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ is used for training the MHANet, where the learning rate, α, is controlled over the course of training as

$$\alpha = d_{model}^{-0.5}\cdot\min\left(\psi^{-0.5},\psi\cdot\Psi^{-1.5}\right), \tag{15}$$

where ψ is the training step and Ψ is the number of warmup steps. The learning rate increases linearly with ψ until Ψ is exceeded and then decreases proportionally to the inverse square root of ψ. This strategy was used in Vaswani et al. (2017) to stabilise learning during the initial stages of training, as MHA has been found difficult to train.

4. Validation error

In this section, the validation error of the MHANet is compared to that of an RNN and a TCN, where the RNN is a residual LSTM (ResLSTM) network (Nicolson and Paliwal, 2019a) and the TCN is a residual network (ResNet) (Zhang et al., 2020). The ResLSTM network and the ResNet are described in Section 4.3. The MHANet, ResNet, and ResLSTM are implemented within the Deep Xi framework using TensorFlow 1.14 (Abadi et al., 2015).

Table 2
Minimum, average, and maximum duration of the recordings from the sets of the speech corpora.

Set	Min. (s)	Avg. (s)	Max. (s)
train-clean-100	1.4	12.3	17.2
CSTR VCTK	1.2	3.6	15.1
Train *si** & *sx**	0.9	3.1	7.8

As in Nicolson and Paliwal (2019a), the cross-entropy between the mapped *a priori* SNR $\bar{\xi}[l,k]$, and its estimate $\widehat{\bar{\xi}}[l,k]$, is used as the error:

$$\mathscr{E} = -\frac{1}{LK}\sum_{l=1}^{L}\sum_{k=0}^{K-1}\bar{\xi}[l,k]\log(\widehat{\bar{\xi}}[l,k]) + (1-\bar{\xi}[l,k])\log(1-\widehat{\bar{\xi}}[l,k]). \tag{16}$$

The training and validation sets used in this subsection are described in Section 4.1, followed by the training strategy in Section 4.2. The hyperparameter search for the MHANet is presented in Appendix A. A key finding is that information about the order of events is sufficiently embedded in the noisy speech magnitude spectrum input. Thus, no positional encoding is required. Moreover, it was found that utilising dropout hindered performance. The set of hyperparameters that performed best included no positional encoding, $B = 5$, $d_{ff} = 1\,024$, $d_{model} = 256$, $H = 8$, $P_{drop} = 0.0$, and $\Psi = 40\,000$. This set of hyperparameters is used for the MHANet for the remainder of this work.

4.1. Training & validation set

Here, we describe the clean speech and noise recordings used for training. The clean speech recordings from the following speech corpora are included: the *train-clean-100* set from the Librispeech corpus (Panayotov et al., 2015) (28 539 recordings), the CSTR VCTK corpus (Veaux et al., 2017) (42 015 recordings), and the *si** and *sx** training sets from the TIMIT corpus (Garofolo et al., 1993) (3 696 recordings). This gives a total of 74 250 clean speech recordings. The minimum, average, and maximum duration of the recordings from the sets of the speech corpora are given in Table 2.

The noise recordings from the following noise datasets are included: the QUT-NOISE dataset (Dean et al., 2010), the Nonspeech dataset (Hu and Wang, 2010), the RSG-10 dataset (*voice babble, F16*, and *factory welding* are excluded as they are used for testing in later sections) (Steeneken and Geurtsen, 1988), the Urban Sound dataset (*street music* recording no. 26 270 is excluded as it is used for testing in later sections) (Salamon et al., 2014), the Environmental Background Noise dataset (Saki et al., 2016), the noise set from the MUSAN corpus Snyder et al. (2015), multiple FreeSound packs,[1] and coloured noise recordings (with an α value ranging from -2 to 2 in increments of 0.25). Noise recordings that are over 30 sin length are split into 30 s or less segments. This gives a total of 17 458 noise recordings, each of a length less than or equal to 30 s.

For the validation set, 1 000 clean speech and noise recordings are randomly selected (without replacement) and removed from the training set. Each clean speech recording is paired with one of the noise recordings. The clean speech recording is then mixed with a random section of the noise recordings at a randomly selected SNR level between -10 to 20 dB in 1 dB increments. This forms 1 000 noisy speech signals for the validation set.

All clean speech and noise recordings are single-channel, with a

[1] Freesound packs that are used include 147, 199, 247, 379, 622, 643, 1 133, 1 563, 1 840, 2 432, 4 366, 4 439, 4 780, 8 420, 14 826, 15 046, 15 097, 15 598, 16 204, 17 266, 17 403, 17 430, 17 468, 17 579, 19 093, 20 237, 20 241, 21 558, 22 953, and 24 590.

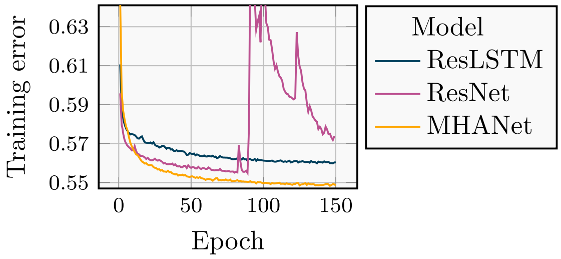

Fig. 3. The training error of the ResLSTM network, the ResNet, and the proposed MHANet.

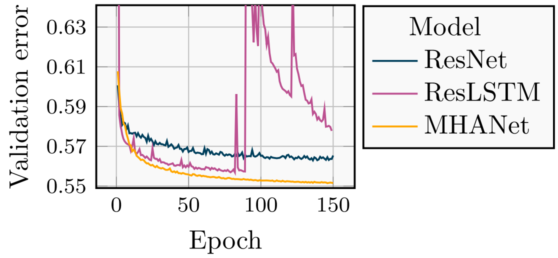

Fig. 4. The validation error of the ResLSTM network, the ResNet, and the proposed MHANet.

sampling frequency of 16 kHz (recordings with a higher sampling frequency are down-sampled to 16 kHz). A description of how the noisy speech is formed for each training iteration is given in Section 4.2.

4.2. Training strategy

The following strategy is used for training:

- As in Nicolson and Paliwal (2019a), cross-entropy is used as the error function — as per Eq. (16).
- For the ResLSTM network and the ResNet, the *Adam* algorithm (Kingma and Ba, 2014) with default hyper-parameters is used for gradient descent optimisation.
- Gradients are clipped between $[-1, 1]$.
- A mini-batch size of 10 noisy speech signals is used for each training iteration.
- The noisy speech signals for each mini-batch are computed on the fly as follows: each clean speech recording selected for the mini-batch is mixed with a random section of a randomly selected noise recording at a randomly selected SNR level (− 10 to 20 dB, in 1 dB increments).
- A total of 7 325 training iterations occurs for each epoch (no. of clean speech recordings divided by the mini-batch size).
- The selection order for the clean speech recordings is randomised for each epoch.

