Speech Communication 142 (2022) 49–60

Contents lists available at ScienceDirect

Speech Communication

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

On supervised LPC estimation training targets for augmented Kalman filter-based speech enhancement

Sujan Kumar Roy [a,*], Aaron Nicolson [b], Kuldip K. Paliwal [a]

[a] Signal Processing Laboratory, Griffith University, Nathan 4111, QLD, Australia
[b] Australian e-Health Research Centre, Commonwealth Scientific and Industrial Research Organisation, Herston 4006, QLD, Australia

ARTICLE INFO

Keywords:
Speech enhancement
Augmented Kalman filter
Linear predication coefficients
Training targets
Temporal convolutional network
Multi-head attention network

ABSTRACT

The performance of speech coding, speech recognition, and speech enhancement systems that rely on the augmented Kalman filter (AKF) largely depend upon the accuracy of clean speech and noise linear prediction coefficient (LPC) estimation. The formulation of clean speech and noise LPC estimation as a supervised learning task has shown considerable promise as of late. Generally, a deep neural network (DNN) learns to map noisy speech features to a training target that can be used for clean speech and noise LPC estimation. Such training targets fall into four categories: Line spectrum frequency (LSF), LPC power spectrum (LPC-PS), power spectrum (PS), and magnitude spectrum (MS) training targets. The choice of training target can have a significant impact on LPC estimation accuracy. Motivated by this, we perform a comprehensive study of the training targets with the aim of determining which is best for LPC estimation. To this end, we evaluate each training target using a temporal convolutional network (TCN) and a multi-head attention-based network. A large training set constructed from a wide variety of conditions, including real-world non-stationary and coloured noise sources over a range of signal-to-noise ratio (SNR) levels, is used for training. Testing on the NOIZEUS corpus demonstrates that the LPC-PS as the training target produces the lowest clean speech LPC spectral distortion (SD) level. We also construct the augmented Kalman filter (AKF) with the estimated speech and noise LPC parameters of each training target. Subjective AB listening tests and seven objective quality and intelligibility evaluation measures (CSIG, CBAK, COVL, PESQ, STOI, SegSNR, and SI-SDR) revealed that the LPC-PS training target produced enhanced speech at the highest quality and intelligibility amongst the training targets.

1. Introduction

Speech processing applications, such as low-bit rate audio coding, speech enhancement, and speech recognition, rely upon the accuracy of linear prediction coefficient (LPC) estimates of clean speech and noise in practice (Vaseghi, 2006, Chapter 8). For example, inaccurate clean speech and noise LPC estimates impact the quality and intelligibility of enhanced speech produced by an augmented Kalman filter (AKF) (Gibson et al., 1991). To address this, deep learning has been employed to accurately estimate LPCs for the Kalman filter (KF) and AKF. This paper focuses on training targets for supervised LPC estimation for AKF-based speech enhancement.

Paliwal and Basu (1987) introduced the KF for speech enhancement. For the KF, each clean speech frame is represented by an auto-regressive (AR) process, whose parameters include the clean speech LPCs and prediction error variance. The LPC parameters and additive noise variance are used to construct the KF recursive equations. Given a frame of noisy speech samples, the KF gives a linear MMSE estimate of the clean speech samples using the recursive equations. Paliwal and Basu (1987) demonstrated that the inaccurate estimates of the LPC parameters and noise variance result in poor quality and intelligibility in the enhanced speech produced by the KF. Later on, Gibson et al. (1991) introduced an AKF for speech enhancement in coloured noise conditions. For the AKF, both the clean speech and additive noise are represented by two AR processes. The speech and noise LPC parameters form an augmented matrix which is used to construct the recursive equations of the AKF. In this speech enhancement algorithm (SEA), the AKF processes the noisy speech iteratively (typically three to four iterations) to eliminate the coloured background noise, yielding the enhanced speech. During this, the LPC parameters for the current frame are computed from the corresponding filtered speech frame of the previous iteration (Gibson et al., 1991). Although iteratively estimating the clean speech and noise LPCs for the AKF improves the signal-to-noise ratio (SNR) of noisy speech, the resultant enhanced speech suffers from *musical noise* and *speech distortion*.

* Corresponding author.
E-mail addresses: sujankumar.roy@griffithuni.edu.au (S.K. Roy), aaron.nicolson@csiro.au (A. Nicolson), k.paliwal@griffith.edu.au (K.K. Paliwal).

https://doi.org/10.1016/j.specom.2022.06.004
Received 12 April 2021; Received in revised form 18 June 2022; Accepted 20 June 2022
Available online 25 June 2022

Multiple training targets have been investigated for deep learning approaches to speech enhancement. Time–frequency (TF) representations were the first training targets investigated for speech enhancement (Wang and Wang, 2013; Williamson et al., 2016). One example is the ideal binary mask (IBM), whose estimate is applied to the noisy speech magnitude spectrum to completely suppress the noise dominant TF components (Wang and Wang, 2013). Xu et al. (2014) applied a feed-forward neural network (FNN) to map the noisy speech log-power spectrum (LPS) to the clean speech LPS. Han et al. (2015) trained a FNN to learn a mapping from the noisy speech magnitude spectrum (MS) to the clean speech MS. Deep learning has also been investigated for statistical filter-based methods, such as MMSE short-time spectral amplitude estimators (Nicolson and Paliwal, 2019), the KF (Yu et al., 2019), and the AKF (Yu et al., 2020). Recently, Nicolson and Paliwal (2021) demonstrated that the choice of training target for clean speech MS estimation has a significant impact on speech enhancement. It was shown that using the *a priori* SNR as the training target produced the highest quality enhanced speech, whilst using the gain of an MMSE estimator or the ideal amplitude mask (IAM) produced the highest intelligibility speech and was most suited as a front-end for robust ASR.

It is found in literature that the baseline DNN-based MS or LPS estimators (Xu et al., 2014; Wang et al., 2014) and TF masked-based SEAs (Wang and Wang, 2013; Williamson et al., 2016) have been shown significant speech enhancement performance than classical SEAs (Boll, 1979; Kamath and Loizou, 2002; Ephraim and Malah, 1984; Ephraim and Malah, 1985). In general, the baseline DNN methods estimate the MS or LPS of clean speech, then the time-domain clean speech reconstruction is performed with the noisy speech phase — which has a significant impact on the quality of enhanced speech as addressed in Paliwal et al. (2011). In addition, the DNN-based SEAs applied a compression function to the MS or LPS of clean speech (in decibel) to form a mapped training target. However, a compression function may over compress values above 0 dB, which correspond to clean speech formants and a very small proportion of the overall distribution as specified in Nicolson and Paliwal (2021). MS training targets were also found to perform the best at lower SNR levels (−5 and 0 dB). It may result due to the distribution of the clean speech MS does not change with the SNR (Nicolson and Paliwal, 2021). On the other hand, the DNN assisted KF and AKF methods (Roy et al., 2020a,b) address speech enhancement in the time-domain — which is not impacted by the noisy speech phase. Recently, Deep learning assisted KF and AKF methods (Roy et al., 2020a,b) has been shown to outperform that of MS spectral amplitude estimator (Xu et al., 2014; Wang et al., 2014). As a result, deep learning-assisted KF and AKF methods have been found interesting to the researchers in the literature. The next section discusses some state-of-the-art deep learning-assisted KF and AKF methods.

1.1. Related work

Deep learning has been investigated for LPC estimation — a key parameter for the KF and AKF-based SEAs (Paliwal and Basu, 1987; Gibson et al., 1991). Pickersgill et al. (2018) proposed a deep neural network (DNN)-based LPC estimation method, termed DNN-LPC. In this method, a FNN learns a mapping from each frame of the noisy speech LPS to the log-LPC power spectra of the clean speech. During inference, the estimated log-LPC PS is converted to the LPC-PS, which is followed by an inverse Fourier transform giving the autocorrelation matrix. Next, the Yule–Walker equations are constructed with the estimated autocorrelation matrix, which is solved by the Levinson–Durbin recursion yielding the LPC parameters of the clean speech (Vaseghi, 2006, Section 8.2.2). However, there were methodology limitations, for one, spectral distortion (SD) levels were not reported below 10 dB. Moreover, only six noise recordings were used for training the FNN, indicating that it would struggle to generalise to unobserved conditions.

Yu et al. (2019) proposed a deep learning assisted KF for speech enhancement (FNN-KF). A three-layered FNN was employed to learn a mapping from the noisy speech line spectrum frequencies (LSFs) to the clean speech LSFs (12th order) (Itakura, 1975). The additive noise variance for the KF is computed from the first noisy speech frame with the assumption that the noise is non-stationary and that there is no speech present in the first frame. However, these assumptions do not account for non-stationary noise sources that have time-varying amplitudes. Moreover, the conditions observed by the FNN during training were derived from only four noise recordings and four SNR levels, indicating that it would struggle to generalise to unobserved conditions.

Yu et al. (2020) used a FNN and an long short-term memory (LSTM) network to estimate the clean speech and noise LPCs for coloured KF-based speech enhancement (FNN-CKFS and LSTM-CKFS). The FNN and LSTM network learn a mapping from the noisy speech LSFs to the clean speech and noise LSFs. During inference, the estimated LSFs are converted to the clean speech and noise LPCs. A maximum likelihood (ML) approach (Srinivasan et al., 2006) is employed to estimate the prediction error variances of the speech and noise AR processes. However, FNN-CKFS and LSTM-CKFS demonstrate poor clean speech and noise LPC estimation accuracy in unobserved noise conditions — leading to the use of multi-band spectral subtraction (MB-SS) (Kamath and Loizou, 2002) for post-processing. This could be due to training the FNN and LSTM network with a small dataset (Yu et al., 2019).

Roy et al. (2020a) utilised the DeepMMSE framework (Zhang et al., 2020) to estimate the parameters of the KF for speech enhancement (denoted as Deep Xi-KF, since DeepMMSE uses Deep Xi (Nicolson and Paliwal, 2019)). DeepMMSE utilises a residual network temporal convolutional network (ResNet-TCN) (He et al., 2016; Bai et al., 2018) to estimate the *a priori* SNR for the MMSE-based noise power spectral density (PSD) estimator. The noise variance for the KF is computed from the noise PSD estimated by DeepMMSE. Roy et al. (2020b) later used DeepMMSE to estimate the noise LPCs for the AKF (Deep Xi-AKF). Roy and Paliwal (2020a) proposed a causal convolutional encoder–decoder (CCED)-based AKF for speech enhancement (denoted as CCED-AKF). In this method, the CCED maps each frame of the noisy speech magnitude spectrum (MS) to the noise magnitude spectrum, from where the noise PSD is computed. Roy and Paliwal (2020b) proposed a residual network (ResNet) assisted AKF for speech enhancement (denoted as ResNet-AKF). This differed by mapping the time-domain samples of a given noisy speech frame to the corresponding noise frame. For Deep Xi-KF, Deep Xi-AKF, CCED-AKF, and ResNet-AKF, a whitening filter is utilised to estimate the clean speech LPCs. The coefficients for the whitening filter are computed from the estimated noise. Each noisy speech frame is then pre-whitened prior to computing the clean speech LPC parameters. However, Roy et al. (2021a) demonstrated that clean speech LPCs estimated in this manner exhibit a high amount of bias.

