Deep Latent Fusion Layers for Binaural Speech Enhancement

Tom Gajecki [1] and Waldo Nogueira [2]

[1]Hannover medical school
[2]Affiliation not available

October 30, 2023

Abstract

In this work, we address the problem of speech enhancement in the context of binaural hearing. We propose deep learning models which are connected by "fusion layers" that perform Hadamard products between specific generated latent representations. Fusion layers are inspired by multi-task learning approaches that combine and/or share weights between models that tackle related tasks. We first present a general fusion model and show that this approach is able to fit synthetic data better than independent linear models, equalize activation variance between learning modules, and exploit input data redundancy to improve the training error. We then apply the concept of fusion layers to enhance speech in binaural listening conditions. Our results show that the proposed approach improves speech enhancement performance on unseen data with respect to the independent models. However, we observe a trade-off between speech enhancement performance and predicted speech intelligibility based on a short-time objective binaural speech intelligibility index, potentially due to distortions introduced by fully fused models.

Results also suggest that fusion layers should share parameterized latent representations in order to properly exploit the information contained in each listening side. In general, this work shows that sharing information between speech enhancement modules may be promising to improve binaural speech enhancement while keeping the number of trainable parameters constant and improving generalization.

Deep Latent Fusion Layers for Binaural Speech Enhancement

Tom Gajecki & Waldo Nogueira

***Abstract*—In this work, we address the problem of speech enhancement in the context of binaural hearing. We propose deep learning models which are connected by "fusion layers" that perform Hadamard products between specific generated latent representations. Fusion layers are inspired by multi-task learning approaches that combine and/or share weights between models that tackle related tasks. We first present a general fusion model and show that this approach is able to fit synthetic data better than independent linear models, equalize activation variance between learning modules, and exploit input data redundancy to improve the training error. We then apply the concept of fusion layers to enhance speech in binaural listening conditions. Our results show that the proposed approach improves speech enhancement performance on unseen data with respect to the independent models. However, we observe a trade-off between speech enhancement performance and predicted speech intelligibility based on a short-time objective binaural speech intelligibility index, potentially due to distortions introduced by fully fused models. Results also suggest that fusion layers should share parameterized latent representations in order to properly exploit the information contained in each listening side. In general, this work shows that sharing information between speech enhancement modules may be promising to improve binaural speech enhancement while keeping the number of trainable parameters constant and improving generalization.**

***Index Terms*—Fusion layers, Binaural speech enhancement, Deep learning, Latent representations**

I. Introduction

Deep learning technology has been successfully applied to perform speech enhancement, i.e., removing or attenuating interfering noise from a speech signal. Recently, binaural speech enhancement methods [1], [2] that share information between listening sides have been developed to exploit redundant information to further improve noise reduction. Here, we address the problem of speech enhancement in binaural listening by introducing a simple weight-sharing mechanism between two monaural speech enhancement algorithms.

Commonly, deep learning models are trained to perform one task at a time. For example, in image processing, a deep neural network (DNN) can be trained to classify images between a set of classes or to segment particular objects of interest within images (e.g., [3]–[5]). In the context of speech processing, DNNs can be trained to recognize the words in speech sentences from the raw audio (e.g., [6]–[8]), or to automatically remove the unwanted components of a corrupted speech signal, such as noise or other speakers (e.g., [9]–[12]). These approaches work generally well, but they may ignore potential rich sources of information contained in real-world problems. For instance, speech enhancement systems improve noise reduction performance when also relying on visual feedback, giving rise to audio-visual speech enhancement [13]. Here is where multi-task learning (MTL) comes into play.

MTL is a subset of deep learning techniques in which multiple learning tasks are solved at the same time while exploiting similarities and differences between them. This technique is generally the result of sharing parameters between different models [14]–[16]. MTL can provide the models with higher generalization capabilities by leveraging the domain-specific information contained in the training signals of related tasks. It does this by training tasks in parallel while sharing latent representations of the input data. This method can be used, for example, to identify an object within an image, recognize the overall scene and generate a verbal caption for it (e.g., [17], [18]). Also, for speech processing, MTL can be used to improve speech activity detection (e.g., [19], [20]).

Much of the current deep learning research has focused on coming up with better architectures, and it is not different for MTL. Actually, architecture plays possibly even a larger role in MTL because of the number of possibilities that one has to tie multiple tasks together. In other words, the way the parameter sharing between the networks is performed is not obvious. In fact, there is research devoted to finding optimal latent multi-task architectures [21], [22]. However, simple approaches such as cross-stitch networks that learn linear combinations of latent representations between the models have proven to be successful in generalizing into multiple tasks [23], [24]. In this work, we present a simple weight-sharing method to perform binaural speech enhancement.

A healthy human auditory system is excellent at isolating target signals in acoustically challenging conditions, this is due to the ability it has to exploit both acoustic inputs captured by each of the ears, and to centrally compare features contained in them; this is known as binaural hearing [25], [26]. The problem of binaural speech enhancement has been an active research problem for already some time (e.g., [27]–[29]). However, more recently, DNNs have proven to be successful at performing speech separation in binaural listening by sharing acoustic binaural features. For example, previous research has used feature concatenation and self-attentive mechanisms to perform binaural speech enhancement (e.g., [2], [30], [31]). These methods rely on explicit feature extraction and are not necessarily motivated by the human binaural auditory system.