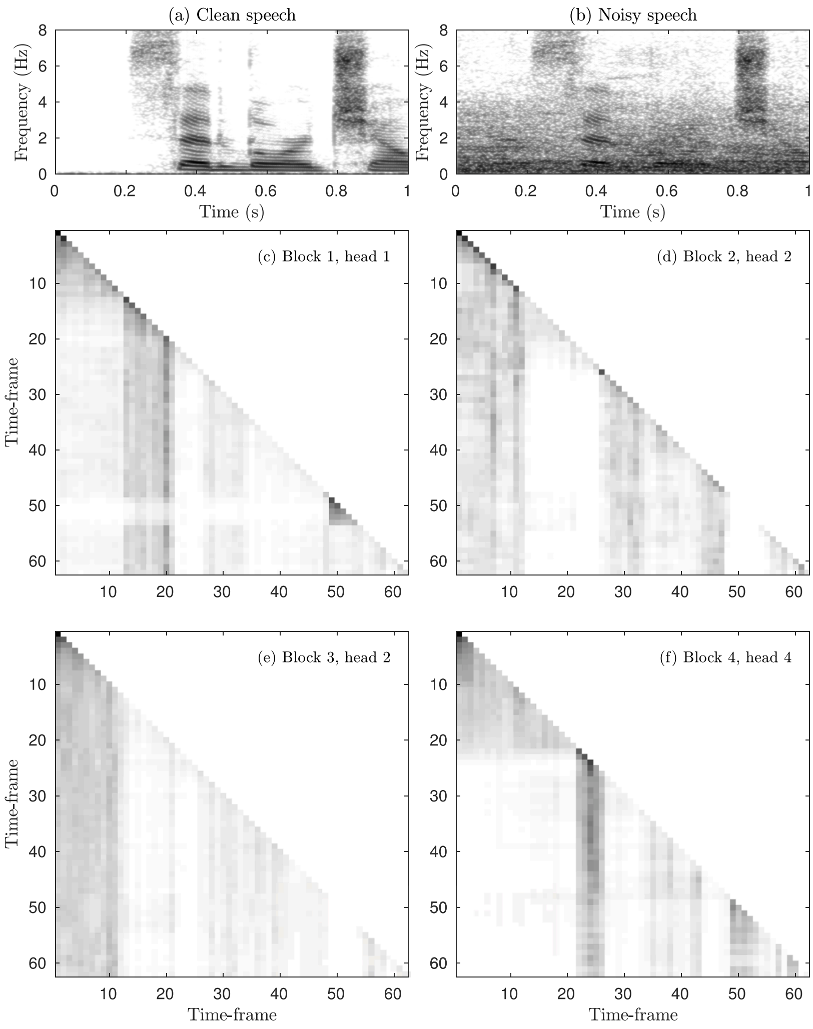

Fig. 5. (a) Clean speech spectrogram of female speaker 5 683 uttering the first second of sentence 1 from book 32 865, “Said Lord Chelford, addressing me” (from the test set described in Section 7.1). (b) A recording of *voice babble* mixed with (a) at an SNR level of − 5 dB. Attention weights produced by the MHANet from (c) head one of block one, (d) head two of block two, (e) head two of block three, and (e) head four of block four. The level of darkness indicates the magnitude of each attention weight.

4.3. Validation error comparison

In this subsection, the validation error of the MHANet is compared to that of an RNN and a TCN. The RNN is the residual LSTM (ResLSTM) network from Nicolson and Paliwal (2019a). It consists of five residual blocks, where each block consists solely of an LSTM cell of size 512. The TCN is the residual network (ResNet) from Zhang et al. (2020). It utilises 40 bottleneck residual blocks, where each block comprises of three convolutional units. The input and output size of each block is 256. The number of filters is 64 for the first and second convolutional units, and 256 for the third convolutional unit. A kernel size of one is used for the first and third convolutional units, and three for the second convolutional unit. The dilation rate, d, for the kernel of the second convolutional unit is controlled via the index of each block, b. The dilation rate is cycled as the block index increases: $d = 2^{(b-1 \bmod (\log_2(D)+1))}$, where $\bmod$ is the modulo operation, and D is the maximum dilation rate which is set to 16. Each convolutional unit is pre-activated using the ReLU activation function (Nair and Hinton, 2010) followed by frame-wise layer normalisation. The ResLSTM network and the ResNet are both trained using the training and validation set described in Section 4.1 and the training strategy described in Section 4.2.

The training and validation error attained by each of the DNNs is shown in Figs. 3 and 4. Each network is trained for 150 epochs. The time taken for one training epoch is approximately 7 h for the ResLSTM network, 40 min for the ResNet, and 30 min for the MHANet on an NVIDIA GTX 1080 Ti. It thus took the ResLSTM network, the ResNet, and the MHANet 1 050, 100, and 75 h to train, respectively. It can be seen that the MHANet produced the lowest training and validation error. The training process for the MHANet is also more stable than that of the ResLSTM network. The training and validation error for the ResLSTM network after epoch 80 becomes volatile. This indicates the occurrence of the exploding gradient problem.

5. Attention weights

In this section, we observe the attention weights of different self-attention heads in the MHANet. The attention weights are highly interpretable, allowing us to observe what each head attends to during speech enhancement. Epoch 150 for configuration *D* from Section A.2 is the used MHANet. The attention weights for the heads of different blocks are shown from Fig. 5(c) to (f). These are the normalised weights attained after the softmax function is applied to the scaled dot product of $\mathcal{Q}_i$ and $\mathcal{K}_i^\top$ in Eq. (14). The noisy speech given to the MHANet is shown in Fig. 5 (b). It is the clean speech from Fig. 5 (a) mixed with *voice babble* at an SNR level of -5 dB.

The attention weights for the first head of block one are shown in Fig. 5 (c). It can be seen that a significant amount of attention is payed to the unvoiced phoneme from frame 13 to 21. This unvoiced phoneme is attended to for the remainder of the sequence, except during the unvoiced phoneme from frame 49 to 53. This indicates that head one of block one attends to the unvoiced phonemes of the target speaker.

The attention weights for the second head of block two are shown in Fig. 5 (d). It attends to three regions that are dominated by noise. These three regions are from frame 1 to 12, 26 to 33, and 44 to 47. It can be seen that each noise dominated region is attended to for the remainder of the sequence — after being observed. The attention weights for the second head of block three are shown in Fig. 5 (e). They are similar to the attention weights in Fig. 5 (d), except that more attention is payed to the region from frame 1 to 12.

The attention weights for the fourth head of block four are shown in Fig. 5 (f). The head first pays attention to the unvoiced phoneme from frame 11 to 21. However, the unvoiced phoneme is not attended to after the head observes the voiced phoneme from frame 22 to 26. This voiced phoneme is attended to for the remainder of the sequence. Another region garnering a high amount of attention is the unvoiced phoneme from frame 49 to 53. The head also pays slight attention to the voiced phonemes from frame 34 to 43 and 57 to 60. It is clear that head four of block four has learnt to focus on the unvoiced and voiced phonemes of the target speaker. In summary, a self-attention head in the MHANet attends to regions of speech and/or noise in the noisy speech.

6. *A priori* SNR estimation accuracy

MMSE approaches to speech enhancement are directly affected by the accuracy of the employed *a priori* SNR estimator. Additionally, a more accurate *a priori* SNR estimator can increase the performance of ideal binary mask (IBM) estimation for missing data approaches (Nicolson and Paliwal, 2020c). Hence, we evaluate the *a priori* SNR estimation accuracy of the MHANet within the Deep Xi framework. Epoch 150 for the MHANet configuration described in Section 4 is used. It is compared to the ResLSTM network and the ResNet from Section 4. Epoch 80 and 150 are used for the ResLSTM network and the ResNet, respectively. SD levels for previous *a priori* SNR estimation methods, including the decision-directed (DD) approach (Ephraim and Malah, 1984), the two-step noise reduction (TSNR) technique (Plapous et al., 2004), harmonic regeneration noise reduction (HRNR) (Plapous et al., 2005), and selective cepstro-temporal smoothing (SCTS) (Breithaupt et al., 2008) are also included. Each of the previous *a priori* SNR estimation methods utilises the noise estimator from Gerkmann and Hendriks (2012). The test set from Nicolson and Paliwal (2019a) is used for evaluation, and is described in Section 6.1. This test set is selected as it has been used previously to evaluate the accuracy of *a priori* SNR estimators. Spectral distortion is employed to evaluate the *a priori* SNR estimation accuracy, as described in Section 6.2. The spectral distortion levels attained by the MHANet are discussed in Section 6.3.

6.1. Test set

The test set from Nicolson and Paliwal (2019a) is described in this subsection. Recordings of four real-world noise sources, including two non-stationary and two coloured, are included in the test set. The two real-world non-stationary noise sources include *voice babble* from the RSG-10 noise dataset (Steeneken and Geurtsen, 1988) and *street music* (recording no. 26 270) from the Urban Sound dataset (Salamon et al., 2014). The two real-world coloured noise sources include *F16*and *factory welding* from the RSG-10 noise dataset (Steeneken and Geurtsen, 1988). 10 clean speech recordings are randomly selected (without replacement) from the TSP corpus (only adult speakers are included) (Kabal, 2002) for each of the four noise recordings. The minimum, mean, and maximum duration of the recordings in the TSP corpus are 1.34, 2.37, and 4.79 s, respectively. To create the noisy speech, a random section of the noise recording is selected and mixed with the clean speech at the following SNR levels: -5 to 15 dB, in 5 dB increments. This creates a test set of 200 noisy speech signals. The noisy speech signals are single channel, with a sampling frequency of 16 kHz.

6.2. Evaluation metric

The frame-wise spectral distortion (SD) (Nicolson and Paliwal, 2019a) is used to evaluate the accuracy of the MHANet. The SD is defined as the root-mean-square difference between the *a priori* SNR estimate in dB, $\widehat{\xi}_{\text{dB}}[l,k]$, and the instantaneous case in dB, $\xi_{\text{dB}}[l,k]$, for the l^{th} frame:

$$D_l^2 = \frac{1}{K/2+1} \sum_{k=0}^{K-1} \left[\xi_{\text{dB}}[l,k] - \widehat{\xi}_{\text{dB}}[l,k] \right]^2. \tag{17}$$

Average SD levels are found over all frames for each test condition.

Table 3
A priori SNR estimation SD levels for each of the *a priori* SNR estimators. The lowest SD for each condition is shown in boldface. The used test set is described in Section 6.1.

