IEEE Open Journal of
Circuits and Systems

Received 20 October 2023; revised 30 December 2023 and 28 February 2024; accepted 8 April 2024. Date of publication 16 April 2024; date of current version 3 May 2024.

Digital Object Identifier 10.1109/OJCAS.2024.3389100

Hardware Efficient Speech Enhancement With Noise Aware Multi-Target Deep Learning

SALINNA ABDULLAH[1] (Graduate Student Member, IEEE), MAJID ZAMANI[1,2] (Member, IEEE), AND ANDREAS DEMOSTHENOUS[1] (Fellow, IEEE)

[1]Department of Electronic and Electrical Engineering, University College London, WC1E 7JE London, U.K.

[2]School of Electronics and Computer Science, University of Southampton, SO17 1BJ Southampton, U.K.
This article was recommended by Associate Editor J. McAllister.

CORRESPONDING AUTHOR: S. ABDULLAH (e-mail: salinna.abdullah.13@ucl.ac.uk)

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R512400/1.

ABSTRACT This paper describes a supervised speech enhancement (SE) method utilising a noise-aware four-layer deep neural network and training target switching. For optimal speech denoising, the SE system, trained with multiple-target joint learning, switches between mapping-based, masking-based, or complementary processing, depending on the level of noise contamination detected. Optimisation techniques, including ternary quantisation, structural pruning, efficient sparse matrix representation and cost-effective approximations for complex computations, were implemented to reduce area, memory, and power requirements. Up to 19.1x compression was obtained, and all weights could be stored on the on-chip memory. When processing NOISEX-92 noises, the system achieved an average short-time objective intelligibility (STOI) and perceptual evaluation of speech quality (PESQ) scores of 0.81 and 1.62, respectively, outperforming SE algorithms trained with only a single learning target. The proposed SE processor was implemented on a field programmable gate array (FPGA) for proof of concept. Mapping the design on a 65-nm CMOS process led to a chip core area of 3.88 mm^2 and a power consumption of 1.91 mW when operating at a 10 MHz clock frequency.

INDEX TERMS Deep neural network, digital circuits, field programmable gate array (FPGA), mapping, masking, multi-target learning, speech enhancement, structured pruning, ternary quantisation.

I. INTRODUCTION

THE USE of artificial neural networks, for example, deep neural networks (DNNs) [1], recurrent neural networks (RNNs) [2], generative adversarial networks (GANs) [3] and convolutional neural networks (CNN) [4], [5] in speech enhancement (SE) algorithms, can provide significant improvement in denoising performance compared to conventional non-deep learning-based methods [6], [7]. Supervised deep learning SE algorithms use a data-driven approach to derive enhanced speech from noisy speech. Learning targets, usually two-dimensional time-frequency (T-F) masks or spectrograms of clean speech, are often used for learning the necessary mapping or processing to convert noisy speech into clean speech.

Different learning targets exhibit complementary properties at different signal-to-noise ratio (SNR) levels and when processing different noise types. This is suggested by studies which found that mapping targets are preferred over masking targets at low SNR conditions [4], [6]. To exploit the complementary properties of different learning targets, a deep learning framework with multiple-target joint learning is needed. Multi-target learning [7], an example of multi-task learning [8], has attracted increased research interest due to its ability to predict multiple targets simultaneously and to solve challenging problems where multiple conditions are considered. Single jointly learned neural network models exploit shared knowledge across relevant targets and capture inter-target correlations, leading to improved regression performance. Furthermore, they achieve comparable performance with ensemble networks while benefitting from significantly smaller model sizes and lower computational complexity [2].

Hardware implemented SE processors have been reported [5], [9], [10], [11]. In [5], hardware sharing was used to reduce the SE processor area by leveraging the high similarity between CNN and fast Fourier transform (FFT) computations. Low-rank expansion and weight quantisation were used to further compress the CNN-FFT model. For quantisation, a 4-bit index was used to indicate 16-bit weight values, providing 75% memory reduction. Weight pruning was not extensively explored in [5]. In [9], the SE processor contains multi-modal speech selection, lookup table-based (LUT-based) non-linear approximation circuits and speech detection controlled dynamic clock gating. The non-deep learning-based processor relies on independent component analysis to separate noisy inputs into statistically independent signals. Although the SE processor in [9] is compact and performs well in multi-talker scenarios, it struggles to effectively remove unvoiced noise. The non-deep learning-based SE processor in [10] achieved a 34% reduction in both power consumption and area footprint by employing a more efficient spectral-change enhancement technique. This was further optimised by using 6-bit coefficients in a bandpass filter. Coefficient sharing reduced execution time by 73.4% and 48.9% for spectral smoothing and convolution in the difference-of-Gaussian function, respectively. A DNN-based SE processor that uses adaptive step-size-based slope and intercepting to approximate the sigmoid function was described in [11]. A common drawback in all these deep learning-based SE processors is that they only adopt a single type of processing for all noise conditions. In addition, there remains the challenge of reducing computational complexity and memory, particularly for applications where a compact design is required.

In this paper, a compact and low-power deep learning-based SE processor with speech intelligibility and quality improvements in different acoustic noise environments is proposed. The novel SE method utilises both mapping and masking learning targets, where the switching between the learning targets or a complementary approach combining both targets is made possible through multi-target learning. It incorporates dynamic noise level sensing (DNLS) to provide information on temporal noise changes. The DNLS also provides voice activity detection (VAD) to improve noise estimation, and to suppress burst and non-stationary noise more effectively. The DNN is made more compact by using a customised ternary quantisation model that is optimised by the weight distributions of the SE network, and a structured weight pruning scheme that is compatible with the sparse ternary network. To estimate on-chip power and area requirements for integrated implementation, the proposed SE method was implemented on an FPGA platform and synthesised in a 65-nm CMOS process.

The major contributions of this paper are as follows:

(i) A noise-aware (through DNLS) SE approach with multi-target learning is introduced, featuring bespoke schemes for dynamically switching between mapping-based, masking-based, and combined processing. This adaptability significantly improves denoising performance under varying noise conditions.

(ii) A dynamic ternary quantisation approach, optimised for SE, that employs statistical distribution to enhance computational efficiency while maintaining high denoising quality.

(iii) An adaptive structural pruning technique for feedforward fully connected layers is proposed, specifically designed to complement the ternary network.

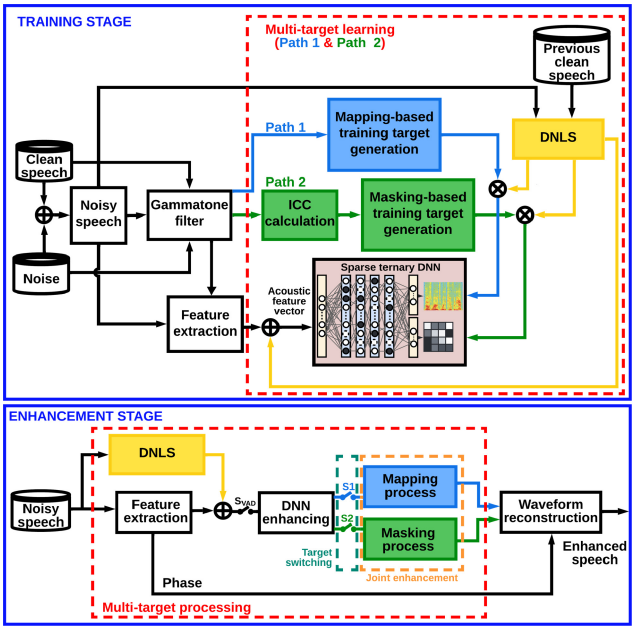


FIGURE 1. High-level system overview of the proposed SE method with multi-target learning.

The rest of this paper is organised as follows. Section II describes the proposed SE design and its major functional units. Section III describes the ternary quantisation and structured pruning optimisation techniques. Section IV presents the FPGA implementation. The datasets used for evaluation, discussion of the measured results and comparison with other work are presented in Section V. Concluding remarks are drawn in Section VI.

II. SPEECH ENHANCEMENT WITH MULTI-TARGET LEARNING

Fig. 1 presents an overview of the proposed multi-target learning for SE. During the training phase, clean speech and noise from the training dataset are merged to produce a noisy speech signal. Feature extraction is then performed on the noisy speech signal. The extracted features, combined with the noise estimate from the DNLS, form the acoustic feature vector for DNN training across various noise types and levels. The DNN training utilises two targets: the mapping-based target (Path 1 in Fig. 1) and the masking-based target (Path 2 in Fig. 1). The SE system employs the DNLS noise estimate to allocate weight coefficients, activating three training modes:

- Mode 1 focuses on mapping target training and involves computing the cochleagram of the clean speech. Mode 1 utilises only Path 1 in Fig. 1.
- Mode 2 emphasises masking target training and involves calculating correlation-based masks using both clean speech and noise. Mode 2 utilises only Path 2 in Fig. 1.
- Mode 3 (simultaneously activating Paths 1 and 2) integrates both mapping and masking training for multi-target learning.