TG and WN are with the department of Otolaryngology, Medical University of Hanover and Cluster of Excellence, Hearing4all, Hannover, 30625, Germany.

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) Project ID 446611346. Thanks to all the CI participants that took part in this study.

*Correspondence email: gajecki.tomas@mh-hannover.de

Although the exact fundamental physiological mechanisms by which the binaural hearing system exploits different acoustic cues are not fully understood [32], [33], there have been attempts to develop computational models that explain empirically observed human binaural hearing abilities, such as the equalization-cancellation model [34], [35]. This model suggests explaining binaural masking level differences with processes of relative delay compensation and then subtraction of particular acoustic features captured by each ear to attenuate the interfering noise. In this work, we propose DNNs that although do not perform the same operations as the equalization model, may learn to combine latent features to emulate neural excitation and inhibition processes that happen in the brain stem for binaural acoustic processing [33].

Inspired by MTL weight sharing and the binaural equalization-cancellation model, we investigate the influence that sharing the latent representation of two single-channel end-to-end speech enhancement DNNs has on the speech enhancement performance of binaural noisy speech signals. We will share the latent representations through fusion layers that apply element-wise dot product operations to each of the features contained in them. The fusion layers are designed to introduce non-linearities to the speech enhancement model that will allow fitting better the training data while improving generalization without affecting the number of trainable parameters. We expect that the fused models will emphasize latent target feature representations in the fused layers by canceling unwanted noisy elements contained in the input audio signal, causing also a decrease in layer activation variance. This work extends a previous study[1] presented at the 2021 Clarity speech enhancement challenge [36] by formalizing the concept and by analyzing the effect of input data correlation, latent activation variance, and encoding methods.

The rest of this manuscript is organized as follows. Section II describes the method. The experimental results are presented in Section III, and Section IV concludes this manuscript.

II. Methodology

A. General fused model

The main aspect we aim at investigating in this study is the effect that sharing information between deep learning models has on data fitting and generalization performance. We propose to share this information by means of fusion layers that apply dot-product operations to specific latent representations at different stages of data processing. We will first describe a general fused model to formalize the notation that will be used throughout the manuscript.

Let $\mathbf{Y}_m = \Omega_m(\mathbf{X}_m) \in \mathbb{R}^{D_L}$, where D_L is the dimensionality of the output tensors, be the output tensors computed by a set of learning models given by $\Omega_m(\cdot)$, for a given set of input tensors $\mathbf{X}_m \in \mathbb{R}^{D_0}$, where D_0 is the dimensionality of the input tensors, $\{m = 1, \ldots, M\}$, and M is the number of DNNs. Each of the models contains L learning modules (i.e., layers, multi-layer perceptrons, etc...) that apply a function $\omega_{l,m}(\cdot)$ to transform its input tensor into a latent representation of it, i.e., $\mathbf{X}_{l,m} = \omega_{l,m}(\mathbf{X}_{l-1,m})$, $\{l = 1, \ldots, L-1\}$ (note that for the input and output tensors the index l is omitted). At this point, we introduce the fusion layer. This layer is designed to share information between the different models by means of an element-wise dot product of the latent representations at different stages of the processing. Let $\rho(\cdot)$ be the Hadamard product operator. The output of the fusion layers will be represented by tensors $\boldsymbol{\chi}_{l,m} = \rho(\mathbf{X}_{l,m}, \Lambda_{l,m})$, where $\mathbf{X}_{l,m}$ is the output of the learning module (l, m), and $\Lambda_{l,m}$ is the set of tensors that will be fused at layer (l, m) with $\mathbf{X}_{l,m}$, such that $\Lambda_{l,m} := \{\mathbf{X}_{l,m'} | m' \neq m \wedge 1 \leq m' \leq M\}$. Here, the direct path without fusion is indicated by $\Lambda_{l,m} = \{J_l\} \in \mathbb{R}^{D_l}$ (all-ones tensor), with D_l being the output dimensionality of layer l. In this case $\boldsymbol{\chi}_{l,m} = \mathbf{X}_{l,m}$.

A general deep fusion model is shown in Figure 1. In this graph, learning modules and fusion layers are indicated by black and white vertices, respectively, whereas the flow of tensors is indicated by directed edges. This model can be simply described with matrix notation through the deep latent fusion matrix Δ for each fusion set $\Lambda_{l,m} \in \mathbb{R}^{D_l}$, as follows:

$$\Delta = \begin{pmatrix} \Lambda_{1,1} & \Lambda_{1,2} & \cdots & \cdots & \cdots & \Lambda_{1,M} \\ \Lambda_{2,1} & \Lambda_{2,2} & \cdots & \cdots & \cdots & \Lambda_{2,M} \\ \vdots & \vdots & \ddots & & & \vdots \\ \Lambda_{l,1} & \Lambda_{l,2} & & \Lambda_{l,m} & & \Lambda_{l,M} \\ \vdots & \vdots & & & \ddots & \vdots \\ \Lambda_{L-1,1} & \Lambda_{L-1,2} & \cdots & \cdots & \cdots & \Lambda_{L-1,M} \end{pmatrix}. \quad (1)$$

The here presented fusion layers have three purposes, namely: 1) Introduce non-linearities to the model in a controlled way; 2) Leverage input feature redundancy (i.e., correlations) to improve data fitting 2), and; Act as a channel for the gradients to back-propagate through, to reduce the activation variance between learning modules and improving generalization on unseen data [37].