Noise	$\hat{\xi}[n,k]$	SNR level (dB)				
		− 5	0	5	10	15
Voice babble	DD	19.0	18.1	17.6	17.4	17.6
	TSNR	18.8	17.9	17.4	17.3	17.5
	HRNR	20.4	19.7	19.4	19.3	19.5
	SCTS	17.9	17.2	16.9	16.9	17.2
	ResNet	15.1	14.3	13.7	13.3	12.9
	ResLSTM	14.9	14.1	13.5	13.0	12.7
	MHANet	**14.7**	**13.9**	**13.2**	**12.7**	**12.4**
Street music	DD	20.4	19.0	18.0	17.4	17.1
	TSNR	20.1	18.8	17.8	17.2	16.9
	HRNR	20.3	19.2	18.5	18.2	18.1
	SCTS	19.0	17.8	17.0	16.5	16.5
	ResNet	13.7	13.2	12.8	12.6	12.3
	ResLSTM	13.5	13.1	12.7	12.3	12.1
	MHANet	**13.4**	**12.9**	**12.5**	**12.1**	**11.9**
F16	DD	22.7	21.0	19.7	18.6	17.9
	TSNR	22.3	20.7	19.3	18.3	17.6
	HRNR	21.3	19.9	18.9	18.2	17.9
	SCTS	21.3	19.7	18.4	17.5	16.9
	ResNet	13.7	13.1	12.7	12.4	12.1
	ResLSTM	13.3	12.9	12.4	12.1	11.9
	MHANet	**13.2**	**12.7**	**12.2**	**12.0**	**11.8**
Factory	DD	24.5	22.7	21.1	19.9	18.9
	TSNR	24.2	22.5	20.9	19.6	18.7
	HRNR	23.5	22.0	20.7	19.7	19.1
	SCTS	23.0	21.2	19.8	18.6	17.8
	ResNet	14.8	14.2	13.7	13.2	12.9
	ResLSTM	**14.7**	14.0	13.4	12.9	12.5
	MHANet	**14.7**	**13.8**	**13.1**	**12.6**	**12.2**

6.3. Spectral distortion levels

The SD levels attained by the MHANet are given in Table 3. It can be seen that for both real-world non-stationary (*voice babble* and *street music*) and coloured (*F16* and *factory*) noise sources, the MHANet is able to produce lower SD levels than the previous *a priori* SNR estimation methods (DD, TSNR, HRNR, and SCTS), as well as the ResLSTM network and the ResNet (except for *factory* at -5 dB, where the ResLSTM network attained the same SD level). As shown in Nicolson and Paliwal (2020b), the high *a priori* SNR estimation accuracy attained by the MHANet will be of benefit to the employed MMSE approach.

7. Speech enhancement performance

Here, we evaluate the speech enhancement performance of the MHANet. Epoch 150 for the MHANet configuration described in Section 4 is used. It is compared to the ResLSTM network and the ResNet from Section 4. Epoch 80 and 150 were used for the ResLSTM network and the ResNet, respectively. Each DNN is employed within the Deep Xi framework to estimate the *a priori* SNR for the MMSE-LSA estimator. The MHANet is also compared to a benchmark method, namely the MMSE-LSA estimator employing the DD approach and the noise estimator from Gerkmann and Hendriks (2012).

Objective quality (MOS-LQO) and intelligibility (STOI) measures are used to evaluate the speech enhancement performance of the MHANet, where each metric is described in Figs. 6 and 7, respectively. The used test set is described in Section 7.1. The objective scores attained by the MHANet are discussed in Section 7.2. The enhanced speech spectrogram produced by the MHANet is also evaluated in Section 7.3. Subjective quality scores for the enhanced speech produced by the MHANet are presented and discussed in Sections 7.4 and 7.5. Finally, the execution time of the MHANet is evaluated in Section 7.6.

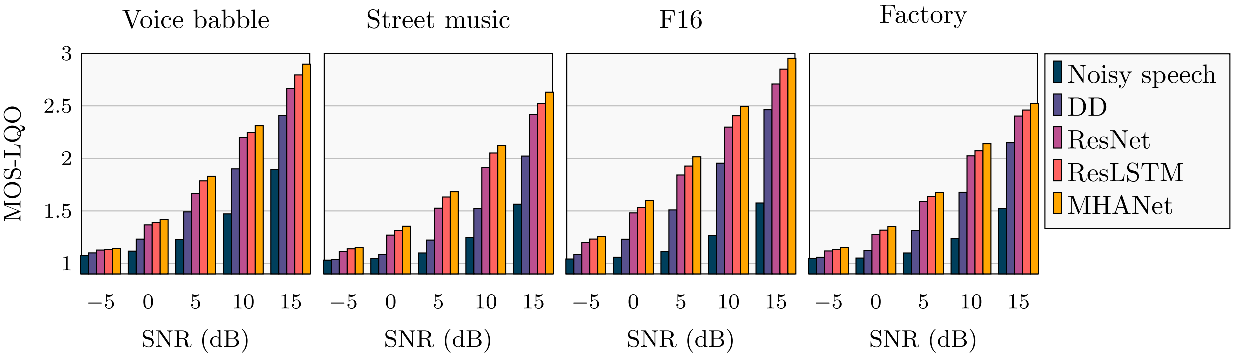

Fig. 6. Enhanced speech objective quality scores for the **MMSE-LSA estimator**. The mean opinion score of the listening quality objective (MOS-LQO) is used as the metric, where the wideband perceptual evaluation of quality (Wideband PESQ) is the objective model used to obtain the MOS-LQO score (Rec, 2005).

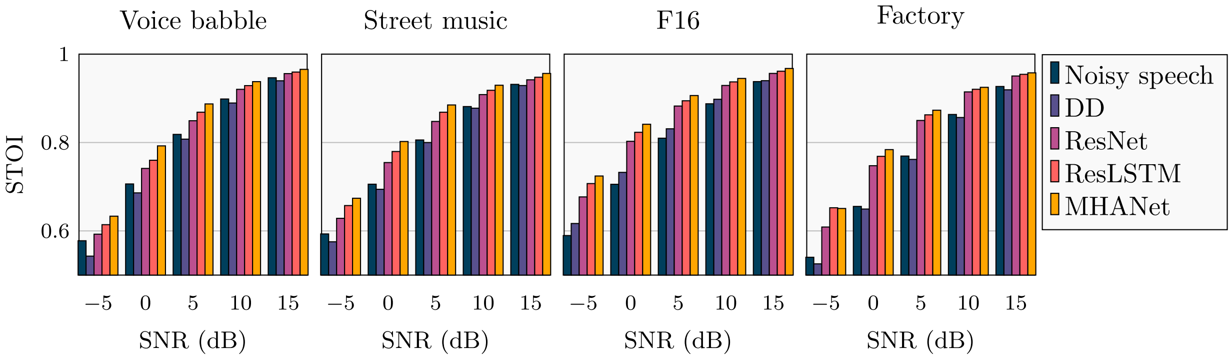

Fig. 7. Enhanced speech objective intelligibility scores (in %) for the **MMSE-LSA estimator**. The short-time objective intelligibility (STOI) measure is used to compute the objective intelligibility scores (Taal et al., 2011).