During the enhancement phase, the trained DNN is activated using the DNLS. If speech is detected, the S_{VAD} switch in Fig. 1 engages, allowing the input frame to proceed to the subsequent enhancement blocks. An acoustic feature vector is extracted from the input frame and then processed by the trained DNN to either derive the denoised cochleagram or estimate the ratio mask for the noisy cochleagram. Switches S1 and S2 in Fig. 1 control this process. When S1 is active, the frame undergoes a direct mapping process (Mode 1) to produce an enhanced cochleagram. This mode modifies the noisy speech to yield an enhanced speech cochleagram, which is then reconstructed into a waveform. Activating S2 results in a pure masking enhancement (Mode 2), where the noisy speech cochleagram is overlaid with the DNN-estimated mask. The resultant output is then reconstructed into its time-domain representation. Lastly, joint processing (Mode 3) combines both masking and mapping approaches, initially determining the enhanced cochleagram and subsequently refining it with the enhancement mask. Further explanation is outlined in Section II-B on how the modes and switches are activated using the DNLS block.

A. ACOUSTIC FEATURE EXTRACTION

As shown in Fig. 1, the noisy speech frame is applied to the feature extraction unit, before DNN training and inference in the training and enhancement stages. Each signal frame spans 25 ms and is sampled at a rate of 16 kHz, resulting in 400 samples per frame. Adjacent frames share an overlap of 15 ms, equivalent to 240 samples.

In this work, the amplitude modulation spectrogram (AMS) and Gammatone frequency cepstral coefficients (GFCC) features [12] are used to form the acoustic feature set, because they offer a synergistic balance between computational efficiency and effective speech signal representation. AMS features are adept at capturing temporal modulation characteristics, which are vital for maintaining speech intelligibility in noisy environments, while GFCC features, derived from the Gammatone filterbank – also employed in training target generation – provide a physiologically relevant analysis of speech that closely mimics the human auditory response.

The Gammatone filters are expressed by:

$$g(t) = a_f t^{n-1} e^{1-2\pi b_f t} \cos(2\pi f_c t + \phi) \tag{1}$$

where a_f is a constant that controls the filter gain, t is time, n is the order of the filter, f_c is the filter central frequency and ϕ is the phase. The decaying factor b_f determines the filter bandwidth. The Gammatone filter centre frequencies are equally distributed on the equivalent rectangular bandwidth scale (ERB) defined as:

$$\text{ERB} = 24.7\left(4.73\frac{f_c}{1000} + 1\right) \tag{2}$$

Reconstructing the time-domain speech signal from the cochleagram is done indirectly through several stages of inversion, which include performing short-time auto-correlation on all outputs of the cochleagram [13].

The combination of AMS and GFCC features avoids the need for multiple filterbanks, such as the Mel filterbank for Mel frequency cepstral coefficients (MFCCs) and the Bark filterbank for relative spectral transformed perceptual linear prediction coefficients (RASTA-PLP), thereby reducing computational complexity. An empirical analysis also further substantiates that the integration of AMS and GFCC features achieves the best tradeoff between SE performance and computational demand.

The AMS computation involves extracting the signal's envelope using full-wave rectification before decimation by a factor of 4 is applied. The decimated envelope is then Hanning windowed, zero-padded and then integrated by 15 triangular windows uniformly spaced from 15.6 Hz to 400 Hz. This produces a 15-D AMS feature vector. To obtain the GFCC features, the signal is first decomposed using a 64-channel Gammatone filter before it is decimated to an effective sampling rate of 100 Hz. The output subsequently goes through loudness compression by a cubic operation, followed by a discrete cosine transform to yield a 31-D GFCC vector. In addition to the coefficients, the first- and second-order time differences are obtained and concatenated to the feature vector.

B. DYNAMIC NOISE LEVEL SENSING (DNLS)

The introduction of the DNLS method is primarily motivated by its capability to dynamically adapt to temporal variations in noise characteristics, a feature essential for real-time SE applications. Prior works on noise estimation include [31] and [32]; the proposed DNLS extends these concepts by incorporating real-time adaptability and integration with multi-target deep learning frameworks for SE. Furthermore, it has been designed to be computationally efficient whilst effective at providing a responsive approach that improves the SE performance in non-stationary noise.

The operations of the proposed DNLS are shown in Fig. 2. The DNLS estimates the noise level within a noisy speech frame. In Fig. 2(a) the DNLS block calculates the weighted average noise level estimate, N'_k. This noise level estimate is utilised to compute the weight coefficient γ, which determines the training mode. Under high noise levels (or low SNR conditions), the preference is for mapping target learning. Consequently, the weight coefficient γ should lean towards the mapping target or complementary target training, with greater emphasis on the mapping target.

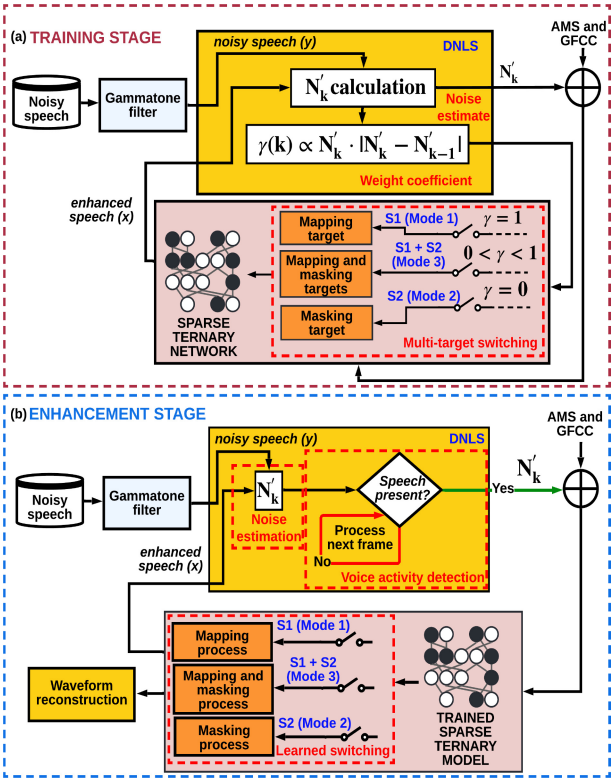


FIGURE 2. (a) The operation of the DNLS in the training stage includes noise estimation and calculation of the weight coefficients for multi-target training. (b) The operation of the DNLS in the enhancement stage includes voice activity detection and noise estimation.

The weighted average noise level estimate N'_k necessitates the computation of the current noise level estimate N_k. The noise level for the current 25 ms frame is given by:

$$N_k = 1 - \frac{R(j, j-1)}{R(j-1, j-1)} \tag{3}$$

where N_k represents the estimated noise level for frame k and $R(j,\ j-1)$ is the normalised cross-correlation between the current segment j and the preceding segment $j-1$. This equation gauges the reduction in autocorrelation from segment $j-1$ to segment j, offering a noise level estimate.

The noise estimate is continuously updated over time, allowing the algorithm to adjust to noise level variations. This is achieved using a recursive averaging technique, where the new noise estimate N'_k is a weighted average of the prior estimate and the new measurement. The update equation is:

$$N'_k = \ \varphi N_k + (1-\varphi) N'_{k-1} \tag{4}$$

where φ is a weighting factor that determines the influence of the new and old noise estimates. It is dynamically determined during the training process to ensure optimal noise estimation adaptability. If φ is close to 0, the old estimate exerts more influence, causing the noise estimate to evolve slowly. This is suitable when the noise is relatively stationary (determined by $|N'_k - N'_{k-1}|$) and exhibits gradual changes. Conversely, if φ approaches 1, the new measurement has more influence, allowing the noise estimate to shift swiftly. This is appropriate when the noise is highly non-stationary and can change quickly over time.