In order to reduce the amount of bias caused by using the whitening filter, Roy et al. (2021a) proposed the DeepLPC framework, which jointly estimates the clean speech and noise LPC-PS using a ResNet-TCN. During inference, the clean speech and noise LPCs are computed from the corresponding LPC-PS estimates. The DeepLPC produces clean speech LPC estimates with a lower SD level than the aforementioned methods, resulting in higher quality and intelligibility enhanced speech. Recently, Nicolson and Paliwal (2020) demonstrated that the multi-head attention network (MHANet) is able to outperform the ResNet-TCN in terms of speech enhancement performance, citing that the MHANet is better able to model the long-term dependencies of noisy speech. Motivated by this, Roy et al. (2021b) proposed an extension of the DeepLPC framework by replacing ResNet-TCN with MHANet, called DeepLPC-MHANet, to further improve the clean speech and noise LPC estimates for the AKF. DeepLPC-MHANet demonstrates a lower clean speech LPC estimate SD level than DeepLPC-ResNet-TCN in various noise conditions. In addition, the AKF constructed with the clean speech and noise LPC estimates

of DeepLPC-MHANet (DeepLPC-MHANet-AKF) produces higher quality and intelligible enhanced speech than DeepLPC-ResNet-TCN-AKF (Roy et al., 2021a).

This study aims to perform a comprehensive study comparing the LSF, LPC-PS, power spectrum (PS), and magnitude spectrum (MS) training targets for AKF-based speech enhancement. The motivation of this study is to determine which training target produces the most accurate clean speech and noise LPC estimates, as well as which produces AKF-based enhanced speech with the highest quality and intelligibility. Each training target is evaluated using ResNet-TCN and MHANet, where a large training set consisting of a wide variety of conditions is used for training (Roy et al., 2021b). The used test set is the NOIZEUS dataset, which consists of real-world non-stationary and coloured noise conditions over a wide range of SNR levels. We compare the SD level of the clean speech LPC estimates for each training target. We also evaluate the AKF-based speech enhancement performance of each training target using subjective AB listening tests and seven different objective quality and intelligibility measures (CSIG, CBAK, COVL, PESQ, STOI, SegSNR, and SI-SDR).

The structure of this paper is as follows: background knowledge is presented in Section 2, including the signal model and the AKF for speech enhancement. In Section 3, we present the training targets. Following this, Section 4 describes the experimental setup. The experimental results are then presented in Section 5. Finally, Section 6 gives some concluding remarks.

2. Background

2.1. Signal model

The noisy speech $y(n)$, at discrete-time sample n, is assumed to be given by

$$y(n) = s(n) + v(n), \tag{1}$$

where $s(n)$ is the clean speech and $v(n)$ is uncorrelated additive coloured noise. A 32 ms rectangular window with 50% overlap is used to convert $y(n)$ into frames, denoted by $y(n,l)$:

$$y(n,l) = s(n,l) + v(n,l), \tag{2}$$

where $l\epsilon\{0,1,\dots,L-1\}$ is the frame index, L is the total number of frames, and N is the total number of samples within each frame, i.e. $n\epsilon\{0,1,\dots,N-1\}$.

2.2. AKF for speech enhancement

For simplicity, the frame index is omitted in this Section. Each frame of the clean speech and noise signal in Eq. (2) can be represented with pth and qth order AR models, as in Vaseghi (2006, Chapter 8):

$$s(n) = -\sum_{i=1}^{p} a_i s(n-i) + w(n), \tag{3}$$

$$v(n) = -\sum_{k=1}^{q} b_k v(n-k) + u(n), \tag{4}$$

where $\{a_i; i = 1,2,\dots,p\}$ and $\{b_k; k = 1,2,\dots,q\}$ are the LPCs. $w(n)$ and $u(n)$ are assumed to be white noise with zero mean and variances σ_w^2 and σ_u^2, respectively.

Eqs. (2)–(4) are used to form the following augmented state-space model (ASSM) of the AKF, as in Gibson et al. (1991):

$$\boldsymbol{x}(n) = \boldsymbol{\Phi}\boldsymbol{x}(n-1) + \boldsymbol{r}\boldsymbol{g}(n), \tag{5}$$

$$y(n) = \boldsymbol{c}^{\top}\boldsymbol{x}(n). \tag{6}$$

In the above ASSM,

1. $\boldsymbol{x}(n) = [s(n) \ \dots s(n-p+1)\, v(n) \ \dots \ v(n-q+1)]^T$ is a $(p+q)\times 1$ state-vector,
2. $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_s & 0 \\ 0 & \boldsymbol{\Phi}_v \end{bmatrix}$ is a $(p+q)\times(p+q)$ state-transition matrix with

$$\boldsymbol{\Phi}_s = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{p-1} & -a_p \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \tag{7}$$

$$\boldsymbol{\Phi}_v = \begin{bmatrix} -b_1 & -b_2 & \dots & -b_{q-1} & -b_q \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}, \tag{8}$$

3. $\boldsymbol{r} = \begin{bmatrix} \boldsymbol{r}_s & 0 \\ 0 & \boldsymbol{r}_v \end{bmatrix}$, where $\boldsymbol{r}_s = [1\ 0\ \dots\ 0]^{\top}$, $\boldsymbol{r}_v = [1\quad 0\quad \dots\quad 0]^{\top}$,
4. $$\boldsymbol{g}(n) = \begin{bmatrix} w(n) \\ u(n) \end{bmatrix}, \tag{9}$$

5. $\boldsymbol{c}^{\top} = [\boldsymbol{c}_s^{\top} \quad \boldsymbol{c}_v^{\top}]$, where $\boldsymbol{c}_s = [1\quad 0\quad \dots\quad 0]^{\top}$ and $\boldsymbol{c}_v = [1\quad 0\quad \dots\quad 0]^{\top}$ are $p\times 1$ and $q\times 1$ vectors,
6. $y(n)$ is the noisy measurement at sample n.

For each frame, the AKF computes an unbiased linear MMSE estimate $\hat{\boldsymbol{x}}(n|n)$ at sample n, given $y(n)$, by using the following recursive equations (Gibson et al., 1991):

$$\hat{\boldsymbol{x}}(n|n-1) = \boldsymbol{\Phi}\hat{\boldsymbol{x}}(n-1|n-1), \tag{10}$$

$$\boldsymbol{\Psi}(n|n-1) = \boldsymbol{\Phi}\boldsymbol{\Psi}(n-1|n-1)\boldsymbol{\Phi}^{\top} + \boldsymbol{Q}\boldsymbol{r}\boldsymbol{r}^{\top}, \tag{11}$$

$$\boldsymbol{K}(n) = \boldsymbol{\Psi}(n|n-1)\boldsymbol{c}(\boldsymbol{c}^{\top}\boldsymbol{\Psi}(n|n-1)\boldsymbol{c})^{-1}, \tag{12}$$

$$\hat{\boldsymbol{x}}(n|n) = \hat{\boldsymbol{x}}(n|n-1) + \boldsymbol{K}(n)[y(n) - \boldsymbol{c}^{\top}\hat{\boldsymbol{x}}(n|n-1)], \tag{13}$$

$$\boldsymbol{\Psi}(n|n) = [\boldsymbol{I} - \boldsymbol{K}(n)\boldsymbol{c}^{\top}]\boldsymbol{\Psi}(n|n-1), \tag{14}$$

where $\boldsymbol{Q} = \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_u^2 \end{bmatrix}$ is the process noise covariance.

For a noisy speech frame, the error covariances ($\boldsymbol{\Psi}(n|n-1)$ and $\boldsymbol{\Psi}(n|n)$ corresponding to $\hat{\boldsymbol{x}}(n|n-1)$ and $\hat{\boldsymbol{x}}(n|n)$) and the Kalman gain $\boldsymbol{K}(n)$ are continually updated on a sample-by-sample basis, while ($\{a_i\}$, σ_w^2) and ($\{b_k\}$, σ_u^2) remain constant. At sample n, $\boldsymbol{h}^{\top}\hat{\boldsymbol{x}}(n|n)$ gives the output of the AKF, $\hat{s}(n|n)$, where $\boldsymbol{h} = [1\quad 0\quad 0\quad \dots\quad 0]^{\top}$ is a $(p+q)\times 1$ column vector. As demonstrated in AKF-RMBT (George et al., 2018), $\hat{s}(n|n)$ is given by

$$\hat{s}(n|n) = [1 - K_0(n)]\hat{s}(n|n-1) + K_0(n)[y(n) - \hat{v}(n|n-1)], \tag{15}$$

where $K_0(n)$ is the 1st component of $\boldsymbol{K}(n)$, given by

$$K_0(n) = \frac{\alpha^2(n) + \sigma_w^2}{\alpha^2(n) + \sigma_w^2 + \beta^2(n) + \sigma_u^2}, \tag{16}$$

where

$$\alpha^2(n) = \boldsymbol{c}_s^{\top}\boldsymbol{\Phi}_s\boldsymbol{\Psi}_s(n-1|n-1)\boldsymbol{\Phi}_s^{\top}\boldsymbol{c}_s, \tag{17}$$

and

$$\beta^2(n) = \boldsymbol{c}_v^{\top}\boldsymbol{\Phi}_v\boldsymbol{\Psi}_v(n-1|n-1)\boldsymbol{\Phi}_v^{\top}\boldsymbol{c}_v, \tag{18}$$

are the transmission of *a posteriori* error variances of the clean speech and noise augmented dynamic model from the previous sample, $n-1$, respectively (George et al., 2018).

Eq. (15) reveals that $K_0(n)$ has a significant impact on $\hat{s}(n|n)$. In practice, the inaccurate estimates of ($\{a_i\}$, σ_w^2) and ($\{b_k\}$, σ_u^2) introduce bias into $K_0(n)$, which impacts $\hat{s}(n|n)$. In this paper, we determine which training target for supervised learning is best for ($\{a_i\}$, σ_w^2) and ($\{b_k\}$, σ_u^2) estimation. We also investigate a new training target with the aim of outperforming all previous training targets in terms of ($\{a_i\}$, σ_w^2) and ($\{b_k\}$, σ_u^2) estimation accuracy.

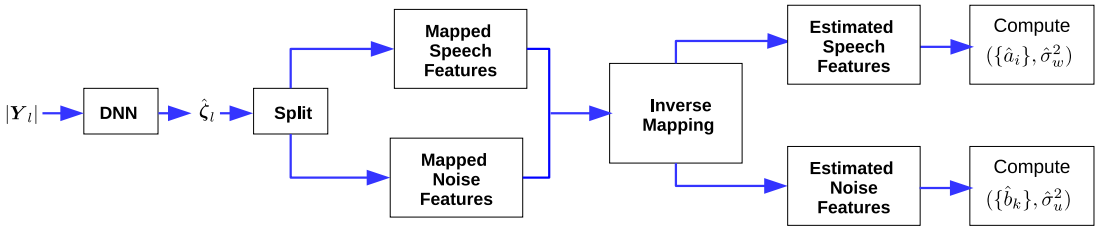

Fig. 1. (Colour online) Supervised LPC estimation framework.