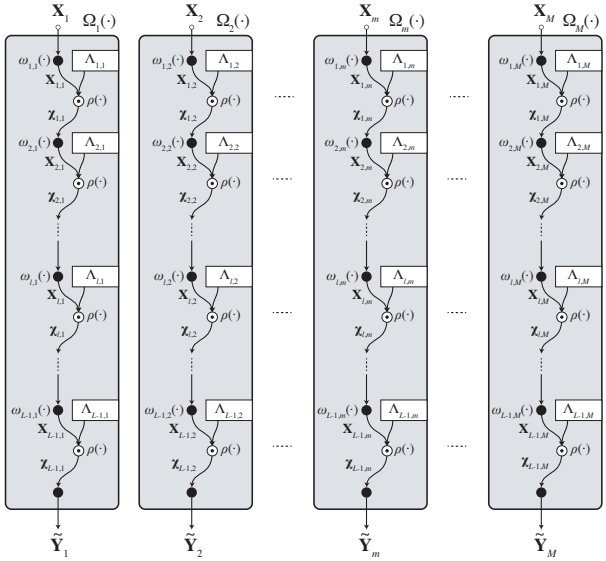

Fig. 1. Graph diagram of a general fused model. Learning modules are indicated by black-filled vertices and fusion layers by white vertices, whereas the flow of tensors is indicated by directed edges.

[1] https://github.com/tomgajecki/FusionLayers/blob/main/Clarity_2021_gajecki.pdf

B. Fully fused linear models

To investigate the effects that the fusion layers have on a specific model we will simplify the generic fused model by assuming that all learning modules (i.e., fully connected layers) are linear, and that input tensors are vectors $\mathbf{X}_m \in \mathbb{R}^{1\times T}$. This will allow us to assess how non-linearities are introduced due to the interconnection of the independent models, characterize how the input data correlation affects the data fitting, and assess how the variance of the layer activations is impacted. The general model shown in Figure 1 that does not contain any fusion layers will be be referred to as "independent" (i.e., $\Lambda_{l,m} = \{J_l\}\ \forall\ l, m$), where $J_l \in D^l$ is the all-ones tensor. Each of the models contains L layers (i.e., the learning modules) consisting of n_l parameters. Activation functions for each of the layers are defined by $\phi_{l,m}(\cdot),\ \forall\ l, m$. The output at layer l for model m is given by $\mathbf{X}_{l,m} = \omega(\mathbf{X}_{l-1,m}; \mathbf{w}_{l,m}, b_{l,m}) = \phi_{l,m}(\mathbf{X}_{l-1,m}^{\top}\mathbf{w}_{l,m} + b_{l,m})$, where $\mathbf{w} \in \mathbb{R}^{(n_l-1)\times n_l}$ and $b_{l,m} \in \mathbb{R}^{1\times n_l}$ are the weights and biases, respectively. Assuming that all activations are linear, the output of each layer and model $\mathbf{X}_{l,m}$ will satisfy $\partial \mathbf{Y}_m(\mathbf{X}_{l,m})/\partial \mathbf{X}_{l-1,m} = C_{l,m} \in \mathbb{R}$; i.e., a constant. Hence, every model m will be reduced to a linear regression.

1) Generating non-linear models through fusion: Now lets define a fused model where all layers are multiplied with each other for all learning modules, that is $\Lambda_{l,m} := \{\mathbf{X}_{l,m'}\ \forall\ l \wedge m' \neq m\}$. We will introduce two fusion modalities, namely: side-wise fusion and depth-wise fusion. These two ways of making the models interact with each other will have different effects on the non-linearities introduced and on how latent information is transmitted throughout the models. These will be described in the following lines.

Side-wise fusion level is defined at a given layer (l, m) as the number of fusion layers, that is, $|\Lambda_{l,m}|$, where $|A|$ represents the cardinality of a set A. In general, the fusion output at layer l with a side-wise fusion level of $|\Lambda_{l,m}| = M - 1$ is given by:

$$\boldsymbol{\chi}_{l,m} = \prod_{m=1}^{M} \omega_{l,m}(\mathbf{X}_{l-1,m}; \mathbf{w}_{l,m}, b_{l,m}). \tag{2}$$

This fusion operator (i.e., chained Hadamard products) will cause the M models to no longer be independent, introducing non-linearities at the output of a given learning module l such that the leading order term (LOT) is:

$$\text{LOT}\Big(\frac{\partial \boldsymbol{\chi}_{l,m}}{\partial \mathbf{X}_{l-1,m}}\Big) \sim \mathcal{O}(n^{M-1}). \tag{3}$$