An adaptive weight coefficient γ is introduced based on the noise estimation from the DNLS and $|N'_k - N'_{k-1}|$, where a large $|N'_k - N'_{k-1}|$ value denotes a non-stationary noise:

$$\gamma(k) \propto N'_k \cdot \left|N'_k - N'_{k-1}\right|. \tag{5}$$

Empirical tests revealed that beyond or below specific noise levels, either purely mapping-based enhancement or purely masking-based enhancement outperforms complementary enhancement. As a result, the weighting coefficient γ is assigned 1 above an upper threshold, thr_{up}, and 0 below a lower threshold, thr_{low}, as follows:

$$\gamma'^{(k)} = \begin{cases} 1 & \gamma(k) > thr_{up} \\ 0 & \gamma(k) < thr_{low}. \end{cases} \tag{6}$$

The thresholds are normalised to maintain consistency with the normalised speech and noise signals used in the training and inference stages. From testing, thr_{up} was established to provide the optimal performance at 0.85 whereas thr_{low} was at 0.15. When $\gamma = 1$, switch S1 (as shown in Fig. 2) is activated, allowing the noisy speech to undergo purely mapping-based enhancement (Mode 1). When $\gamma = 0$ only the masking-based target learning is activated (Mode 2). For conditions where $0 < \gamma < 1$, the network undergoes complementary processing (Mode 3) where both mapping and masking-based SE are jointly learned. Fig. 3 shows how γ, N'_k, N_k and φ vary with an example speech waveform contaminated by factory noise at 0 dB SNR, demonstrating its responsiveness to noise levels. Note that φ has been downscaled by 90x to prevent the occlusion of the other plots.

At the enhancement stage, shown in Fig. 2(b), the DNLS block additionally performs VAD such that enhancement is activated only when speech is present. This is achieved by thresholding the noise level estimate N'_k. Noise-only frames are indicated by high values of N'_k, whereas silent frames are usually characterised by N'_k values of zero. The threshold identifying noise-only frames is determined at the training stage, where noise-only frames are known and correlated to N'_k. The ROC curve depicted in Fig. 4 effectively demonstrates the reliability of the proposed VAD approach when presented with speech contaminated with noise at -5 dB SNR. In the figure, the positive rates indicate noise-only frame detections. As illustrated, the chosen threshold is the one that results in minimal false alarms, a critical factor in determining whether a frame is noise-only. This careful selection ensures that the VAD algorithm tends towards activating the SE process. This approach prioritises the quality of the hearing experience over computational simplicity.

C. SPEECH ENHANCEMENT MODES

Three modes (Mode 1, Mode 2, and Mode 3) in conjunction with the DNLS are designed to perform deep SE.

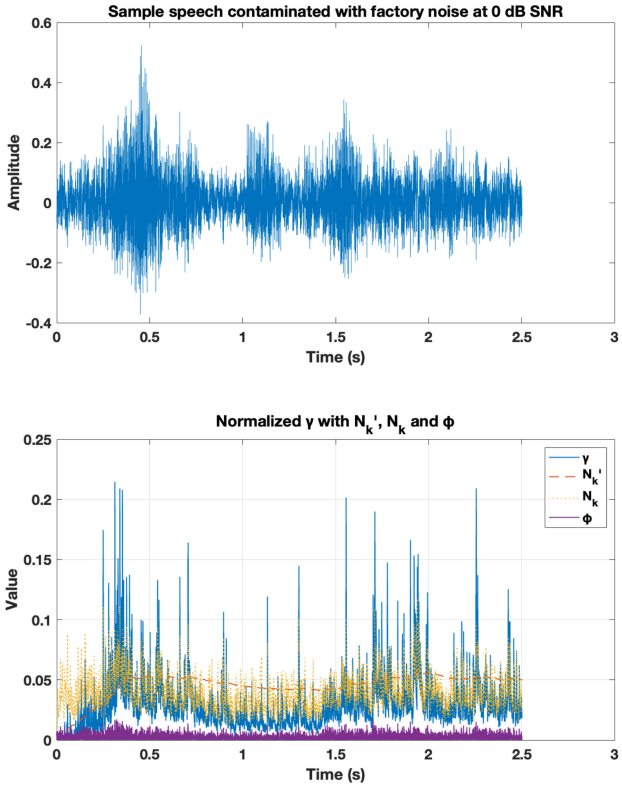


FIGURE 3. Variation of γ, N_k', N_k and φ in response to a speech waveform contaminated by factory noise at 0 dB SNR.

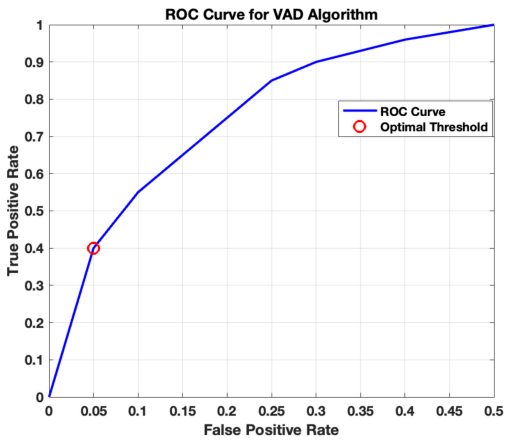


FIGURE 4. ROC curve illustrating the performance of the VAD algorithm. The curve highlights the trade-off between false positive rates and true positive rates, with the optimal threshold marked to minimise false alarms in noise-only detection.

MODE 1: MAPPING-BASED TRAINING TARGET

The activation of S1 leads to Mode 1 processing, which is the mapping-only enhancement. This enhancement maps the noisy speech cochleagram directly to an enhanced one. The cochleagram is derived from the Gammatone filters described in Section II-A. The output of each filter is visualised over time, creating a T-F representation of the sound. Each row in the cochleagram corresponds to one Gammatone filter (one frequency band), and the intensity of the colour or grayscale in the visualisation represents the energy at each frequency band over time.

MODE 2: MASKING-BASED TRAINING TARGET

A cost-effective adaptation of the ideal ratio mask (IRM) employing inter-channel correlation (ICC) factors [14] is used in the masking-based target construction. To minimise computational expense, the ICC factors are determined using pre-established sum tables, as detailed in [14].

Traditionally, the ICC factor computation involves determining the speech energy $P_x(c, m)$ and noise energy $P_n(c, m)$ of the mth frame in the cth channel:

$$ICC_{IRM(c,m)} = \frac{\rho_x(c,m) \cdot P_x(c,m)}{\rho_x(c,m) \cdot P_x(c,m) + \rho_n(c,m) \cdot P_n(c,m)} \quad (7)$$

where $\rho_x(c, m)$ is the normalised cross-correlation (NCC) coefficient between the clean speech and noisy speech power spectra in the cth channel of the mth frame and $\rho_n(c, m)$ is the NCC coefficient between the noise and noisy speech power spectra in the cth channel of the mth frame. They are given by:

$$\begin{cases} \rho_n(c,m) = \dfrac{y_{c,m}^T \cdot n_{c,m}}{\sqrt{\|y_{c,m}\|^2 \cdot \|n_{c,m}\|^2}}, (a) \\ \rho_x(c,m) = \dfrac{y_{c,m}^T \cdot x_{c,m}}{\sqrt{\|y_{c,m}\|^2 \cdot \|x_{c,m}\|^2}}, (b) \end{cases} \quad (8)$$

where $y_{c,m}$, $n_{c,m}$ and $x_{c,m}$ are the magnitude spectrum column vectors of noisy speech, pure noise, and clean speech for each frame in each Gammatone channel respectively. T denotes the transpose operation.

The sum tables method eliminates redundant calculations and approximates cross-correlation with energy differencing between relevant signal windows. Given that many calculations are repetitive due to the comprehensive comparison between noisy speech as reference windows ($y_{c,m}$) and clean speech ($x_{c,m}$) or pure noise ($n_{c,m}$) as comparison windows, the sum tables method replaces squared and dot product operations with simpler subtractions and additions. This substantially reduces the computational complexity for calculating $\rho_n(c, m)$, $\rho_x(c, m)$ and ICCs.

D. MODE 3: MULTI-TARGET LEARNING

The DNN features two output layers for multi-target (or joint) learning. One layer produces the enhanced cochleagram, while the other generates the predicted correlation-based ratio mask to attain clean speech, as shown in Fig. 1. Employing a shared DNN allows the network to discern the relationship between mapping and masking-based processes. It further enables generalisation capability improvement by integrating multiple regularisation elements. Moreover, using a singular DNN for multi-target and multi-task learning results in a more compact model with lower computational demands compared to deploying multiple DNNs.