3. Training targets for LPC estimation

The supervised LPC estimation framework is shown in Fig. 1. It can be seen that the framework is fed as input the single-sided noisy speech magnitude spectrum, $|\boldsymbol{Y}_l| \in \mathbb{R}^M$, where $M = 257$ and $|\boldsymbol{Y}_l| = \{|Y(l,0)|, |Y(l,1)|, \ldots, |Y(l,M-1)|\}$, i.e., $|\boldsymbol{Y}| \in \mathbb{R}^{L\times M}$. $Y(l,m)$ is computed from the noisy speech in Eq. (1) using the short-time Fourier transform (STFT):

$$Y(l,m) = S(l,m) + V(l,m), \tag{19}$$

where $Y(l,m)$, $S(l,m)$, and $V(l,m)$ denote the complex-valued STFT coefficients of the noisy speech, clean speech, and noise, respectively, for time-frame index l and discrete-frequency bin m. The Hamming window is used for analysis and synthesis. In this framework, a DNN learns a mapping from $|\boldsymbol{Y}_l|$ to the clean speech and noise training targets whose concatenated form is denoted as $\hat{\boldsymbol{\zeta}}_l$.

In this study, the LSFs, LPC-PS, PS, and MS (Sections 3.1–3.4) of the clean speech and noise are used as training targets for the DNN. A compression function is typically applied to a training target to compress its dynamic range to obtain convergence during stochastic gradient descent. During inference, we split $\hat{\boldsymbol{\zeta}}_l$ into the clean speech and noise estimates of the training targets, apply the inverse mapping of the compression function, which yields the estimates of the uncompressed training targets. From this, the clean speech and noise LPCs – $(\{\hat{a}_i\}, \hat{\sigma}_w^2)$ and $(\{\hat{b}_k\}, \hat{\sigma}_u^2)$ – are then computed. Following Roy et al. (2021a), a clean speech and noise LPC order of $p = 16$ and $q = 16$ is used, respectively.

3.1. LSF training target

The LSFs of the clean speech and noise (denoted as $\{\rho_i\}$ and $\{\eta_k\}$) are used as a training targets for LPC estimation. First, the clean speech and noise LPC parameters, $(\{a_i\}, \sigma_w^2)$ $(p = 16)$ and $(\{b_k\}, \sigma_u^2)$ $(q = 16)$ are computed from $s(n,l)$ and $v(n,l)$ using the autocorrelation method as in Vaseghi (2006, Chapter 8). Next, $\{\rho_i\}$ and $\{\eta_k\}$ are computed from $\{a_i\}$ and $\{b_k\}$.

Following this, the LPCs are converted to LSFs, which we briefly describe (McLoughlin, 2008). Each time-domain sample $s(n,l)$ under the linear prediction analysis model can be generated as the output of a finite impulse response filter, $A(z)$. Thus, the clean speech LPCs $\{a_i\}$ computed from $s(n,l)$ are used to generate $A(z)$, as in McLoughlin (2008):

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}. \tag{20}$$

To compute LSFs, $A(z)$ is decomposed into both symmetrical and anti-symmetrical parts, represented by the polynomials, $P(z)$ and $Q(z)$, as in McLoughlin (2008):

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}), \tag{21}$$

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}). \tag{22}$$

The clean speech LSFs $\{\rho_i\}$ are expressed as the zeros (or complex roots denoted by $\{\theta_i\}$) of $P(z)$ and $Q(z)$ in terms of angular frequency. Then $\{\rho_i\}$ are computed as (McLoughlin, 2008):

$$\{\rho_i\} = \tan^{-1}\left(\frac{Re\{\theta_i\}}{Im\{\theta_i\}}\right), \quad i = 1, 2, \ldots, p, \tag{23}$$

where $\{\rho_i\}$ are expressed in radians (between $[0, \pi]$). Using Eqs. (20)–(23), the noise LSFs, $\{\eta_k\}$ are computed from $\{b_k\}$.

To improve the rate of convergence during stochastic gradient descent, the dynamic range of $\{\rho_i\}$ and $\{\eta_k\}$ are compressed to the interval $[0, 1]$ as follows: $\bar{\boldsymbol{\rho}} = \{\frac{\rho_1}{\pi}, \frac{\rho_2}{\pi}, \ldots, \frac{\rho_p}{\pi}\}$ and $\bar{\boldsymbol{\eta}} = \{\frac{\eta_1}{\pi}, \frac{\eta_2}{\pi}, \ldots, \frac{\eta_q}{\pi}\}$.

In Yu et al. (2020), the prediction error variances $\hat{\sigma}_w^2$ and $\hat{\sigma}_u^2$ are estimated using an ML approach (Srinivasan et al., 2006) using the estimated $\{\hat{a}_i\}$ and $\{\hat{b}_k\}$. In this study, we jointly estimate $(\{\hat{a}_i\}, \hat{\sigma}_w^2)$ and $(\{\hat{b}_k\}, \hat{\sigma}_u^2)$. Hence, $\boldsymbol{\zeta}_l$ for the LPC estimation framework in Fig. 1 becomes:

$$\begin{aligned}\boldsymbol{\zeta}_l &= \{\bar{\boldsymbol{\rho}}, \bar{\boldsymbol{\eta}}, \sigma_w^2, \sigma_u^2\},\\ &= \{\bar{\rho}_1, \bar{\rho}_2, \ldots, \bar{\rho}_p, \bar{\eta}_1, \bar{\eta}_2, \ldots, \bar{\eta}_q, \sigma_w^2, \sigma_u^2\}.\end{aligned} \tag{24}$$

During inference, $\hat{\boldsymbol{\zeta}}_l$ is split into $\hat{\bar{\boldsymbol{\rho}}}$, $\hat{\bar{\boldsymbol{\eta}}}$, $\hat{\sigma}_w^2$, and $\hat{\sigma}_u^2$. Then, $\hat{\bar{\boldsymbol{\rho}}}$, $\hat{\bar{\boldsymbol{\eta}}}$ are multiplied by π (the inverse mapping of the compression function), which yield $\hat{\boldsymbol{\rho}}$ and $\hat{\boldsymbol{\eta}}$. Finally, $\hat{\boldsymbol{\rho}}$ and $\hat{\boldsymbol{\eta}}$ are converted into $\{\hat{a}_i\}$ and $\{\hat{b}_k\}$ using the LSF to LPC conversion method, as in McLoughlin (2008).

3.2. LPC-PS training target

The LPC-PS of clean speech and noise, $P_s(l,m)$ and $P_v(l,m)$ were used as the training targets in Roy et al. (2021a,b). During training, $P_s(l,m)$ and $P_v(l,m)$ are computed as in Vaseghi (2006, Chapter 9):

$$P_s(l,m) = \frac{\sigma_w^2}{\left|1 + \sum_{i=1}^{p} a_i e^{-j2\pi im/M}\right|^2}, \tag{25}$$

$$P_v(l,m) = \frac{\sigma_u^2}{\left|1 + \sum_{k=1}^{q} b_k e^{-j2\pi km/M}\right|^2}, \tag{26}$$

where $m \epsilon \{0, 1, \ldots, M-1\}$ $(M = 257)$.

The dynamic range of $P_s(l,m)$ and $P_v(l,m)$ are compressed to the interval [0, 1] through utilising the cumulative distribution function (CDF) of $P_s(l,m)_{[\text{dB}]}$ and $P_v(l,m)_{[\text{dB}]}$, where $P_s(l,m)_{[\text{dB}]} = 10\log_{10}(P_s(l,m))$ and $P_v(l,m)_{[\text{dB}]} = 10\log_{10}(P_v(l,m))$ (Roy et al., 2021a). It can be seen from Figs. 2 (a) and (c) that $P_s(l,64)_{[\text{dB}]}$ and $P_v(l,64)_{[\text{dB}]}$ follow a Gaussian distribution. Hence, it is assumed that $P_s(l,m)_{[\text{dB}]}$ and $P_v(l,m)_{[\text{dB}]}$ are distributed normally with mean, μ_s and μ_v, and variance σ_s^2 and σ_v^2, respectively ($P_s(l,m)_{[\text{dB}]} \sim \mathcal{N}(\mu_s, \sigma_s^2)$ and $P_v(l,m)_{[\text{dB}]} \sim \mathcal{N}(\mu_v, \sigma_v^2)$). The statistics of $P_s(l,m)_{[\text{dB}]}$ and $P_v(l,m)_{[\text{dB}]}$, i.e., (μ_s, σ_s^2) and (μ_v, σ_v^2) for each frequency bin m were found over a sample of the training set.[1] The resultant CDFs used to compress the dynamic range of $P_s(l,64)_{[\text{dB}]}$ and $P_v(l,64)_{[\text{dB}]}$ are shown in Figs. 2(b) and (d), respectively, and are applied as follows (Roy et al., 2021a):

$$\bar{P}_s(l,m) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{P_s(l,m)_{[\text{dB}]} - \mu_s}{\sigma_s\sqrt{2}}\right)\right], \tag{27}$$

[1] 2500 randomly selected clean speech recordings were mixed with 2500 randomly selected noise recordings from the training set (Section 4.2) with SNR levels: −10 dB to +20 dB in 1 dB increments, giving 2500 noisy speech signals. For each frequency bin, m, the sample mean and variances, (μ_s, σ_s^2) and (μ_v, σ_v^2) were computed from 2500 concatenated clean speech and scaled noise recordings, respectively.

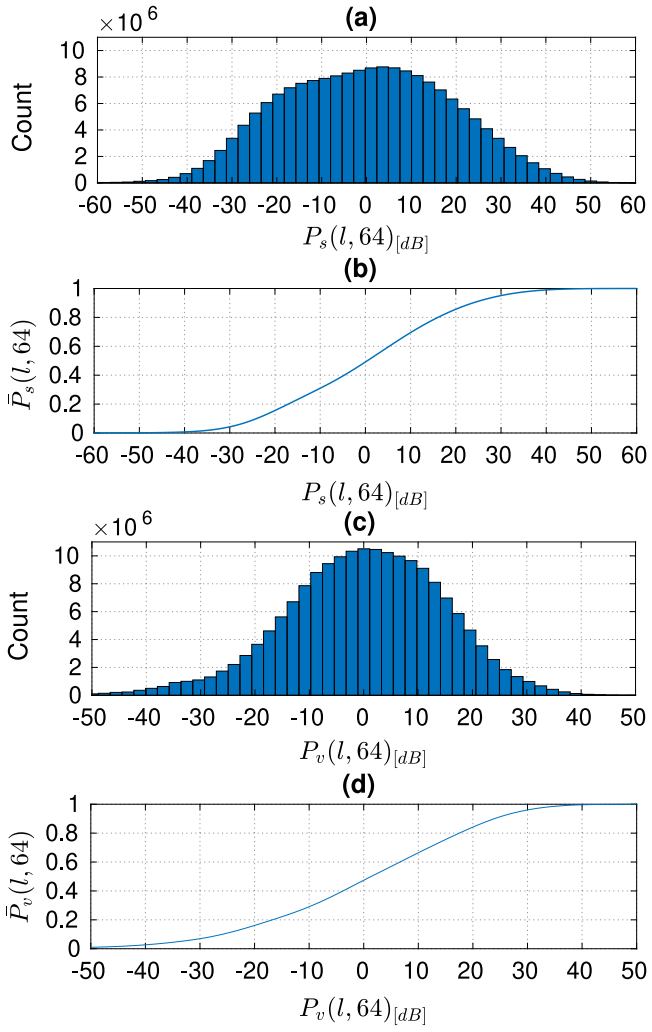

Fig. 2. (Colour online) The distribution of (a) $P_s(l,64)_{[\text{dB}]}$ and (c) $P_v(l,64)_{[\text{dB}]}$. The CDF of (b) $P_s(l,64)_{[\text{dB}]}$ and (d) $P_v(l,64)_{[\text{dB}]}$, where the sample mean and variance were found over the sample of the training set 1.