Depth-wise fusion level occurs for models with multiple learning modules (i.e., deep multi-layer models), that include deeper processing stages to increase the order of the modeled function. If we consider a fully fused linear model, the fusion output of layer l can be written as equation 2. At layer $L-1$ the output of the fusion layer will not only depend on the side-wise fusion operation but also on the previous latent representations. This output can be written as a function of previous fusion operations as follows:

$$\boldsymbol{\chi}_{L-1,m} = \prod_{l>1}^{L-1} \prod_{m=1}^{M} \omega_{l,m}(\mathbf{X}_{l-1,m}; \mathbf{w}_{l,m}, b_{l,m}), \tag{4}$$

where L is the number of learning modules that each model contains. In this case the introduced non-linearities at the output of a given learning module l such that the LOT is:

$$\text{LOT}\Big(\frac{\partial \boldsymbol{\chi}_{L-1,m}}{\partial \mathbf{X}_{L-2,m}}\Big) \sim \mathcal{O}\big(n^{(L-1)\cdot M-1}\big). \tag{5}$$

C. Experiment 1: Fusion for artificial data fitting

In this experiment, we aim at investigating the effects of the fusion operation on simple regression problems on an artificially generated dataset. We divide this experiment into two sub-experiments; one will show empirically that non-linearities are introduced by the operation shown in equation 2, and in the second one we investigate the trade-off between the correlation of the input data at each model and its fitting capabilities.

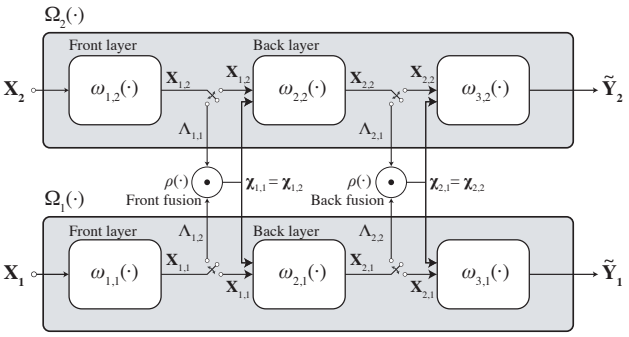

Fig. 2. Block diagram of a model comprised of two deep learning sub-models, each containing three learning modules. Fusion layers are included or bypassed using the switches depicted in the block diagram. Each sub-model is represented by the grey blocks and each of the learning modules is represented by the white blocks. This model has a side-wise fusion level of one and a depth-wise fusion level of two.

Model: In this experiment we will keep the number of sub-models $m = 2$. The input and output layers of all sub-models consist of one single unit and the number of units in each of the hidden layers will be specified by "layer size", for which we tested n_l ={32, 64, 128, 256}.

Dataset: The dataset for this experiment was artificially generated by creating input vectors with elements sampled from random uniform distributions. Because we keep the number of models $m = 2$, two input vectors were created, $\mathbf{X}_1 \in \mathcal{U}\{0, 1\}$ and $\mathbf{X}_2 \in \mathcal{U}\{0, 1\}$ containing 500 samples each (see Figure 5, first panel). From the input data we generated a non-linear output for each sub-model ($\mathbf{Y}_1$ for sub-model 1 and $\mathbf{Y}_2$ for sub-model 2) as follows:

$$\begin{cases} \mathbf{Y}_1 = 0.5\cdot sin(10\cdot \mathbf{X}_1) + \mathbf{X}_{n1} + 0.4 \\ \mathbf{Y}_2 = 0.5\cdot sin(10\cdot (\mathbf{X}_1\cdot(1-d) + d\cdot \mathbf{X}_2) + 2) + \mathbf{X}_{n2} + 0.9 \end{cases}, \tag{6}$$

where $\mathbf{X}_{n1}$ and $\mathbf{X}_{n2}$ are noisy samples with a maximum amplitude of 0.3, and d is a multiplicative factor that controls the amount of correlation at the input ($d = 0$ for fully correlated inputs and $d = 1$ for fully uncorrelated inputs).

Loss function: To fit the artificial training data to the target functions described in equation 6, we minimized the mean-

squared-error (MSE) between the predicted output $\mathbf{Y}$ and the target $\hat{\mathbf{Y}}$. The MSE computed over n samples is defined as:

$$\text{MSE}(\mathbf{X}, \tilde{\mathbf{Y}}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{Y}_i - \tilde{\mathbf{Y}}_i)^2. \quad (7)$$

Training: The models were trained for a maximum of 100 epochs in batches of 10 samples. The initial learning rate was set to 1e-3. The learning rate was halved if the accuracy of the validation set did not improve during 3 consecutive epochs, early stopping with a patience of 5 epochs was applied as a regularization method, and only the best performing model was saved. For the model optimization, Adam [38] was used to minimize the MSE (see equation 7) between the estimated and true outputs.

1) Experiment 1.1: In this experiment, we aim at investigating the effect that the non-linearities introduced by the fusion mechanisms have on the training error. We do this by comparing the output errors obtained by the linear independent and fused models. Also, for this experiment the input vector fed to each sub-model will be identical ($\mathbf{X}_1 = \mathbf{X}_2$). This experiment may reveal if one can profit by adding non-linearities in a controlled way through fusion when compared to using a completely linear model.

2) Experiment 1.2: In this experiment, we aim at investigating how susceptible is the proposed attention mechanism to differences between inputs in each sub-model. We will aim at answering this question by computing the errors at the output of each independent and fused model as a function of the correlation of the input data. This aspect is important because of the motivation behind using fusion layers in binaural speech enhancement systems, where the correlation between hearing sides is present. However, we aim at assessing a potential threshold below which fusion is no longer that beneficial to fit the training data distribution.