During the joint learning of mapping and masking-based targets, the minimum mean square errors (MMSEs) from

both outputs guide the adjustment of network parameters in training:

$$Err = \gamma * \frac{1}{K}\sum_{k=1}^{K}\frac{\left\|\mathbb{X}^{map} - \boldsymbol{X}_k^{map}\right\|_2^2}{\left\|\boldsymbol{X}_n^{map}\right\|_2^2} + (1-\gamma) * \frac{1}{K}\sum_{k=1}^{K}\frac{\left\|\mathbb{X}^{mask} - \boldsymbol{X}_k^{mask}\right\|_2^2}{\left\|\boldsymbol{X}_k^{mask}\right\|_2^2} \quad (9)$$

$$\mathbb{X}^{map} = \widehat{\boldsymbol{X}}_k^{map}\left(\boldsymbol{Y}_{k\pm r}^{map}, \boldsymbol{Y}_{k\pm r}^{mask}, \ \boldsymbol{W}, \boldsymbol{b}\right) \quad (10)$$

$$\mathbb{X}^{mask} = \widehat{\boldsymbol{X}}_k^{mask}\left(\boldsymbol{Y}_{k\pm r}^{map}, \boldsymbol{Y}_{k\pm r}^{mask}, \ \boldsymbol{W}, \boldsymbol{b}\right) \quad (11)$$

where $\widehat{\boldsymbol{X}}_k$ and $\boldsymbol{X}_k$ represent the estimated and clean cochleagrams at sample index k, respectively. K denotes the mini-batch size. $\boldsymbol{Y}_{k\pm r}$ is the noisy speech feature vector where the window size is $2*\tau+1$. $(\boldsymbol{W}, \boldsymbol{b})$ denotes the weight and bias parameters within the network. The superscripts '*map*' and '*mask*' refer to the cochleagram-based mapping target and correlation-based masking target, respectively.

III. DNN OPTIMISATION TECHNIQUES

A pre-trained DNN with a restricted Boltzmann machine [14] employing four hidden layers with 1024 nodes each is utilised for SE in this work. A standard backpropagation algorithm using the normalised MMSE cost function is considered for weight tuning across learning iterations. A typical DNN often demands more than 10 MB of memory when weights are stored in the floating-point format. To address this, this work introduces a two-fold strategy to reduce the DNN's memory requirements: (i) dynamic layer-wise ternary quantisation of parameters, minimising both the memory footprint for storage and the memory bandwidth for parameter retrieval, and (ii) compressing the architecture of the feedforward ternary network through structured pruning.

A. TERNARY QUANTISATION

This work uses a dynamic ternary quantisation approach to reduce the neural network weight precision to three values: $\{+1,0,-1\}$, with each ternary weight represented by 2 bits. The approach concurrently trains network weights, quantisation thresholds and a layer-wise scaling factor. It bridges the gap between full precision and quantised weights, mitigating accuracy loss from model compression. Fig. 5 shows the ternary quantisation training process with dynamic quantisation thresholds (instead of employing fixed or uniform ternarisation).

As shown in Fig. 5(a), to achieve dynamic ternarisation, the kernel density estimation (KDE) evaluates the distribution of the network's pre-trained full-precision weights. The KDE for the network's pre-trained full-precision weights is:

$$KDE(x) = \frac{1}{\mathbb{N}B}\sum_{i=1}^{\mathbb{N}}\frac{K(x-W_i)}{B} \quad (12)$$

where x represents the weight value at which the probability density function is calculated, $\mathbb{N}$ is the number of weights in the layer, W_i indicates the layer weights, K is a non-negative kernel function and B is a smoothing parameter called the bandwidth. In this work, the normal kernel is used. The bandwidth B is set to the Silverman's Rule of Thumb [15].

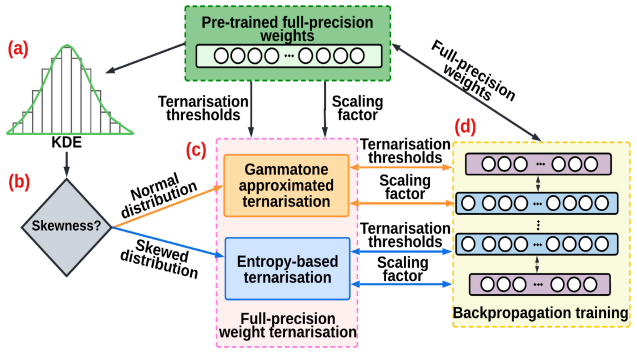


FIGURE 5. The dynamic ternary quantisation process where the weight distribution at pre-training determines the ternarisation method to be employed. The ternarisation of the full precision weight is executed with respect to the ternarisation thresholds and scaling factor.

The nature of the weight distribution (i.e., normal or skewed distribution) is then determined by assessing the skewness of the KDE as shown in Fig. 5(b):

$$Skew = \frac{\sum_{i=1}^{\mathbb{N}}\left(KDE_i - \overline{KDE}\right)^3}{(\mathbb{N}-1)\sigma^3} \quad (13)$$

A normal (symmetric) distribution is defined when the probability density function possesses a skewness between -0.5 and 0.5. For a symmetric weight distribution, a truncated Gaussian approximated ternarisation is performed as shown in Fig. 5(c). In this case, the ternarisation thresholds $(\theta_l^{\pm})$ are set to be where the KDE reaches a certain fraction of its maximum value. The ternarisation thresholds determine the boundaries for quantising the full-precision weights into ternary values.

The layer-wise ternarisation thresholds for a symmetric distribution are calculated by the weight ternarisation function:

$$W'_{l,i} = \zeta_l(KDE_l)\cdot Tern\left(W_{l,i}, \theta_l^{\pm}\right) = \zeta_l(KDE_l)\cdot\begin{cases}+1, & W_{l,i} > \theta_l^{+}\\ 0, & \theta_l^{-} \le W_{l,i} \le \theta_l^{+}\\ -1, & W_{l,i} < \theta_l^{-},\end{cases} \quad (14)$$

where $W_{l,i}$ and $W'_{l,i}$ denote the full-precision and ternarised version of the weight at index i of layer l, respectively. ζ_l is the layer-wise scaling factor calculated using the extracted mean KDE_l. $\theta_l^{\pm}$ represents the ternarisation thresholds in each layer. For asymmetric or skewed distributions, the ternarisation thresholds are determined by identifying the three primary weight clusters using weight entropy [16]. Additionally, the straight-through estimator [17] is employed to enhance gradient approximation for the non-differentiable ternarisation function, ensuring quicker convergence and high inference accuracy.

The scaling factor, acting as a multiplier, adjusts the magnitude of the ternarised weights to ensure they are

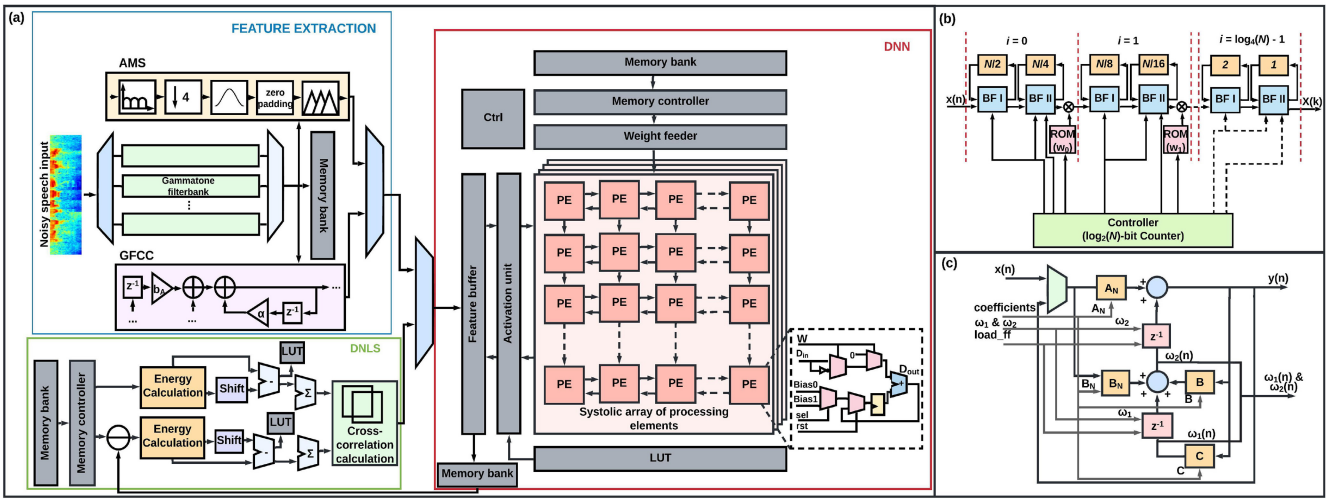


FIGURE 6. (a) Architecture of the proposed SE inference processor. The training target generation unit is not part of the SE enhancement (inference) stage. (b) The Radix-2^2 FFT used for obtaining the AMS features. (c) Structure of a single SOS IIR stage for the Gammatone filterbank.

representative of the original full-precision weights. The full-precision weights, ternarisation thresholds and scaling factor are all integral to the backpropagation training shown in Fig. 5(d).