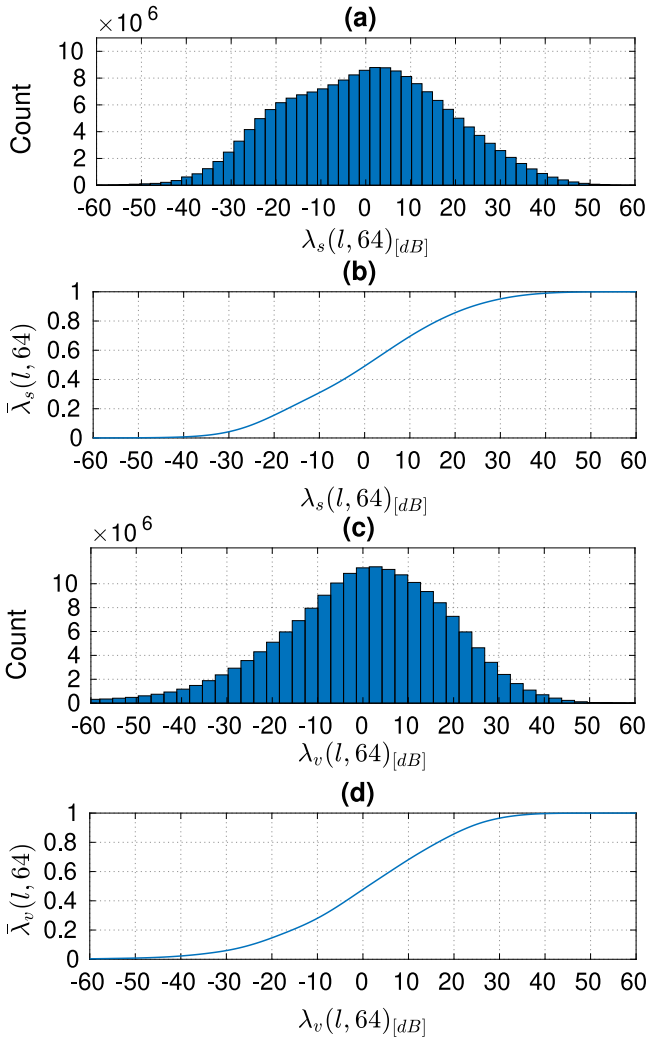

Fig. 3. (Colour online) The distribution of (a) $\lambda_s(l,64)_{[\text{dB}]}$ and (c) $\lambda_v(l,64)_{[\text{dB}]}$. The CDF of (b) $\lambda_s(l,64)_{[\text{dB}]}$ and (d) $\lambda_v(l,64)_{[\text{dB}]}$, where the sample mean and variance were found over the sample of the training set 1.

$$\bar{P}_v(l,m) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{P_v(l,m)_{[\text{dB}]} - \mu_v}{\sigma_v\sqrt{2}}\right)\right]. \tag{28}$$

The training target for the LPC estimation framework in Fig. 1 is formed by concatenating $\bar{P}_s(l,m)$ and $\bar{P}_v(l,m)$:

$$\begin{aligned}\boldsymbol{\zeta}_l = \{&\bar{P}_s(l,0), \bar{P}_s(l,1), \ldots, \bar{P}_s(l,M-1), \bar{P}_v(l,0),\\ &\bar{P}_v(l,1), \ldots, \bar{P}_v(l,M-1)\}.\end{aligned} \tag{29}$$

During inference, $\hat{\boldsymbol{\zeta}}_l$ is first split into $\hat{\bar{P}}_s(l,m)$ and $\hat{\bar{P}}_v(l,m)$. Following this, the inverse mapping of the CDFs compute the estimated LPC-PS:

$$\hat{P}_s(l,m) = 10^{((\sigma_s\sqrt{2}\text{erf}^{-1}(2\hat{\bar{P}}_s(l,m)-1)+\mu_s)/10)}, \tag{30}$$

$$\hat{P}_v(l,m) = 10^{((\sigma_v\sqrt{2}\text{erf}^{-1}(2\hat{\bar{P}}_v(l,m)-1)+\mu_v)/10)}. \tag{31}$$

The |IDFT| of $\hat{P}_s(l,m)$ and $\hat{P}_v(l,m)$ gives an estimate of the autocorrelation matrices, $\hat{R}_{ss}(\tau)$ and $\hat{R}_{vv}(\tau)$, respectively. Using Roy et al. (2021a, Equations (26) and (27)), we construct the Yule–Walker equations using $\hat{R}_{ss}(\tau)$ and $\hat{R}_{vv}(\tau)$. We solve the Yule–Walker equations using the Levinson–Durbin recursion (Vaseghi, 2006, Chapter 8), yielding ($\{\hat{a}_i\}$, $\hat{\sigma}_w^2$) ($p = 16$) and ($\{\hat{b}_k\}$, $\hat{\sigma}_u^2$) ($q = 16$).

3.3. PS training target

The PS of the clean speech and noise (denoted as $\lambda_s(l,m)$ and $\lambda_v(l,m)$) can also be used as the training targets for supervised LPC estimation. $\lambda_s(l,m)$ and $\lambda_v(l,m)$ are computed directly from the squared magnitude of the single-sided clean speech and noise spectrum, respectively: $\lambda_s(l,m) = |S(l,m)|^2$ and $\lambda_v(l,m) = |V(l,m)|^2$. As in Nicolson and Paliwal (2021), we utilise the CDF of the training targets (in dB), $\lambda_s(l,m)_{[\text{dB}]}$ and $\lambda_v(l,m)_{[\text{dB}]}$ to compress their dynamic range to the interval [0, 1], where $\lambda_s(l,m)_{[\text{dB}]} = 10\log_{10}(\lambda_s(l,m))$ and $\lambda_v(l,m)_{[\text{dB}]} = 10\log_{10}(\lambda_v(l,m))$.

We observe in Figs. 3(a) and (c) that $\lambda_s(l,64)_{[\text{dB}]}$ and $\lambda_v(l,64)_{[\text{dB}]}$ follow a Gaussian distribution. Therefore, we assume that $\lambda_s(l,m)_{[\text{dB}]}$ and $\lambda_v(l,m)_{[\text{dB}]}$ are also distributed normally, with mean μ_s and μ_v, and variance σ_s^2 and σ_v^2, respectively ($\lambda_s(l,m)_{[\text{dB}]} \sim \mathcal{N}(\mu_s, \sigma_s^2)$ and $\lambda_v(l,m)_{[\text{dB}]} \sim \mathcal{N}(\mu_v, \sigma_v^2)$). The statistics of $\lambda_s(l,m)_{[\text{dB}]}$ and $\lambda_v(l,m)_{[\text{dB}]}$, i.e., ($\mu_s$, σ_s^2) and (μ_v, σ_v^2) for each frequency bin m were found over a sample of the training set 1. The resultant CDFs used to compress the dynamic range of $\lambda_s(l,64)_{[\text{dB}]}$ and $\lambda_v(l,64)_{[\text{dB}]}$ are shown in (Fig. 3(b) and (d), respectively, and are applied as follows:

$$\bar{\lambda}_s(l,m) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{\lambda_s(l,m)_{[\text{dB}]} - \mu_s}{\sigma_s\sqrt{2}}\right)\right], \tag{32}$$

$$\bar{\lambda}_v(l,m) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{\lambda_v(l,m)_{[\text{dB}]} - \mu_v}{\sigma_v\sqrt{2}}\right)\right]. \tag{33}$$

The training target for the LPC estimation framework in Fig. 1 is formed by concatenating $\bar{\lambda}_s(l,m)$ and $\bar{\lambda}_v(l,m)$:

$$\boldsymbol{\zeta}_l = \{\bar{\lambda}_s(l,0), \bar{\lambda}_s(l,1), \ldots, \bar{\lambda}_s(l,M-1), \bar{\lambda}_v(l,0), \bar{\lambda}_v(l,1), \ldots, \bar{\lambda}_v(l,M-1)\}. \tag{34}$$

During inference, $\hat{\boldsymbol{\zeta}}_l$ is first split into $\hat{\bar{\lambda}}_s(l,m)$ and $\hat{\bar{\lambda}}_v(l,m)$. Following this, the inverse mapping of the CDFs are used to obtain $\hat{\lambda}_s(l,m)$ and $\hat{\lambda}_v(l,m)$:

$$\hat{\lambda}_s(l,m) = 10^{((\sigma_s\sqrt{2}\text{erf}^{-1}(2\hat{\bar{\lambda}}_s(l,m)-1)+\mu_s)/10)}, \tag{35}$$

$$\hat{\lambda}_v(l,m) = 10^{((\sigma_v\sqrt{2}\text{erf}^{-1}(2\hat{\bar{\lambda}}_v(l,m)-1)+\mu_v)/10)}. \tag{36}$$

The |IDFT| of $\hat{\lambda}_s(l,m)$ and $\hat{\lambda}_v(l,m)$ yields an estimate of the autocorrelation matrices, $\hat{R}_{ss}(\tau)$ and $\hat{R}_{vv}(\tau)$, respectively. Using Roy et al. (2021a, Equations (26) and(27)), we construct the Yule–Walker equations using $\hat{R}_{ss}(\tau)$ and $\hat{R}_{vv}(\tau)$. The Yule–Walker equations are then solved using the Levinson–Durbin recursion (Vaseghi, 2006, Chapter 8), yielding $(\{\hat{a}_i\}, \hat{\sigma}_w^2)$ $(p = 16)$ and $(\{\hat{b}_k\}, \hat{\sigma}_u^2)$ $(q = 16)$.