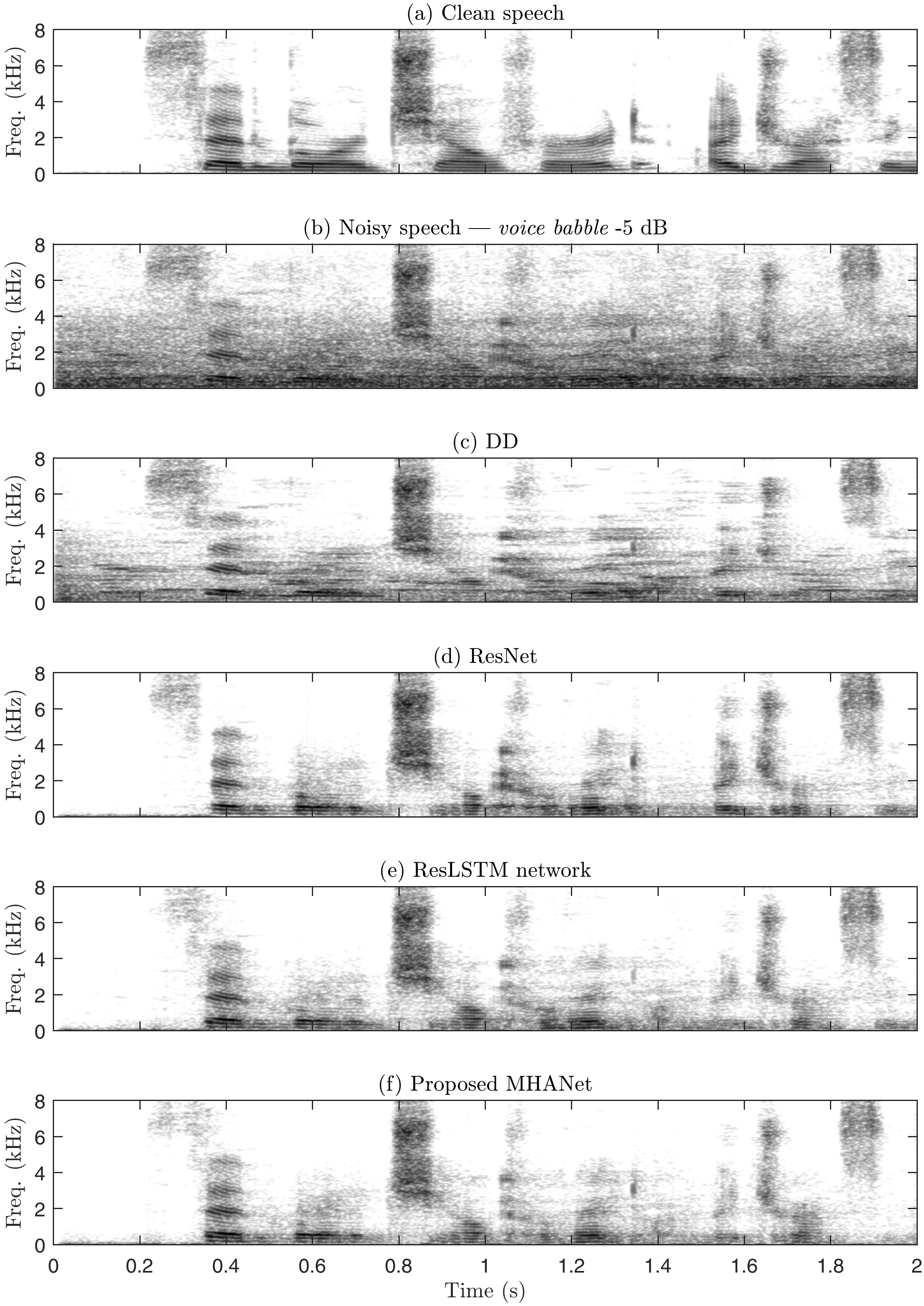

Fig. 8. (a) Clean speech spectrogram of female speaker 5 683 uttering the first two seconds of sentence 1 from book 32 865, "Said Lord Chelford, addressing me" (from the test set described in Section 7.1). (b) A recording of *voice babble* mixed with (a) at an SNR level of − 5 dB. Enhanced speech spectrogram produced by the **MMSE-LSA estimator**using (c) the DD approach, (d) the ResNet, (e) the ResLSTM network, and (f) the proposed MHANet.

7.1. Test set

The test set used in this section is identical to that described in Section 6.1, except that 10 clean speech recordings are randomly selected (without replacement) from the *test-clean* set of the Librispeech corpus for each of the four noise recordings, instead of the TSP corpus. The clean speech recordings from the *test-clean* set have a longer duration than that of the TSP corpus. The clean speech recordings from the

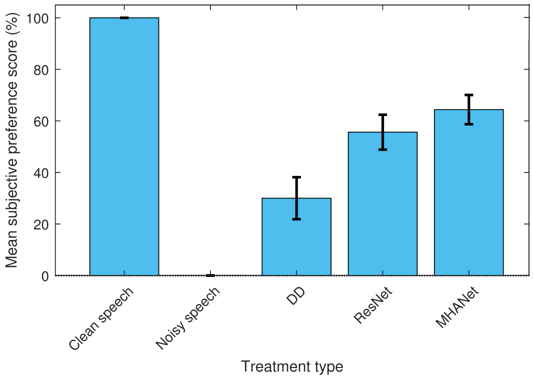

Fig. 9. Mean subjective preference (%) scores for the **MMSE-LSA estimator**. The subjective testing procedure is described in Section 7.4. *Voice babble* and *F16* at an SNR level of 0 dB are the used conditions for the subjective tests. The error bars indicate the standard deviation of the scores.

test-clean set have a duration of up to 34 s, whereas recordings from the TSP corpus have a duration of up to 4.8 s.

7.2. Objective scores

The objective quality scores attained by the MHANet are given in Fig. 6. It can be seen that the MHANet is able to produce enhanced speech at a higher quality than the ResLSTM network and the ResNet for all tested conditions, including real-world non-stationary (*voice babble* and *street music*) and coloured (*F16* and *factory*) noise sources at all SNR levels. The objective intelligibility scores attained by the MHANet are given in Fig. 7. The MHANet is able to produce more intelligible enhanced speech than the ResLSTM network and the ResNet for all conditions (except for *factory* at −5 dB, where the ResLSTM performs best). It is even able to produce enhanced speech that is significantly more intelligible than noisy speech at low SNR levels. This indicates that the MHANet is more suited to the task of speech enhancement than both the ResLSTM network and the ResNet.

7.3. Enhanced speech spectrograms

In this section we examine the enhanced speech spectrogram produced by the MHANet. The clean speech spectrogram is shown in Fig. 8 (a). *Voice babble* at an SNR level of -5 dB is used to create the noisy speech in Fig. 8 (b). This is a particularly tough condition, as the background noise exhibits characteristics similar to the speech produced by the target speaker. Moreover, the background noise is more predominant than the target speech in the band from 0 to 4 kHz.

The enhanced speech for the DD approach is shown in Fig. 8 (c). It can be seen that a significant amount of musical noise is introduced. A large amount of residual background noise and speech distortion is also present. The enhanced speech produced by the ResNet is shown in Fig. 8 (d). It can be seen that there is less residual background noise than the enhanced speech produced by the DD approach. Moreover, there is no musical noise present. The enhanced speech for the ResLSTM network is shown in Fig. 8 (e). There is less residual background noise and speech distortion present than in the enhanced speech produced by the ResNet. The enhanced speech produced by the MHANet is shown in Fig. 8(f). It can be seen that there is less residual background noise and speech distortion than in the enhanced speech produced by the ResLSTM network.

7.4. Subjective testing procedure

In this subsection, we describe the procedure used to obtain the subjective quality scores in Fig. 9. The mean subjective preference (%) scores are obtained using AB listening tests (So and Paliwal, 2011). Each AB listening test involves a stimuli pair. Each stimulus is either clean, noisy, or enhanced speech. The enhanced speech stimuli are produced by the DD approach, the ResNet, and the proposed MHANet — each using the MMSE-LSA estimator. Therefore, each stimulus belongs to one of the following classes: clean speech, noisy speech, enhanced speech produced by the DD approach, the ResNet, or the proposed MHANet.

After listening to a stimuli pair, the listeners' preference is determined by selecting one of three options. The first and second options indicate a preference for one of the two stimuli, while the third option indicates an equal preference for both stimuli. Pair-wise scoring is used, with a score of +1 awarded to the preferred class, and 0 to the other. If the listener has an equal preference for both stimuli, each class is awarded a score of +0.5. Participants could re-listen to the stimuli pair before selecting an option.

Two utterances from the test set described in Section 7.1 are used as the clean speech stimuli: sentence 1 from book 32 865, as uttered by female speaker 5 683, and sentence 7 from book 122 612, as uttered by male speaker 1 320. *Voice babble*and *F16* is mixed with the clean speech stimuli from speaker 5 683 and 1 320, respectively, at an SNR level of 0 dB, producing the noisy speech stimuli. The enhanced speech stimuli for each of the speech enhancement methods is produced from the noisy speech stimuli. For each utterance, all possible stimuli pair combinations are presented to the listener (i.e. double-blind testing). Each participant listens to a total of 40 stimuli pair combinations. A total of five English-speaking listeners participated. Each listening test is conducted in a separate session, in a quiet room using closed circumaural headphones at a comfortable listening level.

7.5. Subjective quality scores

Here, we evaluate the subjective quality of the enhanced speech produced by the MHANet. Details about the subjective testing procedure are given in Section 7.4. *Voice babble* and *F16* at an SNR level of 0 dB are the used conditions for the subjective tests. *Voice babble* is a real-world non-stationary noise source, while *F16* is a real-world coloured noise source. The mean subjective preference (%) for the MMSE-LSA estimator utilising the DD approach, the ResNet, and the MHANet is shown in Fig. 9. It can be seen that the enhanced speech produced by the MHANet is preferred by listeners over the enhanced speech produced by the ResNet. These results support the objective quality results obtained by the MHANet in Fig. 6.