B. STRUCTURAL PRUNING

Group lasso is a regularisation technique that promotes sparse structured pruning [18]. However, simply merging group lasso-based structured pruning with ternary quantisation can hinder performance and slow down training convergence [19]. This is because while group lasso aims to reduce weight mean and variance during training, weight ternarisation seeks to broaden the weight distribution.

Structured pruning can be regulated to encourage meaningful group-wise sparsity patterns (e.g., channel-wise or kernel-wise sparsity) that harmonise with the ternarised DNN. If the intra-group L_2-norm of a weight cluster is significant, it indicates the cluster contains crucial weights. Such clusters should be exempt from group lasso pruning during backpropagation training. To achieve this, weight penalty clipping [19] is employed for more optimal pruning of the ternary network. The weight penalty clipping method in [19] is adapted for fully connected feedforward DNNs in this work as:

$$\widehat{\mathcal{L}} = \mathcal{L}\Big(f\Big(x; Tern\{W_l\}_{l=1}^{L}\Big), t\Big) + \lambda \sum_{l=1}^{L}\sum_{i=1}^{G_l} min\Big(\big\|W_{l,i}\big\|_2, \delta_l\Big) \tag{15}$$

$$\delta_l = \eta \cdot \frac{1}{G_l}\sum_{i=1}^{G_l}\big\|W_{l,i}\big\|_2 \tag{16}$$

where $\mathcal{L}(.,.)$ is the objective function of the DNN, which is the normalised MMSE in this work, and $\widehat{\mathcal{L}}$ is the new loss function added to the backpropagation training. $f(x; Tern\{W_l\}_{l=1}^{L})$ computes the ternarised weights of the DNN with respect to the input x. $\|W_{l,i}\|_2$ calculates the intra-group L_2-norm (Euclidean norm) of the indexed weight group $W_{l,i}$. G_l is the number of groups in the lth layer, and λ is a tuneable hyperparameter. δ_l is a layer-wise self-adaptive threshold used to diminish the L_2-norm penalty on large weights. η denotes a scaling coefficient that is found empirically.

There are two scenarios in the weight penalty clipping process: (i) if $\|W_{l,i}\|_2 \geq \delta_l$, weight clipping is performed to prevent relatively large weights from being pruned from the network, replacing $\|W_{l,i}\|_2$ with δ_l; (ii) if $\|W_{l,i}\|_2 < \delta_l$, weight clipping is not in effect, and the weight penalty remains $\|W_{l,i}\|_2$. To determine whether a single neuron should belong to a group (one of G_l) within a fully connected layer, the average intra-group L_2-norm is assessed whenever a new vector-wise neuron is added. If adding a neuron significantly reduces the average intra-group L_2-norm, it suggests the neuron is unimportant and should be considered part of the weight group.

IV. FPGA IMPLEMENTATION

The proposed supervised SE method with multi-target learning was adapted for FPGA via a multi-step process. Since many complex operations cannot be directly and easily implemented on an FPGA with reasonable resource utilisation, formulae-based approximations or LUTs were often used in replacements. The coordinate rotation digital algorithm (CORDIC) [20] was employed to compute square roots, multiplications, divisions, exponentials, and logarithms to avoid using area-inefficient digital signal processor (DSP) slices on the FPGA.

Fig. 6(a) shows the architecture of the SE inference processor. The input signal undergoes preprocessing via the I/O data buffer and Hanning window before being directed to the DNLS, acoustic feature extraction or deep learning unit. The input features are processed by the DNLS unit and then by a systolic array for DNN operations. To achieve the sigmoid activation function, an LUT with pre-calculated

results is used. Interpolation between two pre-calculated results is obtained when an intermediary output is requested.

The FFT for AMS features was implemented using the Radix-2^2 method [21]. With the Radix-2^2 method, the r-point FFT processor has a log4(r) number of stages, as shown in Fig. 6(b). BF I and BF II in the figure represent a Radix-2 butterfly and a Radix-2 butterfly with trivial multiplication by $-j$, respectively. The multiplication by $-j$ involves real-imaginary swapping and sign inversion. The Gammatone filter was implemented by a cascade of second-order section (SOS) IIR bandpass filters [22]. Fig. 6(c) shows the structure of a single SOS IIR stage. The second order IIR stage composes four 16 x 16-bit multipliers, two adders and two flip-flops. The filter coefficients (A_N, B_N, B and C) are stored on BRAM and loaded from the BRAM into the registers when updated. The internal variables ($\omega 0$ and $\omega 1$), also stored on the BRAM, are loaded into the flip-flops when the load_ff signal is high.

Storing the network weights on the external DRAM (or, more specifically, the DDR2 synchronous DRAM of the Nexys A7) would not be efficient due to longer access time and significantly higher power consumption, especially when the weights are only fetched once for each output computation. Through the ternary quantisation, the bit-width of the DNN was reduced to 2-bit from a 32-bit full precision representation. This optimisation allowed weight storage on the on-chip BRAM and simplified weight operations. Post-structural pruning, the 2-bit representation is further streamlined to a 1-bit format, achieved by eliminating non-zero weights.

Although ternary quantisation and structured pruning can greatly reduce memory footprint, there were some challenges. Since only non-zero weights were stored in memory, relative row index and column pointer were needed to store the sparse matrix. A 5-bit data was used to store the network weight, with 1 bit for the ternary weights and 4 bits for the relative row index. Fig. 7(a) shows the compressed storage format employed, including zero-padding used to encode the weight locations. A column in the weight matrix is located through a pointer, and the absolute addresses of weights are calculated by accumulating the relative indexes.

Load imbalances were introduced in the ternarised sparse DNN as some processing elements may face longer waiting periods due to having fewer non-zero weights, as shown in Fig. 7(b). To address this, a FIFO, built on the distributed RAM was used. It allowed fast-processing elements to retrieve new elements from the FIFO without being hindered by slow-processing elements. The FIFO was 16-bit; its depth could be adjusted from 1 to 16. The utilisation of a FIFO resulted in an approximately 21% increase in throughput and a 14% reduction in latency, indicating a substantial enhancement in performance.

V. PERFORMANCE EVALUATION RESULTS AND ANALYSIS

To evaluate the denoising performance of the SE processor, the short-time objective intelligibility (STOI) [23] and perceptual evaluation of speech quality (PESQ) [24] were employed. The scores were calculated in MATLAB after the enhanced signals were obtained from the processor and transmitted to MATLAB via UART. The SE processor was implemented on a Nexys A7 processing board with a Xilinx Artix-7 XC7A100T-1CSG324C FPGA as a proof-of-concept. Resource utilisation reports were obtained from post-implementation on the Vivado Design Suite, and power and area estimates were acquired when the SE processor design was mapped onto a 65 nm CMOS process.

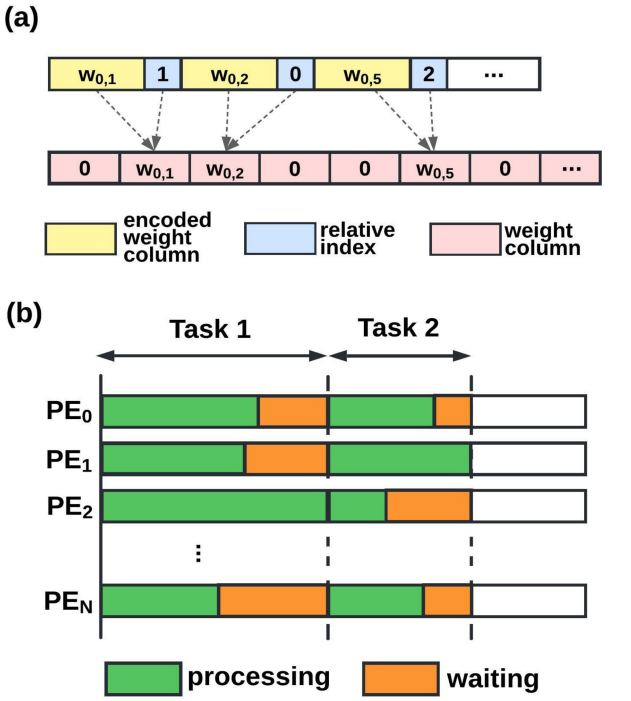


FIGURE 7. (a) Encoding in compressed sparse column with relative index and zero-padding. (b) Example of an imbalanced workload with some processing elements (PEs) having a long wait period.