3.4. MS training target

The magnitude of the single-sided clean speech and noise spectrum (denoted as $C_s(l,m)$ and $C_v(l,m)$) can also be used as the training targets for supervised LPC estimation (Roy and Paliwal, 2020a). $C_s(l,m)$ and $C_v(l,m)$ are computed directly from the magnitude of the clean speech and noise spectral components: $C_s(l,m) = |S(l,m)|$ and $C_v(l,m) = |V(l,m)|$. Min–max normalisation (Han et al., 2011, section 3.5.2) is then employed to compress the dynamic range of $C_s(l,m)$ and $C_v(l,m)$, as in Roy and Paliwal (2020a):

$$\bar{C}_s(l,m) = \frac{C_s(l,m) - S_{min}(l)}{S_{max}(l) - S_{min}(l)}, \tag{37}$$

$$\bar{C}_v(l,m) = \frac{C_v(l,m) - V_{min}(l)}{V_{max}(l) - V_{min}(l)}, \tag{38}$$

where $(S_{max}(l), S_{min}(l))$ and $(V_{max}(l), V_{min}(l))$ are the minimum and maximum values for each frequency bin m of the clean speech and noise MS, respectively, over all frames in the aforementioned sample 1. The training target for the LPC estimation framework in Fig. 1 is formed by concatenating $\bar{C}_s(l,m)$ and $\bar{C}_v(l,m)$:

$$\boldsymbol{\zeta}_l = \{\bar{C}_s(l,0), \bar{C}_s(l,1), \ldots, \bar{C}_s(l,M-1), \bar{C}_v(l,0), \bar{C}_v(l,1), \ldots, \bar{C}_v(l,M-1)\}. \tag{39}$$

During inference, $\hat{\boldsymbol{\zeta}}_l$ is first split into $\hat{\bar{C}}_s(l,m)$ and $\hat{\bar{C}}_v(l,m)$. Next, the clean speech and noise magnitude spectra are computed from $\hat{\bar{C}}_s(l,m)$ and $\hat{\bar{C}}_v(l,m)$ using inverse min–max normalisation (Han et al., 2011, section 3.5.2):

$$\hat{C}_s(l,m) = S_{min}(l) + \hat{\bar{C}}_s(l,m)(S_{max}(l) - S_{min}(l)), \tag{40}$$

$$\hat{C}_v(l,m) = V_{min}(l) + \hat{\bar{C}}_v(l,m)(V_{max}(l) - V_{min}(l)). \tag{41}$$

Taking the square of $\hat{C}_s(l,m)$ and $\hat{C}_v(l,m)$ followed by the |IDFT| gives the autocorrelation matrices, $\hat{R}_{ss}(\tau)$ and $\hat{R}_{vv}(\tau)$. Using Roy et al. (2021a, Equations (26) and(27)), we construct the Yule–Walker equations using $\hat{R}_{ss}(\tau)$ and $\hat{R}_{vv}(\tau)$. The Yule–Walker equations are then solved using the Levinson–Durbin recursion (Vaseghi, 2006, Chapter 8), yielding $(\{\hat{a}_i\}, \hat{\sigma}_w^2)$ $(p = 16)$ and $(\{\hat{b}_k\}, \hat{\sigma}_u^2)$ $(q = 16)$.

4. Experimental setup

4.1. Deep neural networks

The DNNs used in this study – ResNet-TCN and MHANet – are briefly described below. Each is trained to map $|\boldsymbol{Y}_l|$ to $\boldsymbol{\zeta}_l$, where $\boldsymbol{\zeta}_l$ is given by Eqs. (24), (29), (34), and (39). During inference, each DNN computes $\hat{\boldsymbol{\zeta}}$. The training strategy for each DNN is detailed in Section 4.3.

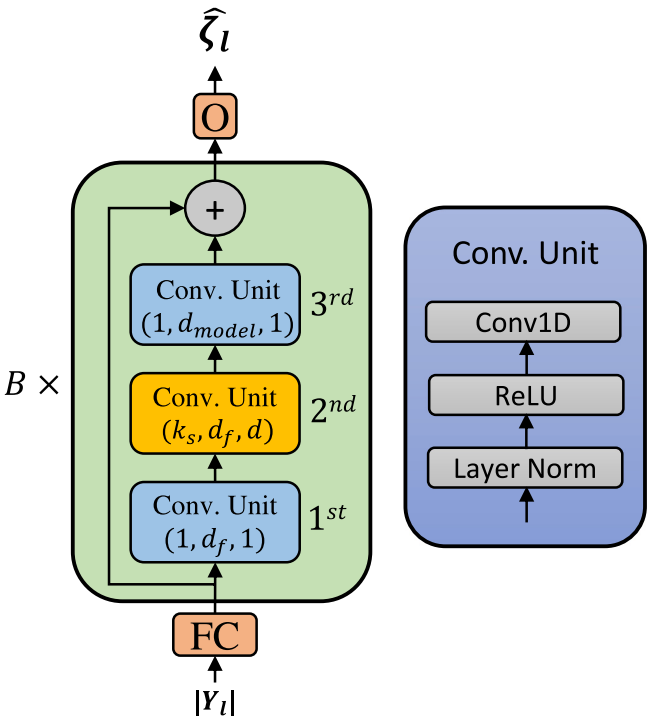

Fig. 4. (Colour online) ResNet-TCN. The kernel size, output size, and dilation rate for each convolutional unit is denoted as (**kernel size**, **output size**, **dilation rate**).

4.1.1. ResNet-TCN

The ResNet-TCN used for the DeepLPC framework (Roy et al., 2021a) is used in this study to estimate $\boldsymbol{\zeta}_l$ (Eqs. (24), (29), (34), and (39)) from $|\boldsymbol{Y}_l|$. The ResNet-TCN is shown in Fig. 4. For each training target (Section 3), the input, $|\boldsymbol{Y}_l|$ is first passed through **FC**, a fully-connected layer of size d_{model}, followed by layer normalisation (LN) (Ba et al., 2016) and the rectified linear unit (ReLU) activation function (He et al., 2015). **FC** is followed by B bottleneck residual blocks, where $j\epsilon\{1,2,\ldots,B\}$ is the block index. Each block comprise of three one-dimensional causal convolutional units. Each convolutional unit (CU) is pre-activated by LN (Ba et al., 2016) followed by the ReLU activation function (He et al., 2015). The kernel size, output size, and dilation rate for each convolutional unit is denoted as (**kernel size**, **output size**, **dilation rate**).

The first and third CUs in each block have a kernel size of one, whilst the second convolutional unit has a kernel size of k_s. The output size of the first and second CU is d_f, while the third one is d_{model}. A dilation rate of one is set for the first and the third CU, which is d for the second CU. The second CU provides a contextual field over previous time steps. The dilation rate, d is cycled as the block index j increases as: $d = 2^{(j-1 \bmod (\log_2(D)+1))}$, where mod is the modulo operation, and D is the maximum dilation rate. The last residual block is followed by the output layer, **O**, which is a fully-connected layer with sigmoidal units. The **O** layer gives an estimate of $\hat{\boldsymbol{\zeta}}_l$. For the LPC-PS, PS, and MS training targets, the hyperparameters used in DeepLPC (Roy et al., 2021a) were used: $d_{model} = 256$, $d_f = 64$, $B = 40$, $k_s = 3$, and $D = 16$. With this set of hyperparameters, ResNet-TCN exhibits approximately 2.1 million parameters. For the LSF training target, all the above hyperparameters were used except $d_f = 34$, giving around 1.91 million parameters.

4.1.2. MHANet

MHANet is an attention-based architecture for speech enhancement that is based on the encoder of the Transformer (Nicolson and Paliwal, 2020). Along with ResNet-TCN, we use it to compare LPC estimation training targets. MHANet is briefly summarised in this Section. The

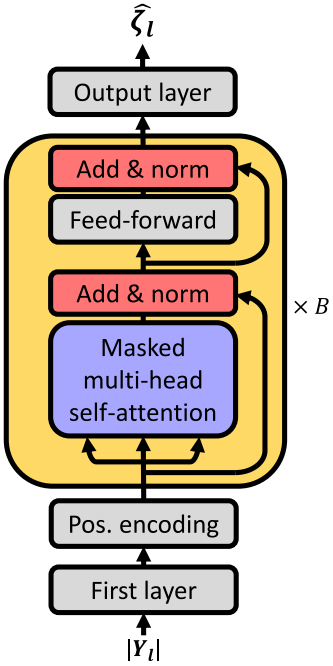

Fig. 5. (Colour online) MHANet for LPC estimation.

simplest form of MHANet is shown in Fig. 5. The processing steps of the MHANet from input to output are described as follows. The first layer in MHANet is used to project the input to a size of d_{model}. As in Nicolson and Paliwal (2019), the first layer is formed as: $\max(0, \text{LN}(|\boldsymbol{X}|\boldsymbol{W}^I + \boldsymbol{b}_s^I))$, where LN is frame-wise layer normalisation (Ba et al., 2016), and $\boldsymbol{W}^I \in \mathbb{R}^{M \times d_{model}}$ and $\boldsymbol{b}_s^I \in \mathbb{R}^{d_{model}}$ are the learnable weights and biases of the first layer, respectively. Next, the positional encoding from Nicolson and Paliwal (2021, Section A.2) is added after the first layer, where the time-frame index indicates the position. The positional encoding is learned using weight matrix $\mathbf{W}_p$, with a maximum length of 2048 time-frames (i.e. $\mathbf{W}_p \in \mathbb{R}^{2048 \times 256}$). This is followed by B cascading blocks. Each block includes an MHA module, a two-layer feed-forward neural network (FNN), residual connections (He et al., 2016), and frame-wise LN (Ba et al., 2016). For a detailed description of the blocks, we refer the reader to Nicolson and Paliwal (2020, Section 3.1). The last block is followed by the output layer, which is a sigmoidal feed-forward layer, as in Nicolson and Paliwal (2019). We use configuration from Nicolson and Paliwal (2020, Section A.2) for MHANet: $B = 5$, $d_f = 1024$, $d_{model} = 256$, $H = 8$, $P_{drop} = 0.0$, and $\Gamma = 40\,000$. With this set of hyperparameters, MHANet exhibits approximately 4.27 million parameters.

4.2. Training and validation set

The noisy speech for the training and validation sets are formed from clean speech and noise recordings. For the clean speech recordings, the *train-clean-100* set of the Librispeech corpus (Panayotov et al., 2015) (28 539), the CSTR VCTK corpus (Veaux et al., 2019) (42 015), and the si^* and sx^* training sets of the TIMIT corpus (Garofolo et al., 1993) (3696) were used, giving a total 74 250 clean speech recordings. To form the validation set, 5% of the clean speech recordings (3713) are randomly selected. Thus, 70 537 of the clean speech recordings are used for the training set. For the noise recordings, the QUT-NOISE dataset (Dean et al., 2010), the Nonspeech dataset (Hu, 2004), the Environmental Background Noise dataset (Saki et al., 2016; Saki and Kehtarnavaz, 2016), the noise set from the MUSAN corpus (Snyder et al., 2015), multiple FreeSound packs (https://freesound.org/),[2] and coloured noise recordings (with an α value ranging from 2 to 2 in increments of 0.25) were used, giving a total of 16 243 noise recordings. For the validation set, 5% of the noise recordings (813) are randomly selected. The remaining 15 430 noise recordings are used for the training set. All the clean speech and noise recordings are single-channel with a sampling frequency of 16 kHz. To create the noisy speech for the validation set, each of the 3713 clean speech recordings is corrupted by a random section of a randomly selected noise recording (from the set of 813 noise recordings) at a randomly selected SNR level (−10 to +20 dB, in 1 dB increments). The noisy speech for the training set was created using the method described in Section 4.3.