3) Experiment 1.3: In this experiment, we empirically measure the activation variances across predictions of the fused layers (See $\Lambda_{i,j}\ \forall\ \{i,j\} = \{1,2\}$ in Figure 2) as well as their counter independent layers. The variance of the activations contained in each layer is defined as:

$$Var[\text{activations}] = \text{E}[(\mathbf{w}_{l,m} - \bar{\mathbf{w}}_{l,m})^2], \quad (8)$$

where $\text{E}[\cdot]$ is the expected value operator, $\mathbf{w}_{l,m}$ is the tensor containing all of the learned wights in layer l in model m, and $\bar{\mathbf{w}}_{l,m}$ is the average activation value in layer l and model m.

To assess how variance changes across models, we train an independent and all possible fused models (from Figure 2 using only $\Lambda_{1,1}$ and $\Lambda_{1,2}$, only $\Lambda_{2,1}$ and $\Lambda_{2,2}$, both pair of sets, or none of them) 50 times using different random initialization seeds. This will give an idea of how the activation variance is affected by the fusion operation. Also, we measure the variance including correlated and uncorrelated input data to remove possible training bias.

D. Experiment 2: Fusion layers for binaural speech signals

In this experiment, we investigate the effect that fusion layers have on noise reduction performance in the context of end-to-end speech enhancement.

Model: The speech enhancement algorithm relies on two end-to-end audio speech enhancement models; each consisting of three processing stages, as shown in Figure 3: an encoder, a separator (a temporal convolution module, and a mask estimator), and a decoder. The encoder extracts features from the input audio signal that are then passed into the separator that estimates a mask to remove noisy elements of the input audio, and the de-noised audio is resynthesized by the decoder. The implementation was done in TensorFlow 2.0 [39] and the code for training and evaluating can be found online[2].

Dataset: The speech material used for the evaluation of the speech enhancement models was obtained from the TIMIT acoustic-phonetic Continuous Speech Corpus [40] (consisting of a set dedicated for training and another set for testing). The interfering noisy signals were all obtained from the DEMAND collection of multi-channel recordings of acoustic noise in diverse environments [41]. The training set was obtained by mixing all of the training data contained in the TIMIT speech dataset with 50% of the DEMAND noise signals. The validation dataset, used to monitor the models' training process, consisted of 20% of the training material. The testing set was obtained by mixing the remaining 50% of the DEMAND noise signals with the TIMIT speech testing set.

Each acoustic scene corresponded to a unique target utterance and a unique segment of noise from an interferer, mixed at signal-to-noise ratios (SNRs) ranging from -6 to 6 dB. The three sets were balanced for the target speaker's gender. Binaural room impulse responses (BRIRs) [42] were used to model a listener in a realistic acoustic environment. The BRIR recording data set[3] consisted of 4 different rooms of different sizes and acoustic properties. The audio signals for the scenes were generated by convolving source signals with the BRIRs and summing.

Tested topologies: To investigate how the fusion operation affected the models' performance, we tested four configurations described in Table I.

TABLE I
SPEECH ENHANCEMENT ALGORITHMS AND THEIR CORRESPONDING FUSION MATRIX.

Topology	Fusion matrix
Independent	$\Delta_I = \begin{pmatrix} \{J\} & \{J\} \\ \{J\} & \{J\} \end{pmatrix}$
Front fusion	$\Delta_F = \begin{pmatrix} \{\mathbf{X}_{f,r}\} & \{\mathbf{X}_{f,l}\} \\ \{J\} & \{J\} \end{pmatrix}$
Back fusion	$\Delta_B = \begin{pmatrix} \{J\} & \{J\} \\ \{\mathbf{X}_{b,r}\} & \{\mathbf{X}_{b,l}\} \end{pmatrix}$
Double fusion	$\Delta_D = \begin{pmatrix} \{\mathbf{X}_{f,r}\} & \{\mathbf{X}_{f,l}\} \\ \{\mathbf{X}_{b,r}\} & \{\mathbf{X}_{b,l}\} \end{pmatrix}$

To expand our intuition about the effect that fusion layers have on speech enhancement performance, two different encoder/decoder module pairs (i.e., encodings) and two different cost functions were investigated.

Tested encodings: We investigate how the fusion operation affects the models' performance for different encodings of the input signals. Specifically, we test a non-deterministic learned representation and a deterministic representation. This

[2]https://github.com/tomgajecki/FusionLayers
[3]https://github.com/IoSR-Surrey/RealRoomBRIRs

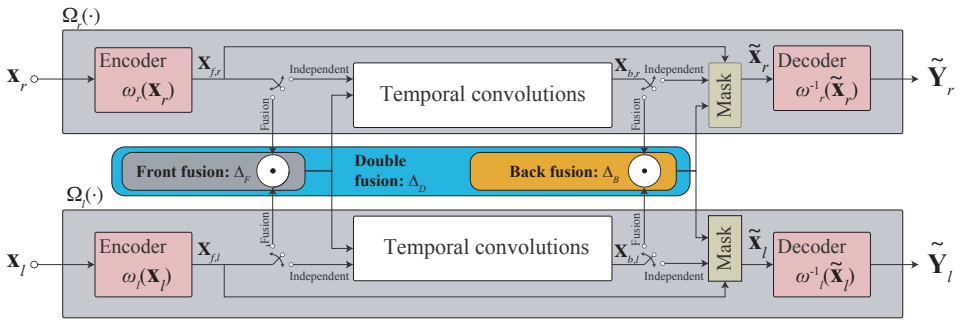

Fig. 3. Block diagram of the evaluated algorithms. "Independent" model bypasses both fusion layers. "Front fusion (Δ_F)" model, "Back fusion (Δ_B)" model, and "Double fusion (Δ_D)" model.

analysis targets the question of whether these layers do indeed leverage redundant binaural information by sharing latent representations between models by introducing non-linearities that adapt to the presented data.