A. DATASETS

Clean speech utterances from the TIMIT [25] corpus and noise samples from the NOISEX-92 [26] database were used. To develop the training set, 1500 randomly chosen clean utterances were mixed with five types of noise (babble, factory, pink, Volvo (car) and white noise) from the NOISEX-92 database at −5 dB, 0 dB and 5 dB SNR. For the testing set, 100 utterances that had not been chosen for the training set were mixed with the same five types of noise at the same SNR values. The 'f16' and 'factory 2' noise from the same noise database were additionally used to evaluate the generalisation performance of the SE system. Random cuts of the first and last 2 minutes of each noise were used for training and testing, respectively, to avoid using the same noise frames for both training and testing.

B. ABLATION STUDY

The effectiveness of the multi-target learning is summarised in Table 1. The multi-target learning demonstrated

TABLE 1. Ablation study of the effectiveness of multi-target learning.

Training target	Validation accuracy (%)	STOI	PESQ
Unprocessed	-	0.46	0.89
Map-only (Path1)	83.77	0.66	1.39
Mask-only (Path2)	84.22	0.73	1.45
Multi-target learning (Path1 and Path2)	**84.43**	**0.82**	**1.71**

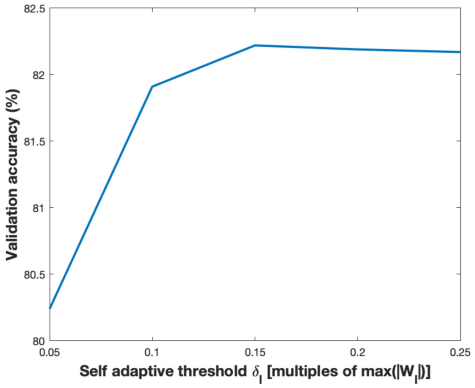


FIGURE 8. Validation accuracy vs. self-adaptive threshold δ_l value. The thresholds are in multiples of max($|W_l|$) (e.g., $\delta_l = 0.05max(|W_l|)$).

TABLE 2. Ablation study of the compression methods.

	Validation accuracy (%)	Overall sparsity (%)	Group sparsity (%)	Compress. rate	PESQ
FP*	84.44	-	-	1.0	1.76
Ternarisation	80.07	60	-	~16×	1.67
Pruning	77.72	43	38	~1.5×	1.65
Naive comb.	75.60	73	16	~17.7×	1.64
Proposed comb.	**80.71**	**64**	**27**	**~19.1×**	**1.71**

*Fully connected full-precision network (baseline).

improvements in average validation accuracy, computed as the proportion of correctly predicted instances over the total number of instances in the validation dataset, as well as denoising performance, evaluated using STOI and PESQ, when compared to training and processing with mapping or masking-only targets. Fig. 8 shows the validation accuracy for different layer-wise self-adaptive thresholds δ_l. $\delta_l = 0.15(|W_l|)$ gave the optimal accuracy.

Table 2 compares the proposed harmonised combination of weight ternarisation and structured pruning against three scenarios: (i) weight ternarisation, (ii) structured pruning, and (iii) naive combination of weight ternarisation and structured pruning. The overall sparsity in Table 2 refers to the percentage of individual zero values within the whole weight tensors. This is different to group sparsity which represents the percentage of the number of channels and frame-wise groups that are all zeros. From Table 2, the proposed approach has the smallest overall but largest useful group sparsity. Compared with the fully connected full-precision network, the compression and quantisation methods generally led to insignificant accuracy loss (< 2%). However,

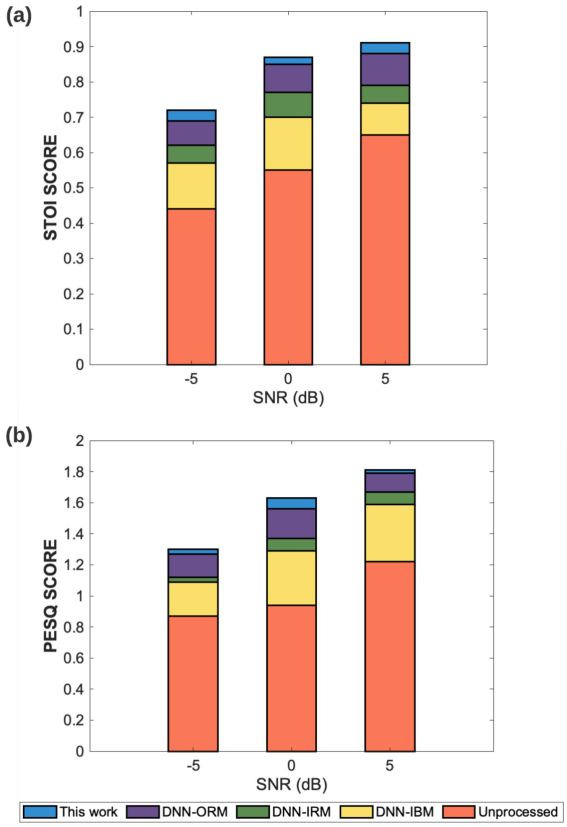


FIGURE 9. (a) STOI scores and (b) PESQ scores obtained from the: (1) unprocessed speech; (2) SE utilising DNN trained with ideal binary mask (DNN-IBM); (3) SE utilising DNN trained with ideal ratio mask (DNN-IRM); (4) SE utilising DNN trained with optimal ratio mask (DNN-ORM); and (4) proposed SE system utilising a hardware efficient DNN with multi-target learning at -5, 0 and 5 dB SNRs.

it was observed that the compression did come with a compromise in terms of validation accuracy and average PESQ scores. Simply combining weight ternarisation and structured pruning produced a more noticeable accuracy deterioration. This highlights the importance of employing a harmonised approach to promote coherent sparsity patterns within the network.

C. SPEECH ENHANCEMENT PERFORMANCE

In addition to the evaluation of SE performance for various training targets (Table 1), the average STOI and PESQ scores of the proposed SE system were compared with those of SE systems utilising fully connected full-precision DNNs trained with: (i) ideal binary masks, (ii) ideal ratio mask and (iii) optimal ratio mask [27]. The comparison was performed after DNN architectural optimisations were applied to the proposed method. The results are shown in Fig. 9. The designed SE demonstrated superior denoising capability with improved STOI and PESQ scores obtained across all SNR values tested despite employing much lesser weight values and sparse connections. Additionally, the proposed approach yielded results that are comparable to those achieved by the recently proposed UNetGAN for time-domain robust speech enhancement, as detailed

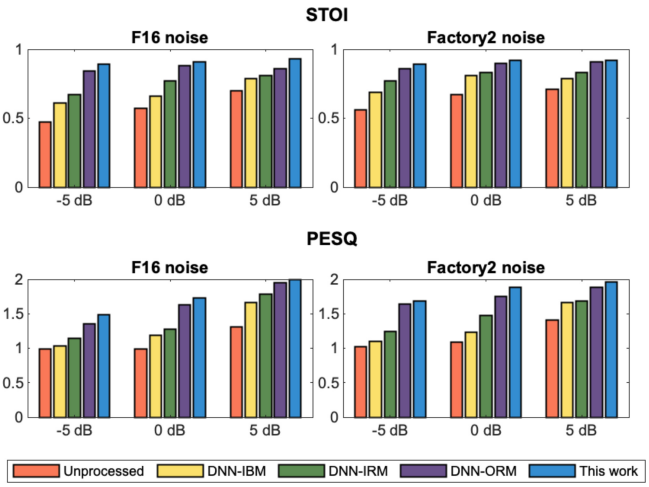


FIGURE 10. STOI and PESQ scores when processing the untrained noise types, 'f16' and 'factory2' noise.

TABLE 3. FPGA resource utilisation for the feedforward DNN with and without optimisation.

Resource	FF	LUT	BRAM*	DSP	DRAM**	Core Area (mm^2)
Unoptimised 32-bit weights	111817	13583	4456	217	8.28	13.29
Optimised with 1-bit weights and pruning	90522	33910	1280	0	1.24	3.88
Available	126800	63400	4860	240	128	-

in [33]. This comparison is particularly relevant as the study involved SE using the NOISEX-92 dataset at similar SNRs. Fig. 10 shows that the proposed system also provided good generalisation performance as favourable enhancement performance continued to be evident when processing the 'f16' and 'factory2' noise.