7.6. Execution time comparison

In this subsection, we evaluate the execution time of the MHANet. It is tasked with processing each noisy speech signal of the test set described in Section 7.1 individually (i.e. no batch processing). The MHANet is compared to the ResLSTM network and the ResNet. Five trials for each model are performed. The total duration of the test set is 1 515.1 sand includes 200 waveforms. The ResLSTM network, ResNet, and MHANet took an average of 518.7, 36.9, and 28.6 s to process the test set on an NVIDIA TITAN X graphics processing unit (GPU). Additionally, the ResLSTM network, ResNet, and MHANet took an average of 360.1, 261.2, and 415.3 s to process the test set on 2× Intel Xeon E5-2670 v3 @ 2.30 GHz (48 total logical processors) central processing units (CPUs). It can be seen that the execution time of the MHANet is faster than that of the ResNet and the ResLSTM network for the GPU case. However, the MHANet is the slowest for the CPU case. The MHANet and the ResNet took significantly longer to process the test set when the CPU was used. Oppositely, the ResLSTM processed the test set faster when a CPU was used. Most modern mobile devices include a GPU that can be used for inference (Yun et al., 2018; Wang et al., 2020). The results for the GPU case thus indicate that the execution time of the MHANet on a modern mobile device would be faster than that of the ResLSTM network and ResNet.

Table 4
Objective scores obtained on the test set described in Section 8.2. As in previous works, the objective scores are averaged over all tested conditions. **CSIG, CBAK**, and **COVL** are mean opinion score (MOS) predictors of the signal distortion, background-noise intrusiveness, and overall signal quality, respectively Hu and Loizou (2008). **PESQ** is the perceptual evaluation of speech quality measure Rix et al. (2001). **STOI** is the short-time objective intelligibility measure (in %) Taal et al. (2011). The highest scores attained for each measure are indicated in boldface.

Method	Causal	CSIG	CBAK	COVL	PESQ	STOI
Noisy speech	–	3.35	2.44	2.63	1.97	92 (91.5)
Wiener Scalart and Filho (1996)	✓	3.23	2.68	2.67	2.22	-
SEGAN Pascual et al. (2017)	×	3.48	2.94	2.80	2.16	93
WaveNet Rethage et al. (2018)	×	3.62	3.23	2.98	-	-
HLGAN Yang et al. (2020)	×	3.65	3.19	3.05	2.48	-
MMSE-GAN Soni et al. (2018)	×	3.80	3.12	3.14	2.53	93
Deep Feature Loss Germain et al. (2019)	✓	3.86	3.33	3.22	-	-
MetricGAN Fu et al. (2019)	×	3.99	3.18	3.42	2.86	-
FNN (Deep Xi–MMSE-LSA)	✓	4.04	3.25	3.36	2.67	93 (92.6)
ResNet (Deep Xi–MMSE-LSA)	✓	4.12	3.33	3.48	2.82	93 (93.3)
Proposed MHANet (Deep Xi–MMSE-LSA)	✓	4.17	3.37	3.53	2.88	**94 (93.6)**
Koizumi2020 Koizumi et al. (2020)	×	4.15	3.42	3.57	2.99	-
T-GSA Kim et al. (2020)	×	**4.18**	**3.59**	**3.62**	**3.06**	-

8. Comparison to multiple deep learning approaches to speech enhancement

In this section, we compare the speech enhancement performance of the MHANet to multiple recent deep learning approaches to speech enhancement. The training and test sets from Valentini-Botinhao et al. (2016) are used, which have been employed previously to evaluate deep learning approaches to speech enhancement. The training set is described in Section 8.1 and the test set is described in Section 8.2. The MHANet and the ResNet from Section 4 are used here and are both trained using the training set described in Section 8.1. A FNN including five hidden layers with 1 024 nodes per layer was also trained within the Deep Xi framework using the training set described in Section 8.1. Each hidden layer of the FNN is followed by frame-wise layer normalisation and the ReLU activation function. The FNN inludes the current frame and 10 previous frames as its input. The MHANet, ResNet, and FNN are trained until convergence (for 115 epochs). MHANet is compared to the following deep learning approaches to speech enhancement: SEGAN (Pascual et al., 2017), WaveNet (Rethage et al., 2018), MMSE-GAN (Soni et al., 2018), Deep Feature Loss (Germain et al., 2019), MetricGAN (Fu et al., 2019), Koizumi2020 Koizumi et al. (2020), and T-GSA Kim et al. (2020). The objective measures used for the comparison are described in Table 4. The objective scores attained by the MHANet are discussed in Section 8.3.

8.1. Training set

The training set from Valentini-Botinhao et al. (2016) is described in this subsection. Clean speech recordings from 28 speakers of the Voice Bank corpus (Veaux et al., 2013) are included in the training set (11 572 recordings). Two synthetic noise sources (*speech-shaped noise* and *babble*, as described in Valentini-Botinhao et al. (2016), as well as eight real-world noise recordings from the DEMAND dataset (Thiemann et al., 2013) are also included in the training set. The clean speech and noise recordings are downsampled from 48 kHz to 16 kHz. Noisy speech signals are formed by mixing each clean speech recording with a random section of a randomly selected noise recording at one of four following SNR levels: 0, 5, 10, and 15 dB. This creates 11 572 noisy speech signals for training.

8.2. Test set

The test set from Valentini-Botinhao et al. (2016) is described in this subsection. The test set includes 824 clean speech recordings of two speakers from the Voice Bank corpus — 393 from *p*232 and 431 from *p*257 (Veaux et al., 2013). Both speakers are separate from those selected in the previous section for the training set. A total of 20 different conditions are used to create the noisy speech, including five noise types from the DEMAND dataset (separate from those included in the training set described in Section 8.1), and 4 SNR levels: 2.5, 7.5, 12.5, and 17.5 dB. This corresponds to approximately 20 different sentences per condition for each speaker (824 noisy speech signals in the second test set). The clean speech and noise recordings are downsampled from 48 kHz to 16 kHz prior to mixing.

8.3. Objective scores

The objective scores for the MHANet and for multiple recent deep learning approaches to speech enhancement are shown in Table 4. The objective scores from Pascual et al. (2017), Rethage et al. (2018), Yang et al. (2020), Soni et al. (2018), Germain et al. (2019), Fu et al. (2019), Koizumi et al. (2018), Kim et al. (2020) are included in the table (PESQ and STOI scores were not reported in some of these articles). Some of the deep learning approaches to speech enhancement are non-causal, such as MetricGAN, WaveNet, MMSE-GAN, Koizumi2020, and T-GSA.[2] As seen in Table 4, the MHANet outperforms all models that do not utilise an attention mechanism.

The MHANet outperforms Koizumi2020 for CSIG. However, Koizumi2020 demonstrates an improvement of 0.05, 0.04, and 0.11 for CBAK, COVL, and PESQ, respectively, over the MHANet. The key advantages that Koizumi2020 has over MHANet is the use of a complex-valued TF mask as the training target (i.e. it makes use of all the information of the noisy speech DFT coefficients), its use of speaker-aware features, and that it is non-causal. T-GSA demonstrates a performance improvement of 0.01, 0.22, 0.09, and 0.18 for CSIG, CBAK, COVL, and PESQ, respectively, when compared to the MHANet. The main advantages that T-GSA has over the MHANet is that it is non-causal and that it utilises PESQ as an optimisation objective. However, utilising PESQ as an optimisation objective may compromise its performance for other objective measures, such as CSIG. For real-time applications, the non-causal approaches will inherently exhibit a delay, and lack responsiveness. The MHANet relies only on current and previous frames, thus avoiding the drawbacks associated with non-causal approaches.

9. Conclusion

In this work, we propose the MHANet for speech enhancement. Masking is used to ensure causality, allowing the MHANet to be used in real-time systems. The MHANet was compared to a RNN and a TCN using both objective and subjective measures of quality and intelligibility. Multiple real-world non-stationary and coloured noise sources at multiple SNR levels were used as the testing conditions. The results

[2] MetricGAN utilises bidirectional LSTM cells, WaveNet employs non-causal dilated kernels, and MMSE-GAN includes past, present, and future frames as its input. SEGAN and HLGAN utilise a window duration of one second, which would also significantly affect response time. Koizumi2020 and T-GSA do not mask out attention weights that consider future frames.

presented in this work show that the MHANet is able to produce enhanced speech at a higher quality and intelligibility than that produced by RNNs and TCNs. This is because MHA is better able to model the long-term dependencies of noisy speech than RNNs and TCNs. It was also found that utilizing dropout hinders the performance of MHANet. Additionally, it was found that no positional encoding for the attention mechanism is required for speech enhancement. This indicates that a sufficient amount of positional information is encoded into the noisy speech. The attention weights of the MHANet were also analysed, where it was observed that each head attends to regions of the target speech and/or background noise. This work demonstrates that MHA is more suitable for causal speech enhancement than RNNs and TCNs.