The input mixture sound can be divided into overlapping segments of length R, represented by $\mathrm{X}_k \in \mathbb{R}^{1\times R}$, where $k = 1,\ldots,\hat{T}$ denotes the segment index and $\hat{T}$ denotes the total number of segments in the input. At the encoding stage, X_k is transformed into an F-dimensional representation, $\boldsymbol{\lambda}_k \in \mathbb{R}^{1\times 1\times F}$. This representation can be obtained through 1-d convolution operations (non-deterministic encoding; *deep encoding*), such as in [9], or with a classic spectro-temporal representation of the signal; i.e., *deterministic encoding* (short-time Fourier transform; STFT), where the encoder and decoder blocks shown in Figure 3 represent the STFT and iSTFT, respectively.

Tested loss functions: To assess whether the effect of the fusion mechanisms is dependent on the loss function used to train the models, we investigated two typical cost functions used in the context of speech enhancement, namely, the SNR and the scale-invariant signal-to-distortion ratio (SI-SDR) [43]. The SNR between a given signal with T samples, $\mathrm{X} \in \mathbb{R}^{1\times T}$ and its estimate $\tilde{\mathrm{Y}} \in \mathbb{R}^{1\times T}$ is defined as:

$$\mathrm{SNR}(\mathrm{X},\tilde{\mathrm{Y}}) = 10\cdot log_{10}\left(\frac{||\mathrm{X}||^2}{||\mathrm{X}-\tilde{\mathrm{Y}}||^2}\right). \tag{9}$$

The SI-SDR between a given signal and its estimate is defined as:

$$\text{SI–SDR}(\mathrm{X},\tilde{\mathrm{Y}}) = 10\cdot log_{10}\left(\frac{||\gamma\cdot\mathrm{X}||^2}{||\gamma\cdot\mathrm{X}-\tilde{\mathrm{Y}}||^2}\right), \gamma = \frac{\tilde{\mathrm{Y}}^\top \mathrm{X}}{||\mathrm{X}||^2}. \tag{10}$$

Training: The models were trained for a maximum of 100 epochs on batches of two 4-s long audio segments. The initial learning rate was set to 1e-3. The learning rate was halved if the accuracy of the validation set did not improve during 3 consecutive epochs, early stopping with a patience of 5 epochs was applied as a regularization method, and only the best performing model was saved. For the model optimization, Adam [38] was used. The models were trained and evaluated using a PC with an Intel(R) Xeon(R) W-2145 CPU @ 3.70GHz, 256 GB of RAM, and an NVIDIA TITAN RTX as the accelerated processing unit.

III. Results

A. Experiment 1

1) Experiment 1.1; Non-linearities introduced by the fusion layers and their effect on data fitting: Figure 4 shows box plots of the MSE improvement given by the fused models with linear activations computed as $\delta\mathrm{MSE} = \mathrm{MSE}_{ind} - \mathrm{MSE}_{\Lambda}$, where MSE_{ind} and MSE_{Λ} represent the MSE produced by the independent and fused model, respectively. δMSE is shown for the front, back, and double fusion.

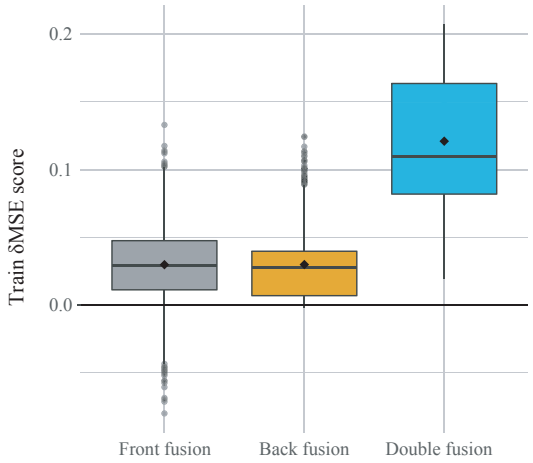

Fig. 4. Box plots showing the increment in MSE error of the the different fused models w.r.t. the independent linear model ($\delta\mathrm{MSE} = \mathrm{MSE}_{ind} - \mathrm{MSE}_{\Lambda}$), for the front, back, and double fusion. The black horizontal bars within each of the boxes represent the median, the diamond-shaped marks indicate the mean improvement, and the top and bottom extremes of the boxes indicate the 75% and 25% quartiles.