D. FPGA RESOURCES AND INFERENCE TIME ANALYSIS

The resource utilisation for the designed SE system is summarised in Table 3. To put into perspective the improvement achieved from the design optimisation techniques, the resource utilisation for 32-bit fixed-point weights is given. The unoptimised full-precision network consumed almost the entire FPGA flip flops and DSP slices because 32-bit weights required plenty of hardware multipliers for operation. With the 32-bit implementation, an external DRAM was required for weight storage as the BRAM capacity was insufficient. A small amount of DRAM was required in the optimised system to store incoming signal frames. The major building blocks of the designed SE system were flip flops, LUT slices and BRAM memory. Moreover, the synthesised chip core area with the unoptimised full-precision network was found to be almost 3.5x than that of the optimised version.

Fig. 11 shows the inference latency reduction achieved by optimising the SE system. The largest reduction in latency was achieved in the DNN modified with parameter pruning

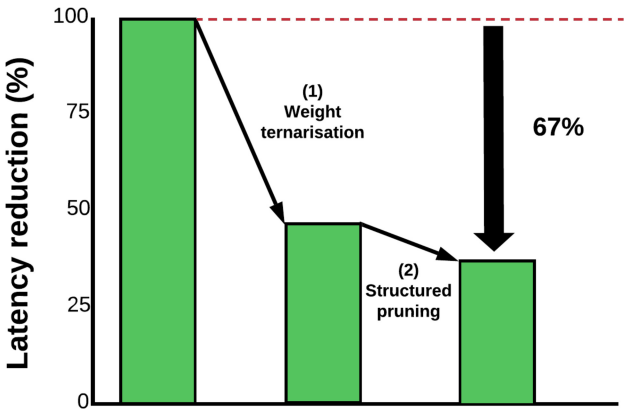


FIGURE 11. Latency reduction by applying (1) weight ternarisation and (2) weight ternarisation optimally combined with structured pruning.

and network weight quantisation. Due to the sparsity of the network, the relative row index was used to indicate weight addresses. Although the relative row index led to more storage required for storing the network weights (1 bit for the ternarised weights and 4 bits for the relative row index), implementing structured pruning was still beneficial since computations within the network were significantly reduced. The optimisations combined achieved an overall latency reduction of 67%.

E. ON-CHIP POWER AND AREA REQUIREMENTS

To estimate the chip resource requirement of the designed system, its architecture was synthesised in TSMC 65 nm technology using Synopsys Design Compiler. The place-and-route was done with Cadence SoC Innovus. A two-step process is used to evaluate the correct operation after Place & Route (P&R). Firstly, timing violations are checked in Innovus. This ensures that all timing constraints are met. Secondly, annotated gate-level simulations are performed to verify the functionality of the design post-P&R.

The correct operation of the entire system was verified at various operating voltages and clock frequencies. The real-time performance at the best STOI score was obtained at a clock frequency of 75 MHz. At this clock frequency, a supply voltage of 1.18 V was required. Under these conditions, the average core power consumption at 100% duty cycle was 3.14 mW. The chip core area was 1.97 mm × 1.97 mm with a total on-chip memory (SRAM) of 262 kB. It was found that a clock frequency of at least 10 MHz was required to process 16 kHz waveforms in real-time. At this clock frequency, 1.91 mW of power was required.

F. COMPARISON WITH OTHER SPEECH PROCESSORS

Table 4 compares the proposed SE design with state-of-the-art speech processors, many of which exerted much effort in hardware optimisation. The comparison includes SE and speech recognition processors implemented with various artificial neural networks (i.e., CNN, feedforward DNN and

TABLE 4. Performance comparison with state-of-the-art deep learning-based speech processors.

Reference	[1]	[5]	[9]	[28]	[29]	[30]	This work
Application	Speech recognition	Speech enhancement	Speech enhancement	Speech enhancement	Speech recognition	Speech recognition	Speech enhancement
Supported Classifier	DNN	CNN	-	LSTM RNN	CNN	CNN	DNN
Technology (nm)	65	40	65	65	180	28	65
Core V_{DD} (V)	0.5 – 1.2	0.6 – 0.9	1.2	0.68 – 1.1	1.2	0.57 – 0.9	0.65 – 1.2
Core Area (mm^2)	9.61	4.2	1.36	7.74	16.96	1.29	3.88
Precision (Bit)	-	16	-	6/13	-	-	2
Frequency (MHz)	10.2 – 86.8	5 – 20	100	8 - 80	16	2.5 - 50	10 – 75
On-chip SRAM size (Kb)	-	327	35	297	51.8	52	262
Energy Efficiency (TOPS/W)	-	1.20 – 2.18	-	2.45 – 8.93	-	90	1.28 – 2.02

LSTM RNN). The SE processor in this work has comparable performance with the CNN-FFT-based speech enhancement processor in [5], where hardware and parameter sharing, zero-skipping, low-rank expansion, weight quantisation and careful arrangement of processing elements were leveraged. Despite integrated circuit implementation and optimisation being minimally explored, the proposed DNN produced a smaller core area than [1], [5], [28] and [29], smaller on-chip memory requirement than [5] and higher energy efficiency in terms of tera-operations per second per Watt (TOPS/W) than [28] and [30].

Future work will include exploring more hardware optimisation techniques, for example, resource sharing and mapping the SE design onto a smaller technology node to further reduce its chip core size. In addition, the conventional cochleagram-based mapping approach could be refined to enhance its sophistication, thereby augmenting its denoising capabilities. The potential benefits of CNNs for multi-target learning in the context of SE also present a worthwhile investigation.

VI. CONCLUSION

This paper discusses an efficient supervised SE design. Using multi-target learning improved STOI and PESQ scores when processing noisy speech contaminated with noise types from the NOISEX-92 database. Allowing the SE processor to switch between mapping-based, masking-based, or joint processing based on the sensed level of noise contamination on speech provided better denoising performance than when only a single type of processing was made available. Applying dynamic ternary quantisation reduced the neural weights to a 2-bit representation from 32-bit, providing 16x compression. Structured pruning was implemented with weight penalty clipping to encourage the formation of more meaningful group sparsity. This resulted in further reduction of neural weights; from 2-bit to 1-bit. Combining ternary quantisation with structured pruning led to ∼19.1x total compression from a fully connected full-precision DNN. SE processor implementation leveraged the flexibility of FPGA. Estimates for power and area were obtained for the TSMC 65 nm CMOS technology. The computational and memory requirements of the processor were substantially reduced, achieving 1.28-2.02 TOPS/W and a core area of 3.88 mm^2. This is comparable to state-of-the-art speech processors focused on hardware implementations and optimisations.

REFERENCES

[1] M. Price, J. Glass, and A. P. Chandrakasan, "A low-power speech recognizer and voice activity detector using deep neural networks," *IEEE J. Solid-State Circuits*, vol. 53, no. 1, pp. 66–75, Jan. 2018.
[2] L. Sun, J. Du, L. Dai, and C. Lee, "Multiple-target deep learning for LSTM-RNN based speech enhancement," in *Proc. Hands-Free Speech Commun. Microphone Arrays (HSCMA)*, San Francisco, CA, USA, 2017, pp. 136–140.
[3] H. Phan et al., "Improving GANs for speech enhancement," *IEEE Signal Process. Lett.*, vol. 27, pp. 1700–1704, 2020.
[4] T. Kounovsky and J. Malek, "Single channel speech enhancement using convolutional neural network," in *Proc. IEEE Int. Workshop Electron., Control, Meas., Signals Appl. Mechatron. (ECMSM)*, Donostia, Spain, 2017, pp. 1–5.
[5] Y. Lee, T. Chi, and C. Yang, "A 2.17mW acoustic DSP processor with CNN-FFT accelerators for intelligent hearing assisstive devices," *IEEE J. Solid-State Circuits*, vol. 55, no. 8, pp. 2247–2258, Aug. 2020.
[6] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 12, pp. 1849–1858, Dec. 2014.
[7] X. Zhen, M. Yu, X. He, and S. Li, "Multi-target regression via robust low-rank learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 497–504, Feb. 2018.
[8] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," in *Proc. Interspeech*, Lyan, France, 2013, pp. 1508–1512.
[9] J. Lee, S. Park, I. Hong, and H.-J. Yoo, "An 8.3mW 1.6Msamples/s multi-modal event-driven speech enhancement processor for robust speech recognition in smart glasses," in *Proc. 42nd Eur. Solid-State Circuits Conf. (ESSCIRC)*, Lausanne, Switzerland, 2016, pp. 117–120.
[10] Y.-J. Lin, Y.-C. Lee, H.-M. Liu, H. Chiueh, T.-S. Chi, and C.-H. Yang, "A 1.5mW programmable acoustic signal processor for hearing assistive devices with speech intelligibility enhancement," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 12, pp. 4984–4993, Dec. 2020.
[11] S. R. Chiluveru, Gyanendra, S. Chunarkar, M. Tripathy, and B. K. Kaushio, "Efficient hardware implementation of DNN-based speech enhancement algorithm with precise sigmoid activation function," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 11, pp. 3461–3465, Nov. 2021.
[12] Y. Wang, K. Han, and D. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 2, pp. 270–279, Feb. 2013.
[13] P. Cosi and E. Zovato, "Lyon's auditory model inversion: A tool for sound separation and speech enhancement," in *Proc. Workshop Auditory Basis Speech Percept.*, 1999, pp. 194–197.
[14] S. Abdullah, M. Zamani, and A. Demosthenous, "Towards more efficient DNN-based speech enhancement using quantized correlation mask," *IEEE Access*, vol. 9, pp. 24350–24362, 2021.