CRediT authorship contribution statement

Aaron Nicolson: Writing - review & editing, Conceptualization, Methodology, Software, Data curation, Visualization, Investigation. **Kuldip K. Paliwal:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Hyperparameter search

In this section, a search is conducted to determine an appropriate set of hyperparameters for the MHANet. The hyperparameters that are investigated are shown in Table A.1. Performing a grid search over multiple values for each hyperparameter is costly. Instead, we conduct a hyperparameter search over two stages. The first stage is a manual search over each hyperparameter. The second stage is a grid search over a subset of the hyperparameters. This subset includes the hyperparameters that had no appropriate value discovered during the first stage of the search. These two stages are used to determine the MHANet hyperparameters for the remainder of this work. The validation error from Eq. (16) is used to determine the best value for each hyperparameter, with the assumption that a lower validation error indicates a better speech enhancement performance.

A1. Manual search

For the first stage of the hyperparameter search, a manual search is conducted. Described in Table A.1 is the set of hyperparameters used initially for the MHANet. For each manual search, the hyperparameter is varied until an appropriate value is found. Once an appropriate value for a hyperparameter is found, it is then used for the remaining manual searches. The validation error over 50 training epochs is used to determine the appropriate value for each hyperparameter.

Table Appendix A.1
Hyperparameters used for the initial MHANet. The values for Ψ and P_{drop}, as well as the positional encoding type from Vaswani et al. (2017) were used. The remaining initial hyperparameters were chosen heuristically.

Param.	Init. val.	Description
Pos. enc.	Additive	Positional encoding
B	4	# blocks
d_{ff}	256	Feed-forward size
d_{model}	128	Model size
H	4	# heads
P_{drop}	0.1	Dropout rate
Ψ	4 000	# warmup steps

A1.1. Positional encoding

We first investigate the positional encoding types, including the addition or concatenation of a positional encoding, and no positional encoding. The trigonometric positional encoding from Vaswani et al. (2017) is used. As can be seen in Fig. A.1, the lowest validation error is attained when no positional encoding is used. This indicates that information about the order of events is sufficiently embedded in the noisy speech magnitude spectrum input. Thus, no positional encoding is used for the remainder of this work.

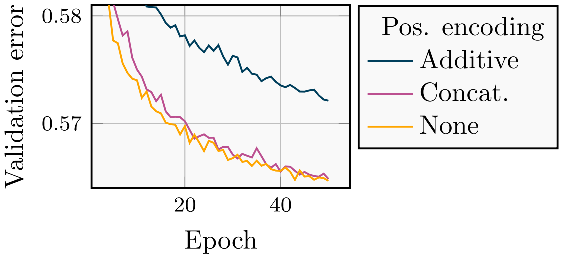

Fig. Appendix A.1. Positional encoding types. MHANet hyperparameters: $B = 4$, $d_{ff} = 256$, $d_{model} = 128$, $H = 4$, $P_{drop} = 0.1$, and $\Psi = 4\,000$.

A1.2. Number of blocks: B

The number of blocks B, for the MHANet is investigated here. The number of blocks contributes significantly to the total number of parameters of the MHANet. Selecting too many blocks can thus cause the MHANet to consume an unnecessarily large amount of parameters. As shown in Fig. A.2, a total of five to six blocks performs best. In Section A.2, the number of blocks is further investigated through a grid search. A total of five blocks are used for the remainder of this subsection.

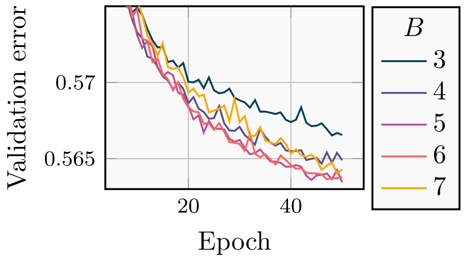

Fig. Appendix A.2. Number of MHA blocks, B. MHANet hyperparameters: $d_{ff} = 256$, $d_{model} = 128$, $H = 4$, $P_{drop} = 0.1$, $\Psi = 4\,000$, and no positional encoding.

A1.3. Feed-forward inner layer size: d_{ff}

Next, we investigate the size of the first feed-forward layer of the FNN of each block, d_{ff}. The size of d_{ff} is important, as the FNN accounts for most of the parameters in each block (Sukhbaatar et al., 2019). Blindly selecting a size that is too high can cause the model to be parameter inefficient. It can be seen in Fig. A.3 that a d_{ff} size of 1 024 attains the lowest validation error. A d_{ff} size of 1 024 is thus used for the remainder of this work.

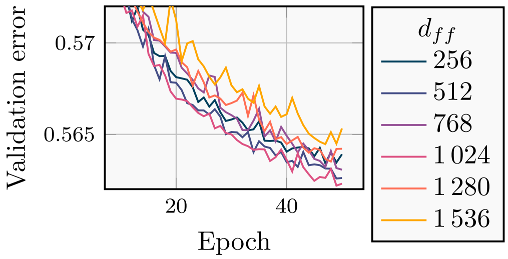

Fig. Appendix A.3. Feed-forward layer size, d_{ff}. MHANet hyperparameters: $B = 5$, $d_{model} = 128$, $H = 4$, $P_{drop} = 0.1$, $\Psi = 4\,000$, and no positional encoding.

A1.4. Model size: d_{model}

In this section, we investigate the model size for the MHANet, d_{model}. The size of d_{model} affects the input and output size of each block, each MHA module, and each FNN. It also affects the size of the first layer, the input size of the output layer, and the size of each head. This indicates that the size of d_{model} will have a significant effect on the performance of the MHANet. As shown in Fig. A.4, a d_{model} size of 128 provides the lowest validation error, with a d_{model} size of 256 demonstrating a similar performance. A d_{model} size of 512 is also tested, but is unable to converge. In Section A.2, the d_{model} size is further investigated through a grid search due to its effect on the performance of the MHANet. A d_{model} size of 128 is used for the remainder of this subsection.

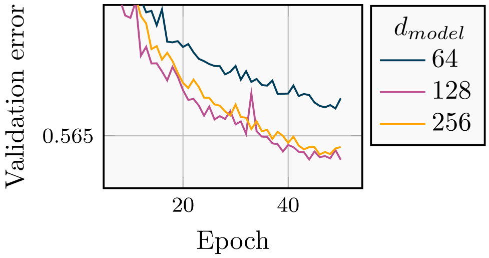

Fig. Appendix A.4. Model size, d_{model}. MHANet hyperparameters: $B = 5$, $d_{ff} = 1\,024$, $H = 4$, $P_{drop} = 0.1$, $\Psi = 4\,000$, and no positional encoding.

A1.5. Number of heads: H

Here, we investigate the number of heads for each MHA module, H. Each head possesses the ability to model different aspects of the noisy speech. For example, one head may learn to model regions of noise, while another may learn to model phonemic information. Another factor to consider is that the policy $d_k = d_v = d_{model}/H$ decreases the size of each head as H increases. It can be seen in Fig. A.5 that utilising four to eight heads attains the lowest validation error (A total of 16 heads is also investigated but consumes too much memory when trained on an NVIDIA GTX 1080 Ti). A total of four heads are used for the remainder of this subsection. The number of heads is further investigated in Section A.2 through a grid search.

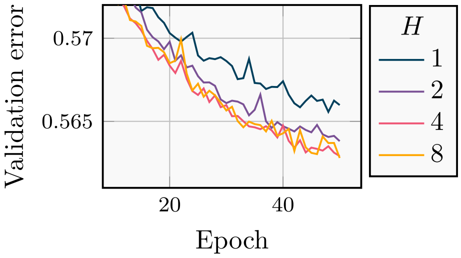

Fig. Appendix A.5. Number of heads, H. MHANet hyperparameters: $B = 5$, $d_{ff} = 1\,024$, $d_{model} = 128$, $P_{drop} = 0.1$, $\Psi = 4\,000$, and no positional encoding.

A1.6. Dropout rate: P_{drop}

Next, we investigate the dropout rate for the MHANet. As shown in Fig. A.6, utilising no dropout (i.e. $P_{drop} = 0.0$) provided the lowest validation error. Dropout tends not to be beneficial for speech enhancement as shown by its absence in other works (Nicolson and Paliwal, 2019a; Rethage et al., 2018; Fu et al., 2019). No dropout is used for the remainder of this work.

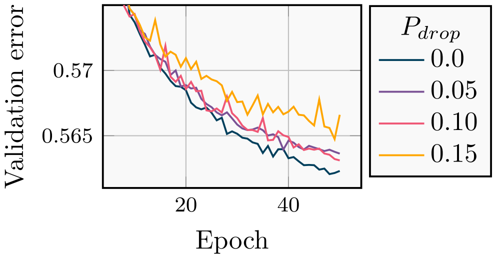

Fig. Appendix A.6. Dropout rate, P_{drop}. MHANet hyperparameters: $B = 5$, $d_{ff} = 1\,024$, $d_{model} = 128$, $H = 4$, $\Psi = 4\,000$, and no positional encoding.