4.3. Training strategy

The following training strategy was employed for training ResNet-TCN and MHANet:

- Mean squared error is used as the loss function.
- The *Adam* optimiser (Kingma and Ba, 2014) is used for stochastic gradient descent optimisation when training ResNet-TCN and MHANet. For ResNet-TCN, the default hyperparameters were used. For MHANet, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ were used, where the learning rate, α_r, depends on the training step (Vaswani et al., 2017):

$$\alpha_r = d_{model}^{-0.5} \cdot \min(\gamma^{-0.5}, \gamma \cdot \Gamma^{-1.5}), \tag{42}$$

  where γ is the total number of training steps and Γ is the number of warmup steps.
- Gradient norms that exceed [−1, 1] are clipped (Pascanu et al., 2013).
- The number of training examples in an epoch is equal to the number of clean speech recordings used in the training set, i.e., 70 537.
- A mini-batch size of eight training examples is used. To generate the mini-batch, at first, we select the shortest length of the corrupted utterance among the 8 utterances in the mini-batch. Then we pad the samples of the other utterances in the min-batch longer than the shortest length utterance.
- The noisy speech signals are generated on the fly as follows: each clean speech recording is randomly selected and corrupted with a random section of a randomly selected noise recording at a randomly selected SNR level (−10 to +20 dB, in 1 dB increments).[3]
- A total of 150 epochs are used to train both ResNet-TCN and MHANet.

4.4. Test set

For the objective experiments, 30 phonetically balanced IEEE utterances belonging to six speakers (three male and three female) are taken from the NOIZEUS corpus (Loizou, 2013, Chapter 12). In this experiment, filtering is not performed to the clean speech utterances as in the original NOIZEUS corpus (Loizou, 2013, Chapter 12). The noisy speech for the test set is formed by mixing the clean speech with real-world non-stationary (*voice babble, street, restaurant*, and *shopping mall*) and coloured (*factory1, factory2, hfchannel*, and *f16*) noise recordings selected from (Saki et al., 2016; Saki and Kehtarnavaz, 2016; Pearce and Hirsch, 2000; Varga and Steeneken, 1993) at multiple SNR levels

[2] FreeSound packs that were used: 147, 199, 247, 379, 622, 643, 1133, 1563, 1840, 2432, 4366, 4439, 15 046, 15 598, 21 558.

[3] For clean speech recordings longer than the noise recordings, we simply append the noise recording until it becomes larger than or equal to the clean speech recording. Then, the noise recording is clipped to the length of the clean speech recording. The same applies when generating the validation set.

varying from −5 dB to +15 dB, in 5 dB increments. This provides 30 examples per condition with 40 total conditions, yielding 1200 examples. All the clean speech and noise recordings are single-channel with a sampling frequency of 16 kHz. Note that the speech and the noise recordings in the test set are different from those used in the training and validation sets.

4.5. SD level evaluation

The frame-wise spectral distortion (SD) (dB) (Gray and Markel, 1976) is used to evaluate the accuracy of LPC estimates obtained using ResNet-TCN and MHANet for the training targets; LSF, LPC-PS, PS, and MS. Specifically, the estimated clean speech LPCs are evaluated. The SD for lth frame, denoted by D_l (in dB) is defined as the root-mean-square-difference between the LPC-PS estimate in dB $\hat{P}_s(l,m)_{[dB]}$, and the oracle case in dB $P_s(l,m)_{[dB]}$, as in Gray and Markel (1976):

$$D_l = \sqrt{\frac{1}{M}\sum_{m=0}^{M-1}\left[P_s(l,m)_{[dB]} - \hat{P}_s(l,m)_{[dB]}\right]^2}. \tag{43}$$

4.6. Speech enhancement methods

We also evaluate the speech enhancement performance of each training target (as described in "3. **AKF**" below). We also compare the performance of each LPC estimation target to other SEAs in the literature:

1. **Noisy**: speech corrupted with additive noise.
2. **Oracle-AKF:** AKF, where $(\{a_i\}, \sigma_w^2)$ and $(\{b_k\}, \sigma_u^2)$ are computed from the clean speech and noise, respectively, where $p = 16$, $q = 16$, $w_f = 32$ ms, $s_f = 16$ ms, and a rectangular window is used for framing.
3. **AKF** constructed from the speech and noise LPC estimates derived from the training targets — LSF, LPC-PS, PS, and MS estimated using ResNet-TCN and MHANet. Thus, there are eight AKF methods, where $p = 16$, $q = 16$, window length=32 ms, frame shift=16 ms, and a rectangular window is used for framing.
4. **LSTM-CKFS** (Yu et al., 2020): AKF constructed using $(\{a_i\}, \sigma_w^2)$ and $(\{b_k\}, \sigma_u^2)$ are computed using an LSTM and maximum-likelihood (ML)-based approaches (Srinivasan et al., 2006), followed by post subtraction using the multi-band spectral subtraction (MB-SS) method (Kamath and Loizou, 2002), where $p = 12$, $q = 12$, $w_f = 20$ ms, $s_f = 0$ ms, and a rectangular window is used for framing.
5. **IAM-IFD** (Zheng and Zhang, 2019): Phase-aware DNN for speech enhancement, where $w_f = 20$ ms, $s_f = 5$ ms, and the Hamming window is used for analysis and synthesis.
6. **ResNet20-AKF** (Roy and Paliwal, 2020b): AKF-based SEA, where $(\{b_k\}, \sigma_u^2)$ is estimated using the ResNet20-based method and $(\{a_i\}, \sigma_w^2)$ are computed from pre-whitened speech corresponding to each noisy speech frame, where $p = 16$, $q = 16$, $w_f = 32$ ms, $s_f = 16$ ms, and a rectangular window is used for framing.
7. **Deep Xi-ResNet-TCN-MMSE-LSA**: Deep Xi-ResNet-TCN (Zhang et al., 2020) estimates the *a priori* SNR for the MMSE-LSA estimator (Ephraim and Malah, 1985), where $w_f = 32$ ms, $s_f = 16$ ms.
8. **Deep Xi-MHANet-MMSE-LSA**: Deep Xi-MHANet (Nicolson and Paliwal, 2020) estimates the *a priori* SNR for the MMSE-LSA estimator (Ephraim and Malah, 1985), where $w_f = 32$ ms, $s_f = 16$ ms.

4.7. Objective quality and intelligibility measures

Objective measures are used to evaluate the quality and intelligibility of the enhanced speech with respect to the corresponding clean speech. Table 1 shows the objective quality and intelligibility measures used in this study.

Table 1
Objective measures, what each assesses, and the range of their scores. For each measure, higher is better.

Measure	Assesses	Range
CSIG (Hu and Loizou, 2008)	Quality	[1, 5]
CBAK (Hu and Loizou, 2008)	Quality	[1, 5]
COVL (Hu and Loizou, 2008)	Quality	[1, 5]
PESQ (Rix et al., 2001)	Quality	[−0.5, 4.5]
STOI (Taal et al., 2011)	Intelligibility	[0, 100]%
SI-SDR (Roux et al., 2019)	Quality	[−∞, ∞]
SegSNR (Mermelstein, 1979)	Quality	[−∞, ∞]

4.8. Subjective evaluation for speech enhancement

The subjective evaluation was carried out through a series of blind AB listening tests (Paliwal et al., 2010, Section 3.3.4). To perform the tests, we generated a set of stimuli by corrupting six IEEE utterances *sp01*, *sp05*, *sp10*, *sp15*, *sp26*, and *sp27* from the NOIZEUS corpus (Loizou, 2013, Chapter 12). The reference transcript for recording *sp01* is: "*The birch canoe slid on the smooth planks*", and is corrupted with *hfchannel* at 0 dB. The reference transcript for recording *sp05* is: "*Wipe the grease off his dirty face*", and is corrupted with *f16* at 5 dB. The reference transcript for recording *sp10* is: "*The sky that morning was clear and bright blue*", and is corrupted with *voice babble* at 10 dB. The reference transcript for recording *sp15* is: "*The clothes dried on a thin wooden rack*", and is corrupted with *shopping mall* at 0 dB. The reference transcript for recording *sp26* is: "*She has a smart way of wearing clothes*", and is corrupted with *street* at 0 dB. The reference transcript for recording *sp27* is: "*Bring your best compass to the third class*", and is corrupted with *factory2* at 10 dB. Utterances *sp01*, *sp05*, and *sp10* were uttered by male and utterances *sp15*, *sp26*, and *sp27* were uttered by female, respectively. In this test, the enhanced speech produced by eight SEAs as well as the corresponding clean speech and noisy speech signals were played as stimuli pairs to the listeners. Specifically, the test is performed on a total of 540 stimuli pairs (90 for each utterance) played in a random order to each listener, excluding the comparisons between the same method. Each listener's perceptual preference for the first or second stimuli was recorded. Pairwise scoring was used, with 100% award is given to the preferred method, 0% to the other, and 50% to both if there was no preference. The participants could re-listen to the stimuli if required. Ten English speaking listeners participate in the blind AB listening tests.[4] The average of the preference scores given by the listeners termed as mean subjective preference score (%), is used to subjectively compare the SEAs.

5. Results and discussion

5.1. SD level comparison

The average clean speech LPC estimation SD levels attained by each of the training targets are shown in Fig. 6. The SD levels for noisy speech indicate the upper bounds of the SD level. It can be seen that the LPC-PS is able to produce the lowest SD levels for both real-world non-stationary as well as coloured noise conditions. PS produced the next lowest SD level. This indicates that the LPC-PS as the training target produces the most accurate clean speech LPC estimates. The low SD levels attained by LPC-PS will be of benefit to the AKF for speech enhancement.

[4] The AB listening tests were conducted with approval from Griffith University's Human Research Ethics Committee: database protocol number 2018/671.

Table 2
Mean objective scores on NOIZEUS corpus in terms of CSIG, CBAK, COVL, PESQ, STOI, SegSNR, and SI-SDR. Apart from Oracle-AKF, the highest score amongst the competing methods for each measure is given in boldface.