An illustrative example of how the fitting of a model of size $n_l = 64$ is affected by the addition of fusion layers is shown in Figure 5. The first panel shows the raw data generated by equation 6. The second panel shows the data fitted by an independent model with $L = 3$ learning modules of size 16 units (see Figure 2). The third panel shows the non-linearity introduced by this model using a side-wise fusion level of 1 and a depth-wise fusion level of 0 (i.e., a polynomial of order 2). Finally, the last panel shows the fitting performed by a fully fused model with a side-wise fusion level of 1 and a depth-wise fusion level of 1; obtaining a polynomial regression of order 4 (see 3 local extrema).

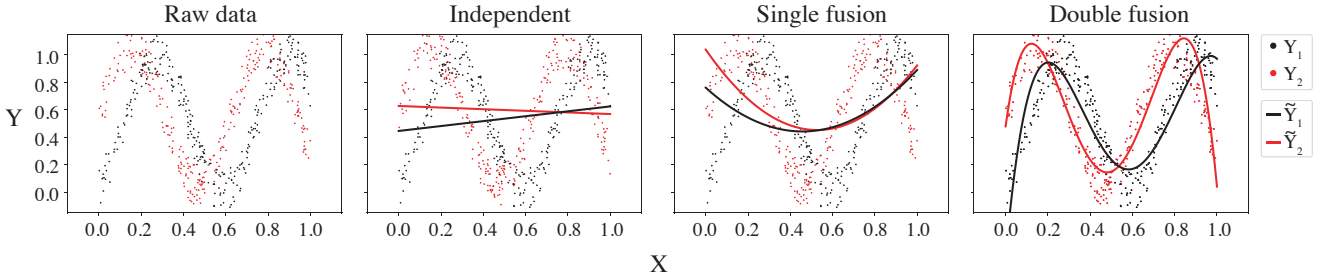

Fig. 5. Data regression plots obtained by the independent and fused models of size 64 on generated synthetic data. Left most plot shows the raw output data Y as a function of the input data X for the left and right channels, and the remaining three plots show the obtained regressions on top of it.

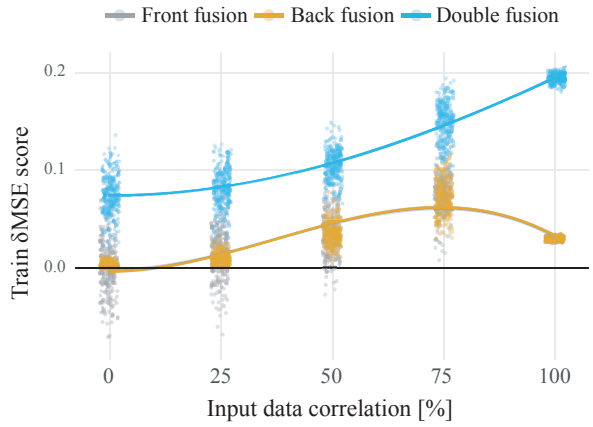

Fig. 6. Dot plot of the training error differences between the independent and fused models (δMSE $=$ MSE$_{ind}$ $-$ MSE$_{\Lambda}$) as a function of input data correlation for generated synthetic data. A second-order polynomial regression is included to show the performance trend for each condition.

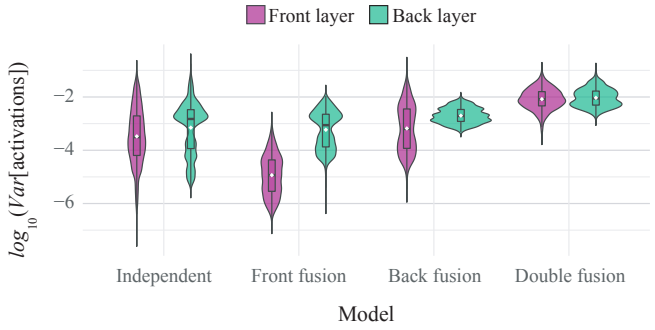

Fig. 7. Violin plots indicating the activation variance across predictions for the front and back fusion layers (see Figure 2) in the different models for generated synthetic data. Data is plotted in a logarithmic scale for visualization purposes. The black horizontal lines within each of the boxes represent the median, the diamond-shaped marks indicate the mean improvement, and the top and bottom extremes of the boxes indicate the 75% and 25% quartiles.

2) Experiment 1.2; The effect of input data correlation on data fitting: Figure 6 shows a dot plot together with its polynomial regression showing how input data correlation affects the training δMSE. It can be seen that for the fully fused model the performance is proportional to the input data correlation whereas for the single fused models the performance reaches its maximum at around 75% correlation. Note that the error of the fully fused models is smaller than the error of the independent models (i.e., δMSE > 0), indicating that the introduced non-linearities do help the model fit the input training data more accurately.

3) Experiment 1.3; Fusion layers and their effect on the models' variance: Figure 7 shows violin plots of the activation variance (in the log_{10} domain) for the front and back fusion layers in the different linear models and fused models. Box plots are also overlapped above the violin plots to show the mean, median, and overall locality of the data.

The violin plot shows, on the one hand, that fusion reduces the range of activation values, especially in the back layers (see in Figure 7 how the violin plots show less deviation from the mean when adding the fusion operation). It can also be seen that variance is not only equalized between sides due to fusion but also between the front and back layers, as depicted by the violin plots corresponding to the double fusion model. It is important to note here that the fact that variance is equalized and balanced through the model is relevant to ensure that all learning modules are learning at the same rate [37].