A1.7. Warmup steps: Ψ

In this subsection, we investigate the number of warmup steps, Ψ. The appropriate number of warmup steps will depend largely on the task (e.g. machine translation versus speech enhancement) and the mini-batch size. It can be determined from Fig. A.7 that 40 000 warmup steps provides the lowest validation error, and is used for the remainder of this work.

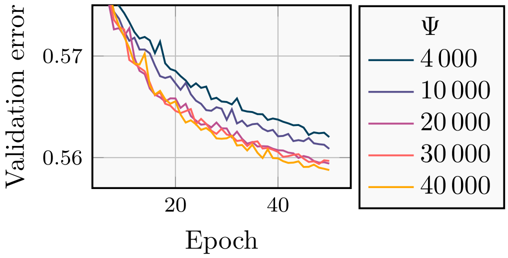

Fig. Appendix A.7. Number of warmup steps, Ψ. MHANet hyperparameters: $B = 5$, $d_{ff} = 1\,024$, $d_{model} = 128$, $H = 4$, $P_{drop} = 0.0$, and no positional encoding.

A2. Grid search

From the manual search, an appropriate value could not be determined for the block size and the number of heads. More specifically, it could not be determined if 5 or 6 blocks is more appropriate, or if 4 or 8 heads is more appropriate. Additionally, a d_{model} size of 128 and 256 are further investigated, due to the ability of d_{model} to affect many aspects of the MHANet (as described in Section A.1.4). This gives eight different combinations, as described in Table A.2. The values for the remaining hyperparameters are those found during the manual search. The validation error curves for each configuration in Table A.2 are shown in Fig. A.8. It can be seen that configuration D and H both produced the lowest validation error. Configuration D and H are similar, in that they both use $d_{ff} = 1\,024$, $d_{model} = 256$, $H = 8$, $P_{drop} = 0.0$, $\Psi = 40\,000$ and no positional encoding. The difference is that D utilises five blocks, while H utilises six blocks. Configuration D is chosen for the MHANet for the remainder of this work, as it consumes fewer parameters than configuration H.

Table Appendix A.2
MHANet configurations used for the grid search.

Model	Hyperparameter						
	Pos. enc.	B	d_{ff}	d_{model}	H	P_{drop}	Ψ
A	None	5	1 024	128	4	0.0	40 000
B	None	5	1 024	128	8	0.0	40 000
C	None	5	1 024	256	4	0.0	40 000
D	None	5	1 024	256	8	0.0	40 000
E	None	6	1 024	128	4	0.0	40 000
F	None	6	1 024	128	8	0.0	40 000
G	None	6	1 024	256	4	0.0	40 000
H	None	6	1 024	256	8	0.0	40 000

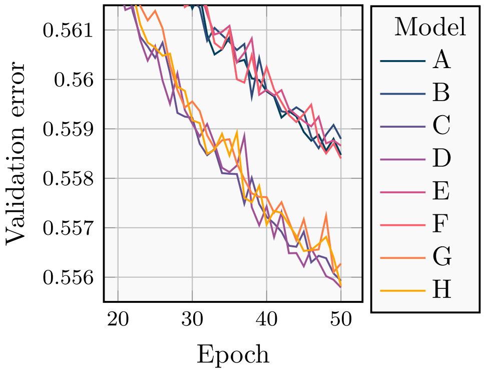

Fig. Appendix A.8. Grid search for the proposed MHANet.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Allen, J., 1977. Short term spectral analysis, synthesis, and modification by discrete Fourier transform. IEEE Trans. Acoust. 25 (3), 235–238.

Allen, J.B., Rabiner, L.R., 1977. A unified approach to short-time Fourier analysis and synthesis. Proc. IEEE 65 (11), 1558–1564.

Ba, J. L., Kiros, J. R., Hinton, G. E., 2016. Layer normalization. arXiv:1607.06450[stat.ML].

Bai, S., Kolter, J. Z., Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271[cs.LG].

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. 5 (2), 157–166.

Breithaupt, C., Gerkmann, T., Martin, R., 2008. A novel a priori SNR estimation approach based on selective cepstro-temporal smoothing. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4897–4900.

Chan, W., Lane, I., 2016. On online attention-based speech recognition and joint Mandarin character-pinyin training. Interspeech 2016, pp. 3404–3408. https://doi.org/10.21437/Interspeech.2016-334.

Chen, J., Wang, D., 2017. Long short-term memory for speaker generalization in supervised speech separation. J. Acoust. Soc. Am. 141 (6), 4705–4714. https://doi.org/10.1121/1.4986931.

Crochiere, R., 1980. A weighted overlap-add method of short-time Fourier analysis/synthesis. IEEE Trans. Acoust. 28 (1), 99–102.

Dean, D.B., Sridharan, S., Vogt, R.J., Mason, M.W., 2010. The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms. INTERSPEECH-2010, pp. 3110–3113.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.

Ephraim, Y., Malah, D., 1984. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. IEEE Trans. Acoust. 32 (6), 1109–1121.

Ephraim, Y., Malah, D., 1985. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE Trans. Acoust. 33 (2), 443–445.

Xie, F., Van Compernolle, D., 1994. A family of MLP based nonlinear spectral estimators for noise reduction. Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. 53–56.

Fu, S.-W., Liao, C.-F., Tsao, Y., Lin, S.-D., 2019. MetricGAN: Generative adversarial networks based black-box metric scores optimization for speech enhancement. In: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning. PMLR, Long Beach, California, USA, pp. 2031–2041.

Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., 1993. TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Technical Report. Philadelphia: Linguistic Data Consortium.

Gerkmann, T., Hendriks, R.C., 2012. Unbiased MMSE-based noise power estimation with low complexity and low tracking delay. IEEE Trans. Audio Speech Lang. Process. 20 (4), 1383–1393.

Germain, F.G., Chen, Q., Koltun, V., 2019. Speech denoising with deep feature losses. Proc. Interspeech 2019, pp. 2723–2727. https://doi.org/10.21437/Interspeech.2019-1924.

Gers, F.A., Schmidhuber, J., Cummins, F., 1999. Learning to forget: continual prediction with LSTM. 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), vol. 2, pp. 850–855.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Hu, G., Wang, D., 2010. A tandem algorithm for pitch estimation and voiced speech segregation. IEEE Trans. Audio Speech Lang. Process. 18 (8), 2067–2079.

Hu, Y., Loizou, P.C., 2008. Evaluation of objective quality measures for speech enhancement. IEEE Trans. Audio Speech Lang. Process. 16 (1), 229–238.

Jin, Y., Tang, C., Liu, Q., Wang, Y., 2020. Multi-head self-attention-based deep clustering for single-channel speech separation. IEEE Access 8, 100013–100021.

Kabal, P., 2002. TSP Speech Database. Technical Report. McGill University.

Kim, J., El-Khamy, M., Lee, J., 2020. T-GSA: Transformer with Gaussian-weighted self-attention for speech enhancement. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6649–6653.

Kingma, D. P., Ba, J., 2014. Adam: a method for stochastic optimization. arXiv:1412.6980[cs.LG].

Koizumi, Y., Niwa, K., Hioka, Y., Kobayashi, K., Haneda, Y., 2018. DNN-Based source enhancement to increase objective sound quality assessment score. IEEE/ACM Trans. Audio Speech Lang. Process. 26 (10), 1780–1792.

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

Kolen, J.F., Kremer, S.C., 2001. Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies. Wiley-IEEE Press.

Li, S., Li, W., Cook, C., Zhu, C., Gao, Y., 2018. Independently recurrent neural network (IndRNN): Building a longer and deeper RNN. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Liao, C.-F., Tsao, Y., Lu, X., Kawai, H., 2019. Incorporating symbolic sequential modeling for speech enhancement. Proc. Interspeech 2019, pp. 2733–2737. https://doi.org/10.21437/Interspeech.2019-1777.

Liu, M., Wang, Y., Wang, J., Wang, J., Xie, X., 2018. Speech enhancement method based on LSTM neural network for speech recognition. 2018 14th IEEE International Conference on Signal Processing (ICSP), pp. 245–249.

Loizou, P.C., 2013. Speech Enhancement: Theory and Practice, second ed. CRC Press, Inc., USA.

McGraw, I., Prabhavalkar, R., Alvarez, R., Arenas, M.G., Rao, K., Rybach, D., Alsharif, O., Sak, H., Gruenstein, A., Beaufays, F., Parada, C., 2016. Personalized speech recognition on mobile devices. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5955–5959.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. Proceedings of the 27th International Conference on Machine Learning. Omnipress, Madison, WI, USA, p. 807814.