Methods	CSIG	CBAK	COVL	PESQ	STOI	SegSNR	SI-SDR
Noisy speech	2.33	2.21	2.02	1.36	52.74	1.13	5.87
LSTM-CKFS	2.86	2.44	2.35	1.87	77.59	7.12	11.89
ResNet-TCN-LSF-AKF	2.94	2.52	2.43	1.96	77.86	7.21	12.12
MHANet-LSF-AKF	3.03	2.68	2.51	2.04	78.33	7.28	12.31
IAM-IFD	3.10	2.74	2.58	2.12	78.46	7.37	12.49
ResNet20-AKF	3.18	2.81	2.64	2.21	79.78	7.45	12.68
ResNet-TCN-MS-AKF	3.26	2.87	2.72	2.27	80.67	7.58	12.91
ResNet-TCN-PS-AKF	3.34	2.93	2.80	2.33	81.47	7.62	13.11
Deep Xi-ResNet-TCN-MMSE-LSA	3.39	3.04	2.89	2.39	82.66	7.75	13.47
MHANet-MS-AKF	3.41	3.11	3.02	2.43	83.79	7.96	13.69
MHANet-PS-AKF	3.47	3.19	3.11	2.52	85.35	8.78	14.14
ResNet-TCN-LPC-PS-AKF	3.54	3.25	3.19	2.59	85.84	9.45	14.81
Deep Xi-MHANet-MMSE-LSA	3.66	3.36	**3.37**	2.67	88.66	**10.05**	15.52
MHANet-LPC-PS-AKF	**3.74**	**3.47**	3.33	**2.76**	**89.12**	9.97	**16.01**
Oracle-AKF	4.36	4.17	4.03	2.81	95.88	11.04	17.02

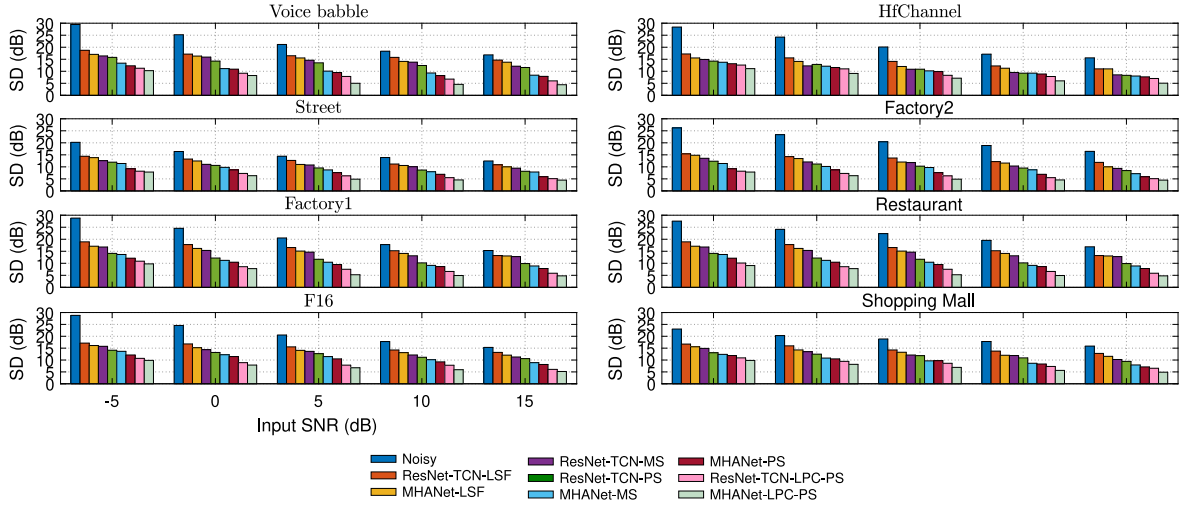

Fig. 6. (Colour online) Average clean speech LPC estimation SD (dB) level for each SEA found over all frames for each test condition in Section 4.4.

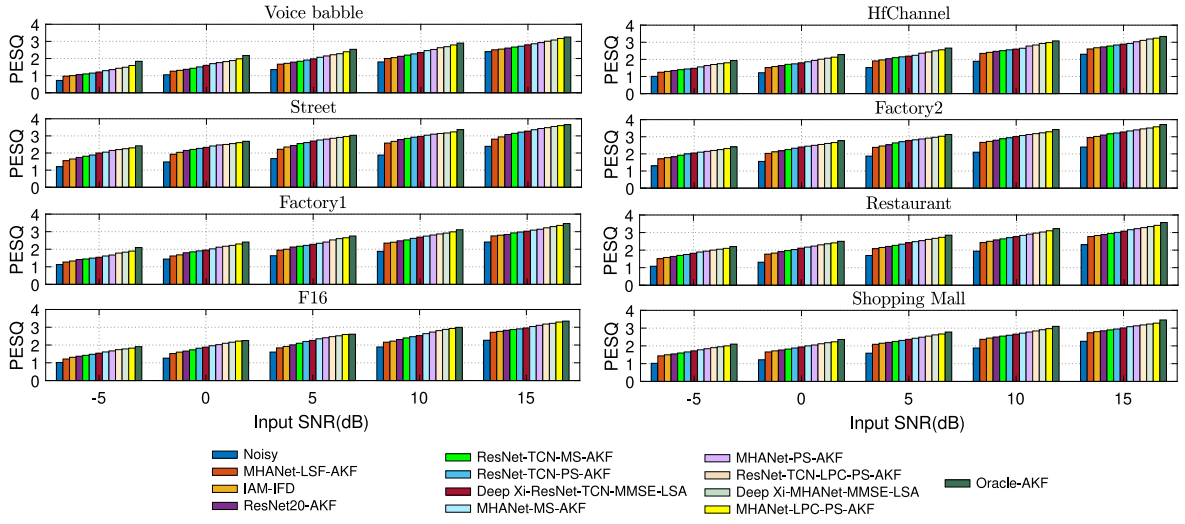

Fig. 7. (Colour online) PESQ score for each SEA and for each condition specified in Section 4.4.

5.2. Objective evaluation

In this section, we analyse the speech enhancement performance of the AKF constructed using the clean speech and noise LPC estimates given by each training target. The objective measures are described in Table 1. We also compare each LPC estimation training target to other deep learning approaches to speech enhancement described in Section 4.6.

The mean objective scores on the NOIZEUS corpus are shown in Tables 2. It can be seen that Oracle-AKF produces the highest objective

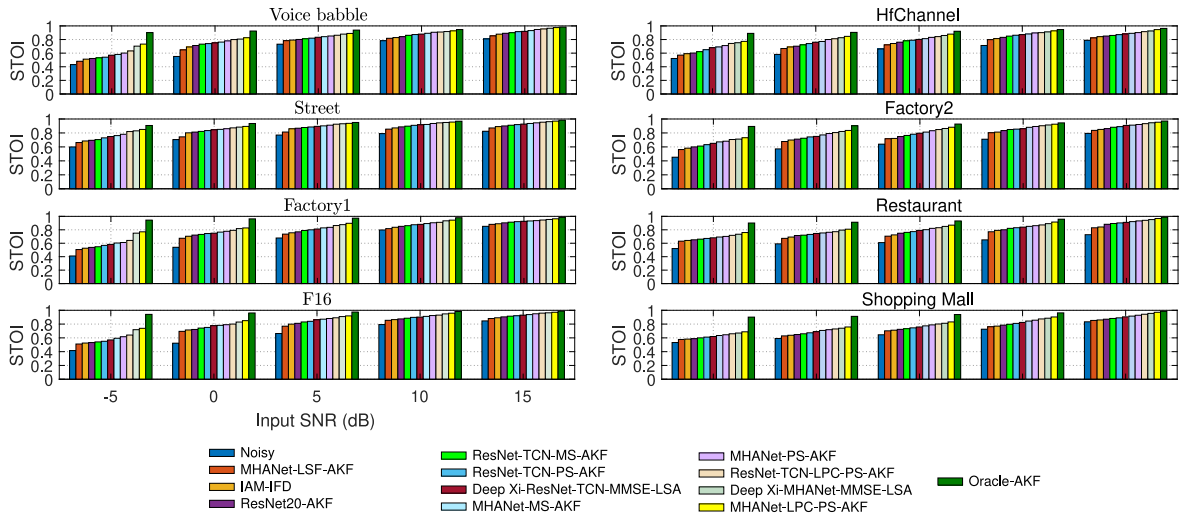

Fig. 8. (Colour online) STOI score for each SEA and for each condition specified in Section 4.4.

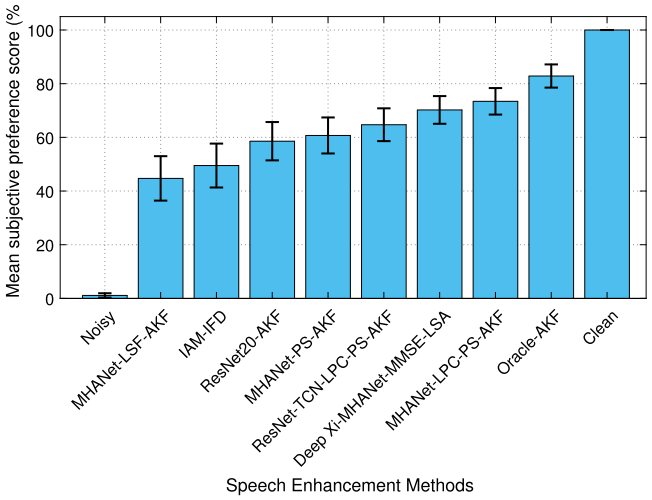

Fig. 9. (Colour online) The mean subjective preference score (%) comparison among the competing SEAs for the stimuli set described in Section 4.8.

scores amongst all methods, which is the upper boundary of speech enhancement performance for the AKF. Noisy speech produced the lowest objective scores amongst all methods, indicating the lower boundary of performance. LPC-PS produced the best objective scores amongst the LPC estimation training targets. This is likely due to the fact that LPC-PS as the training target exhibits the least amount of bias. Amongst the SEAs, MHANet-LPC-PS-AKF performed best, attaining the highest CSIG, CBAK, PESQ, STOI, and SI-SDR scores (except for COVL, SegSNR). Deep Xi-MHANet-MMSE-LSA was the next best performing SEA, producing the highest COVL and SegSNR scores.

Figs. 7 and 8 show the PESQ and STOI scores, respectively, of each SEA for multiple conditions. MHANet-LPC-PS-AKF produced the highest PESQ and STOI scores for each condition. Following MHANet-LPC-PS-AKF, Deep Xi-MHANet-MMSE-LSA (Nicolson and Paliwal, 2020) attained the next highest objective scores for each condition. This indicates that LPC-PS as the training target enables the AKF to objectively outperform the MMSE-LSA estimator with the *a priori* SNR as the training target.

5.3. Subjective evaluation by AB listening test

The mean subjective preference score (%) for each SEA is shown in Fig. 9. For this study, we selected the eight SEAs from Section 5.2 that achieved the highest objective quality and intelligibility scores. It can be seen that MHANet-LPC-PS-AKF is preferred (73.43%) by the listeners, apart from the clean speech (100%) and the Oracle-AKF method (82.86%). Deep Xi-MHANet-MMSE-LSA was the next most preferred (70.21%), followed by ResNet-TCN-LPC-PS-AKF (64.70%), MHANet-PS-AKF (60.71%), ResNet20-AKF (58.57%), IAM-IFD (49.50%) and then MHANet-LSF-AKF (44.71%). This indicates that the enhanced speech produced by MHANet-LPC-PS-AKF exhibits the highest perceived quality amongst all tested SEAs.