[15] B. W. Silverman, *Density Estimation*. London, U.K.: Chapman and Hall, 1986.

[16] E. Park, J. Ahn, and S. Yoo, "Weighted-entropy-based quantization for deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 5456–5464.

[17] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, " Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1," 2016, *arXiv:1602.02830*.

[18] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Proc. 29th Conf. Neural Inf. Process. Syst. (NIPS)*, Barcelona, Spain, 2016, pp. 2074–2082.

[19] L. Yang, Z. He, and D. Fan, "Harmonious coexistence of structured weight pruning and ternarization for deep neural networks," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, New York, NY, USA, vol. 34, no. 4, pp. 6623–6630, Apr. 2020.

[20] B. Yang, D. Wang, and L. Liu, "Complex division and square-root using CORDIC," in *Proc. 2nd Int. Conf. Consumer Electron., Commun. Netw. (CECNet)*, Yichang, China, 2012, pp. 2464–2468.

[21] A. Saeed, M. Elbably, G. Abdelfadeel, and M. I. Eladawy, "Efficient FPGA implementation of FFT/IFFT processor," *Int. J. Circuits, Syst. Signal Process.*, vol. 3, no. 3, pp. 103–110, Jan. 2009

[22] A. Rojo-Hernandez, G. Sanchez-Rivera, G. Avalos-Ochoa, H. Perez-Meana, and L. Smith, "A compact digital gamma-tone filter processor," *Microprocess. Microsyst.*, vol. 45, pp. 216–225, Aug. 2016.

[23] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep. 2011.

[24] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, USA, 2001, pp. 749–752.

[25] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallet, and N. L. Dahlgren, "DARPA-TIMIT: Acoustic-phonetic continuous speech corpus," U.S. Dept. Commer., Washington, DC, USA, document NISTIR 4930, 1993.

[26] A. Varga and H. J. M. Steeneken, "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.*, vol. 12, no. 3, pp. 247–251, Jul. 1993.

[27] S. Xia, H. Li, and X. Zhang, "Using optimal ratio mask as training target for supervised speech separation," in *Proc. Asia-Pac. Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Kuala Lumpur, Malaysia, 2017, pp. 163–166.

[28] D. Kadetotad, S. Yin, V. Berisha, C. Chakrabarti, and J. Seo, "An 8.93 TOPS/W LSTM recurrent neural network accelerator featuring hierarchical coarse-grain sparsity for on-device speech recognition," *IEEE J. Solid-State Circuits*, vol. 55, no. 7, pp. 1877–1887, Jul. 2020.

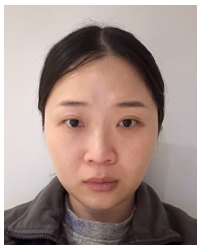
[29] L. Wu, Z. Wang, M. Zhao, W. Hu, Y. Cai, and R. Huang, "A high ac curacy multiple-command speech recognition ASIC based on configurable one-dimension convolutional neural network," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Daegu, South Korea, 2021, pp. 1–4.

[30] S. Zheng et al., "An ultra-low power binarized convolutional neural network-based speech recognition processor with on-chip self-learning," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 12, pp. 4648–4661, Dec. 2019.

[31] P. Papadopoulos, A. Tsiartas, and S. Narayanan, "Long-term SNR estimation of speech signals in known and unknown channel conditions," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 12, pp. 2495–2506, Dec. 2016.

[32] J. Lee, Y. Jung, M. Jung, and H. Kim, "Dynamic noise embedding: Noise aware training and adaptation for speech enhancement," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Auckland, New Zealand, 2020, pp. 1–8.

[33] X. Hao, X. Su, Z. Wang, H. Zhang, and Batushiren, "UNetGAN: A robust speech enhancement approach in time domain for extremely low signal-to-noise ratio condition," in *Proc. Interspeech*, 2019, pp. 1786–1790, doi: 10.21437/interspeech.2019-1567.



SALINNA ABDULLAH (Graduate Student, IEEE) received the M.Eng. degree in electronic engineering with computer science and the Ph.D. degree in electronic and electrical engineering from University College London (UCL), London, U.K., in 2017 and 2023, respectively. She was a recipient of an EPSRC Industrial Strategy Studentship and conducted her Ph.D. work with the Bioelectronics Group, UCL Department of Electrical and Electronic Engineering. Her research interests include FPGA design, and efficient application of speech enhancement, image processing and deep-learning methods in wearable devices. She was awarded the Cisco Prize for Most Outstanding Female Engineer during her final year of undergraduate study.

MAJID ZAMANI (Member, IEEE) received the M.Sc. degree in microelectronics from Islamic Azad University, Science, and Research Branch, Tehran, in 2011, and the Ph.D. degree with outstanding contribution in implantable brain decoding from University College London (UCL), London, U.K., in 2017, where he was a Postdoctoral Research Fellow with the Analog and Biomedical Electronics Group from 2017 to 2022. He is currently a Lecturer (Assistant Professor) with the School of Electronics and Computer Science, University of Southampton, U.K. His research interests include design and fabrication of advanced and energy-efficient intelligent systems, especially for implantable brain sensing, processing, brain-machine interfacing, and biomedical applications. He was a recipient of the Oversea Research Scholarship and the UCL Graduate Research Scholarship to pursue the Ph.D. degree. He was also a recipient of the Best Researcher M.Sc. Student Award.

ANDREAS DEMOSTHENOUS (Fellow, IEEE) received the B.Eng. degree in electrical and electronic engineering from the University of Leicester, Leicester, U.K., in 1992, the M.Sc. degree in telecommunications technology from Aston University, Birmingham, U.K., in 1994, and the Ph.D. degree in electronic and electrical engineering from University College London (UCL), London, U.K., in 1998.

He is currently a Professor with the Department of Electronic and Electrical Engineering, UCL, where he leads Bioelectronics Group. He has made outstanding contributions to improving safety and performance in integrated circuit design for active medical devices, such as spinal cord and brain stimulators. He has numerous collaborations for cross-disciplinary research, both within the U.K. and internationally. He has authored over 350 articles in journals and international conference proceedings, several book chapters, and holds several patents. His research interests include analog and mixed-signal integrated circuits for biomedical, sensor, and signal processing applications. He was a co-recipient of a number of best paper awards and has graduated many Ph.D. students. He was an Associate Editor from 2006 to 2007 and the Deputy Editor-in-Chief from 2014 to 2015 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS; and an Associate Editor from 2008 to 2009 and the Editor-in-Chief from 2016 to 2019 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS. He was an Associate Editor of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS from 2013 to 2023, and currently serves on its steering committee. He serves on the editorial board of Physiological Measurement. He has served on the Technical Programme Committee of numerous conferences including ISCAS, BIOCAS, ICECS, ESSCIRC, and NER. He was the Chair of the IEEE Circuits and Systems Society (CASS) Fellows Evaluation Committee from 2022 to 2023, and has served on many CASS committees, including the Board of Editors, the John Choma Education Award Evaluation Committee, and the Mac Van Valkenburg Award Evaluation Committee. He is the Chair of the U.K. and Ireland IEEE CASS Chapter and the General Co-Chair of the 2025 IEEE International Symposium on Circuits and Systems. He is a Fellow of the Institution of Engineering and Technology and of the European Alliance for Medical and Biological Engineering Sciences, and a Chartered Engineer.