Narayanan, A., Wang, D., 2013. Ideal ratio mask estimation using deep neural networks for robust speech recognition. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7092–7096.

Nicolson, A., Paliwal, K.K., 2018. Bidirectional long-short term memory network-based estimation of reliable spectral component locations. Proc. Interspeech 2018, pp. 1606–1610. https://doi.org/10.21437/Interspeech.2018-1134.

Nicolson, A., Paliwal, K.K., 2019. Deep learning for minimum mean-square error approaches to speech enhancement. Speech Commun. 111, 44–55. https://doi.org/10.1016/j.specom.2019.06.002.

Nicolson, A., Paliwal, K. K., 2019b. Deep Xi as a front-end for robust automatic speech recognition. arXiv:1906.07319[eess.AS].

Nicolson, A., Paliwal, K. K., 2020a. On training targets for deep learning approaches to clean speech magnitude spectrum estimation. 10.36227/techrxiv.13012760.v1.

Nicolson, A., Paliwal, K.K., 2020. Spectral distortion level resulting in a just-noticeable difference between an *a priori* signal-to-noise ratio estimate and its instantaneous case. J. Acoust. Soc. Am. 148 (4), 1879–1889. https://doi.org/10.1121/10.0002113.

Nicolson, A., Paliwal, K.K., 2020. Sum-product networks for robust automatic speaker identification. Proc. Interspeech 2020.

Panayotov, V., Chen, G., Povey, D., Khudanpur, S., 2015. Librispeech: an ASR corpus based on public domain audio books. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206–5210.

Pascual, S., Bonafonte, A., Serr, J., 2017. SEGAN: Speech enhancement generative adversarial network. Proc. Interspeech 2017, pp. 3642–3646. https://doi.org/10.21437/Interspeech.2017-1428.

Pisoni, D.B., 1993. Long-term memory in speech perception: some new findings on talker variability, speaking rate and perceptual learning. Speech Commun. 13 (1), 109–125. https://doi.org/10.1016/0167-6393(93)90063-Q.

Plapous, C., Marro, C., Mauuary, L., Scalart, P., 2004. A two-step noise reduction technique. 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 289–292.

Plapous, C., Marro, C., Scalart, P., 2005. Speech enhancement using harmonic regeneration. Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, vol. 1, pp. 157–160.

Potamitis, I., Fakotakis, N., Kokkinakis, G., 2002. Gender-dependent and speaker-dependent speech enhancement. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1.

Prabhavalkar, R., Alsharif, O., Bruguier, A., McGraw, L., 2016. On the compression of recurrent neural networks with an application to LVCSR acoustic modeling for embedded speech recognition. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5970–5974.

Rao Naidu, D.H., Srinivasan, S., 2012. Speech enhancement using emotion-dependent codebooks. IWAENC 2012; International Workshop on Acoustic Signal Enhancement, pp. 1–4.

Rec, I., 2005. P. 862.2: Wideband Extension to Recommendation P. 862 for the Assessment of Wideband Telephone Networks and Speech Codecs. Technical Report. International Telecommunication Union, CH–Geneva.

Rethage, D., Pons, J., Serra, X., 2018. A WaveNet for speech denoising. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5069–5073.

Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P., 2001. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221), vol. 2, pp. 749–752 vol.2.

Saki, F., Sehgal, A., Panahi, I., Kehtarnavaz, N., 2016. Smartphone-based real-time classification of noise signals using subband features and random forest classifier. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2204–2208.

Salamon, J., Jacoby, C., Bello, J.P., 2014. A dataset and taxonomy for urban sound research. Proceedings of the 22nd ACM International Conference on Multimedia. Association for Computing Machinery, New York, NY, USA, p. 10411044. https://doi.org/10.1145/2647868.2655045.

Scalart, P., Filho, J.V., 1996. Speech enhancement based on a priori signal to noise estimation. 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, vol. 2, pp. 629–632.

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 45 (11), 2673–2681.

Snyder, D., Chen, G., Povey, D., 2015. MUSAN: A music, speech, and noise corpus. arXiv:1510.08484[cs.SD].

So, S., Paliwal, K.K., 2011. Modulation-domain Kalman filtering for single-channel speech enhancement. Speech Commun. 53 (6), 818–829. https://doi.org/10.1016/j.specom.2011.02.001.

Soni, M.H., Shah, N., Patil, H.A., 2018. Time-frequency masking-based speech enhancement using generative adversarial network. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5039–5043.

Sperber, M., Niehues, J., Neubig, G., Stõker, S., Waibel, A., 2018. Self-attentional acoustic models. Proc. Interspeech 2018, pp. 3723–3727. https://doi.org/10.21437/Interspeech.2018-1910.

Steeneken, H.J., Geurtsen, F.W., 1988. Description of the RSG-10 Noise Database. Report IZF 1988-3. TNO Institute for Perception, Soesterberg, The Netherlands.

Subramanian, A.S., Chen, S.-J., Watanabe, S., 2018. Student-teacher learning for BLSTM mask-based speech enhancement. Proc. Interspeech 2018, pp. 3249–3253. https://doi.org/10.21437/Interspeech.2018-2440.

Sukhbaatar, S., Grave, E., Lample, G., Jegou, H., Joulin, A., 2019. Augmenting self-attention with persistent memory. arXiv:1907.01470[cs.LG].

Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J., 2011. An algorithm for intelligibility prediction of time-frequency weighted noisy speech. IEEE Trans. Audio Speech Lang. Process. 19 (7), 2125–2136.

Tamura, S., 1989. An analysis of a noise reduction neural network. International Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp. 2001–2004.

Tamura, S., Waibel, A., 1988. Noise reduction using connectionist models. ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 553–556.

Thiemann, J., Ito, N., Vincent, E., 2013. The diverse environments multi-channel acoustic noise database (DEMAND): a database of multichannel environmental noise recordings. Proc. Mtgs. Acoust. 19 (1), 035081. https://doi.org/10.1121/1.4799597.

Tu, Y., Du, J., Lee, C., 2019. Speech enhancement based on teacher-student deep learning using improved speech presence probability for noise-robust speech recognition. IEEE/ACM Trans. Audio Speech Lang. Process. 27 (12), 2080–2091.

Valentini-Botinhao, C., Wang, X., Takaki, S., Yamagishi, J., 2016. Investigating RNN-based speech enhancement methods for noise-robust text-to-speech. 9th ISCA Speech Synthesis Workshop, pp. 146–152. https://doi.org/10.21437/SSW.2016-24.

Vary, P., Martin, R., 2006. Digital Speech Transmission: Enhancement, Coding And Error Concealment. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 5998–6008

Veaux, C., Yamagishi, J., King, S., 2013. The voice bank corpus: design, collection and data analysis of a large regional accent speech database. 2013 International Conference Oriental COCOSDA held jointly with 2013 Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE), pp. 1–4.

Veaux, C., Yamagishi, J., MacDonald, K., 2017. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit. Technical Report. University of Edinburgh, The Centre for Speech Technology Research (CSTR).

Wang, S., Pathania, A., Mitra, T., 2020. Neural network inference on mobile socs. IEEE Des. Test.1–1

Wang, Y., Narayanan, A., Wang, D., 2014. On training targets for supervised speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. 22 (12), 1849–1858.

Wang, Z., Poon, J., Poon, S., 2019. AMI-Net+: A novel multi-instance neural network for medical diagnosis from incomplete and imbalanced data. arXiv:1907.01734[cs.LG].

Xu, Y., Du, J., Dai, L., Lee, C., 2014. An experimental study on speech enhancement based on deep neural networks. IEEE Signal Process. Lett. 21 (1), 65–68.

Xu, Y., Du, J., Dai, L., Lee, C., 2015. A regression approach to speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (1), 7–19.

Yang, F., Wang, Z., Li, J., Xia, R., Yan, Y., 2020. Improving generative adversarial networks for speech enhancement through regularization of latent representations. Speech Commun. 118, 1–9. https://doi.org/10.1016/j.specom.2020.02.001.

Yun, J., Lee, J., Kim, C.G., Lim, Y., Nah, J., Kim, Y., Park, W., 2018. A novel performance prediction model for mobile GPUs. IEEE Access 6, 16235–16245.

Zhang, Q., Nicolson, A., Wang, M., Paliwal, K.K., Wang, C., 2020. DeepMMSE: a deep learning approach to MMSE-based noise power spectral density estimation. IEEE/ACM Trans. Audio Speech Lang. Process. 28, 1404–1415.

Zhao, Z., Duan, H., Min, G., Wu, Y., Huang, Z., Zhuang, X., Xi, H., Fu, M., 2019. A lighten CNN-LSTM model for speaker verification on embedded devices. Future Gener Comput Syst 100, 751–758. https://doi.org/10.1016/j.future.2019.05.057.