6. Conclusion

This paper conducts a comprehensive study on LPC estimation training targets, namely LSF, LPC-PS, PS, and MS training targets. Experiments on the NOIZEUS dataset demonstrate that LPC-PS produces the lowest clean speech LPC estimation SD levels amongst all of the training targets. Objective and subjective scores also indicate that the AKF produces the highest quality and intelligibility enhanced speech when constructed with the clean speech and noise LPC estimates derived from the LPC-PS training target. Moreover, we find that pairing the LPC-PS training with the AKF produces higher quality and intelligibility enhanced speech than pairing the *a priori* SNR as the training target with the MMSE-LSA estimator. We also show that MHANet is able to outperform the ResNet-TCN in terms of objective and subjective quality and intelligibility scores, as well as clean speech LPC estimation SD levels.

CRediT authorship contribution statement

Sujan Kumar Roy: Conceptualization, Methodology, Software, Data curation, Writing – review & editing, Investigation, Visualization. **Aaron Nicolson:** Writing – review & editing. **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.

Data availability

Data will be made available on request.

Acknowledgements

We gratefully thank the reviewers and editors for their effort in the improvement of this work.

References

Ba, L.J., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. CoRR abs/1607.06450, arXiv:1607.06450.

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

Boll, S., 1979. Suppression of acoustic noise in speech using spectral subtraction. IEEE Trans. Acoust. Speech Signal Process. 27, 113–120. http://dx.doi.org/10.1109/TASSP.1979.1163209.

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. In: Proceedings Interspeech 2010. pp. 3110–3113.

Ephraim, Y., Malah, D., 1984. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. 32 (6), 1109–1121. http://dx.doi.org/10.1109/TASSP.1984.1164453.

Ephraim, Y., Malah, D., 1985. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. 33 (2), 443–445. http://dx.doi.org/10.1109/TASSP.1985.1164550.

Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S., 1993. DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. NIST Speech Disc 1-1.1. NASA STI/Recon Technical Report N, 93.

George, A.E., So, S., Ghosh, R., Paliwal, K.K., 2018. Robustness metric-based tuning of the augmented Kalman filter for the enhancement of speech corrupted with coloured noise. Speech Commun. 105, 62–76. http://dx.doi.org/10.1016/j.specom.2018.10.002.

Gibson, J.D., Koo, B., Gray, S.D., 1991. Filtering of colored noise for speech enhancement and coding. IEEE Trans. Signal Process. 39 (8), 1732–1742. http://dx.doi.org/10.1109/78.91144.

Gray, A., Markel, J., 1976. Distance measures for speech processing. IEEE Trans. Acoust. Speech Signal Process. 24 (5), 380–391. http://dx.doi.org/10.1109/TASSP.1976.1162849.

Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. In: The Morgan Kaufmann Series in Data Management Systems, Elsevier Science.

Han, K., Wang, Y., Wang, D., Woods, W.S., Merks, I., Zhang, T., 2015. Learning spectral mapping for speech dereverberation and denoising. IEEE/ACM Trans. Audio Speech Lang. Process. 23 (6), 982–992. http://dx.doi.org/10.1109/TASLP.2015.2416653.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. CoRR abs/1502.01852, arXiv:1502.01852.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90.

Hu, G., 2004. 100 nonspeech environmental sounds. The Ohio State University, Department of Computer Science and Engineering.

Hu, Y., Loizou, P.C., 2008. Evaluation of objective quality measures for speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. 16 (1), 229–238. http://dx.doi.org/10.1109/TASL.2007.911054.

Itakura, F., 1975. Line spectrum representation of linear predictor coefficients of speech signals. J. Acoust. Soc. Am. 57, http://dx.doi.org/10.1121/1.1995189.

Kamath, S., Loizou, P., 2002. A multi-band spectral subtraction method for enhancing speech corrupted by colored noise. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4. pp. 4160–4164. http://dx.doi.org/10.1109/ICASSP.2002.5745591.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.

Loizou, P.C., 2013. Speech Enhancement: Theory and Practice, 2nd ed. CRC Press Inc., Boca Raton, FL, USA.

McLoughlin, I.V., 2008. Line spectral pairs. Signal Process. 88 (3), 448–467. http://dx.doi.org/10.1016/j.sigpro.2007.09.003.

Mermelstein, P., 1979. Evaluation of a segmental SNR measure as an indicator of the quality of ADPCM coded speech. J. Acoust. Soc. Am. 66 (6), 1664–1667. http://dx.doi.org/10.1121/1.383638.

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

Nicolson, A., Paliwal, K.K., 2020. Masked multi-head self-attention for causal speech enhancement. Speech Commun. 125, 80–96. http://dx.doi.org/10.1016/j.specom.2020.10.004.

Nicolson, A., Paliwal, K.K., 2021. On training targets for deep learning approaches to clean speech magnitude spectrum estimation. J. Acoust. Soc. Am. 149 (5), 3273–3293. http://dx.doi.org/10.1121/10.0004823.

Paliwal, K., Basu, A., 1987. A speech enhancement method based on Kalman filtering. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 12. pp. 177–180. http://dx.doi.org/10.1109/ICASSP.1987.1169756.

Paliwal, K., Wójcicki, K., Schwerin, B., 2010. Single-channel speech enhancement using spectral subtraction in the short-time modulation domain. Speech Commun. 52 (5), 450–475. http://dx.doi.org/10.1016/j.specom.2010.02.004.

Paliwal, K., Wójcicki, K., Shannon, B., 2011. The importance of phase in speech enhancement. Speech Commun. 53 (4), 465–494. http://dx.doi.org/10.1016/j.specom.2010.12.003.

Panayotov, V., Chen, G., Povey, D., Khudanpur, S., 2015. Librispeech: An ASR corpus based on public domain audio books. In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 5206–5210. http://dx.doi.org/10.1109/ICASSP.2015.7178964.

Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: Proceedings of the 30th International Conference on International Conference on Machine Learning - Vol. 28. In: ICML'13, JMLR.org, pp. III–1310–III–1318.

Pearce, D., Hirsch, H.-G., 2000. The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. Interspeech 29–32.

Pickersgill, C., So, S., Schwerin, B., 2018. Investigation of DNN prediction of power spectral envelopes for speech coding & ASR. In: 17th Speech Science and Technology Conference. SST2018, Sydney, Australia.

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. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2. pp. 749–752. http://dx.doi.org/10.1109/ICASSP.2001.941023.

Roux, J.L., Wisdom, S., Erdogan, H., Hershey, J.R., 2019. SDR – Half-baked or well done?In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 626–630. http://dx.doi.org/10.1109/ICASSP.2019.8683855.

Roy, S.K., Nicolson, A., Paliwal, K.K., 2020a. A deep learning-based Kalman filter for speech enhancement. In: Proc. Interspeech 2020. pp. 2692–2696. http://dx.doi.org/10.21437/Interspeech.2020-1551.

Roy, S.K., Nicolson, A., Paliwal, K.K., 2020b. Deep learning with augmented Kalman filter for single-channel speech enhancement. In: 2020 IEEE International Symposium on Circuits and Systems. ISCAS, pp. 1–5. http://dx.doi.org/10.1109/ISCAS45731.2020.9180820.

Roy, S.K., Nicolson, A., Paliwal, K.K., 2021a. DeepLPC: A deep learning approach to augmented Kalman filter-based single-channel speech enhancement. IEEE Access 9, 64524–64538. http://dx.doi.org/10.1109/ACCESS.2021.3075209.

Roy, S.K., Nicolson, A., Paliwal, K.K., 2021b. DeepLPC-MHANet: Multi-head self-attention for augmented Kalman filter-based speech enhancement. IEEE Access 9, 70516–70530. http://dx.doi.org/10.1109/ACCESS.2021.3077281.

Roy, S.K., Paliwal, K.K., 2020a. Causal convolutional encoder decoder-based augmented Kalman filter for speech enhancement. In: 2020 14th International Conference on Signal Processing and Communication Systems. ICSPCS, pp. 1–7. http://dx.doi.org/10.1109/ICSPCS50536.2020.9310011.

Roy, S.K., Paliwal, K.K., 2020b. Deep residual network-based augmented Kalman filter for speech enhancement. In: 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. APSIPA ASC, pp. 667–673.

Saki, F., Kehtarnavaz, N., 2016. Automatic switching between noise classification and speech enhancement for hearing aid devices. In: 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. EMBC, pp. 736–739. http://dx.doi.org/10.1109/EMBC.2016.7590807.

Saki, F., Sehgal, A., Panahi, I., Kehtarnavaz, N., 2016. Smartphone-based real-time classification of noise signals using subband features and random forest classifier. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 2204–2208. http://dx.doi.org/10.1109/ICASSP.2016.7472068.

Snyder, D., Chen, G., Povey, D., 2015. MUSAN: A music, speech, and noise corpus. CoRR, abs/1510.08484, arXiv:1510.08484.

Srinivasan, S., Samuelsson, J., Kleijn, W.B., 2006. Codebook driven short-term predictor parameter estimation for speech enhancement. IEEE/ACM Trans. Audio Speech Lang. Process. 14 (1), 163–176. http://dx.doi.org/10.1109/TSA.2005.854113.

Taal, C.H., Hendriks, R.C., Heusdens, R., Jensen, J., 2011. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. IEEE/ACM Trans. Audio Speech Lang. Process. 19 (7), 2125–2136. http://dx.doi.org/10.1109/TASL.2011.2114881.

Varga, A., Steeneken, H.J., 1993. Assessment for automatic speech recognition: Ii. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems. Speech Commun. 12 (3), 247–251. http://dx.doi.org/10.1016/0167-6393(93)90095-3.

Vaseghi, S.V., 2006. Advanced Digital Signal Processing and Noise Reduction. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., 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, Vol. 30. Curran Associates, Inc., pp. 5998–6008.

Veaux, C., Yamagishi, J., MacDonald, K., 2019. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. http://dx.doi.org/10.7488/ds/2645, University of Edinburgh, the Centre for Speech Technology Research (CSTR).

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. http://dx.doi.org/10.1109/TASLP.2014.2352935.

Wang, Y., Wang, D., 2013. Towards scaling up classification-based speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. 21 (7), 1381–1390. http://dx.doi.org/10.1109/TASL.2013.2250961.

Williamson, D.S., Wang, Y., Wang, D., 2016. Complex ratio masking for monaural speech separation. IEEE/ACM Trans. Audio Speech Lang. Process. 24 (3), 483–492. http://dx.doi.org/10.1109/TASLP.2015.2512042.

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. http://dx.doi.org/10.1109/LSP.2013.2291240.

Yu, H., Ouyang, Z., Zhu, W., Champagne, B., Ji, Y., 2019. A deep neural network based Kalman filter for time domain speech enhancement. In: IEEE International Symposium on Circuits and Systems. pp. 1–5. http://dx.doi.org/10.1109/ISCAS.2019.8702161.

Yu, H., Zhu, W.-P., Champagne, B., 2020. Speech enhancement using a DNN-augmented colored-noise Kalman filter. Speech Commun. 125, 142–151. http://dx.doi.org/10.1016/j.specom.2020.10.007.

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. http://dx.doi.org/10.1109/TASLP.2020.2987441.

Zheng, N., Zhang, X., 2019. Phase-aware speech enhancement based on deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. 27 (1), 63–76. http://dx.doi.org/10.1109/TASLP.2018.2870742.