B. Experiment 2

1) Binaural speech enhancement results: Table II shows the absolute testing and validation results of the speech enhancement algorithm with no fusion layers for the tested loss functions (SNR and SI-SDR), encodings (deep non-deterministic encoding based on 1-D convolutions, and deterministic encoding based on the STFT), N (encoding size; the number of filters in the 1-D convolution or number of STFT bins), and S (number of filters in the latent representation at the output of the temporal convolutions, before the mask estimation module; for more details refer to [9]).

In order to assess the generalization capabilities of the fusion layers, we will be reporting on the test score difference (δ) of the different fused models with respect to the values shown in Table II. Figure 8 shows bar plots of the increment in the validation and testing error (δ test score = Loss$_{ind}$ $-$ Loss$_{\Lambda}$) of the different fused models (see Table I) as a function of fusion size, loss function and encodings. Here it can be seen that fusion seems to improve the performance of the "independent" models only when using deep encoding. In the case of the deterministic STFT encoding, the fusion mechanisms may blur or distort the signal and fail in producing final faithful

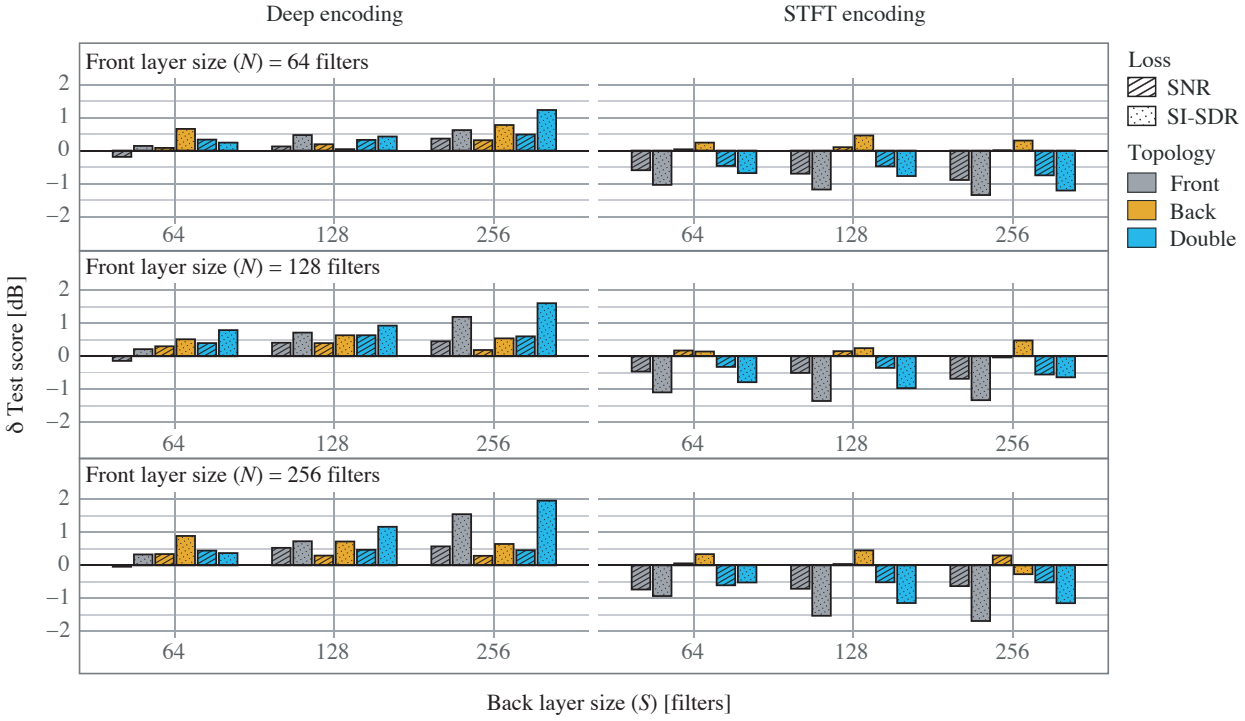

Fig. 8. Bar plots of the increment in the speech enhancement testing error with respect to the independent model (δ test score = $\text{Loss}_{ind} - \text{Loss}_{\Lambda}$) of the different fused models as a function of fusion size, loss function, and encoding type.

TABLE II
ABSOLUTE VALIDATION AND TESTING VALUES OF THE INDEPENDENT MODEL FOR THE DIFFERENT TESTED LOSS FUNCTIONS AND ENCODINGS. BOLD VALUES SHOW THE BEST PERFORMANCE FOR EACH LOSS.

Validation Loss/Test Loss [dB]

Objective	SNR		SI-SDR	
Enc. / N/S	Deep	STFT	Deep	STFT
64/64	9.18/9.23	8.91/8.83	15.63/15.74	15.75/15.34
64/128	9.15/9.18	8.89/8.77	16.97/15.84	15.85/15.44
64/256	9.26/9.26	8.90/8.94	15.89/15.99	15.87/15.25
128/64	9.29/9.28	9.35/9.15	15.91/15.98	16.94/16.25
128/128	9.23/9.26	9.32/9.09	16.01/16.01	16.99/16.44
128/256	9.28/9.33	9.44/9.29	16.11/16.21	17.11/16.71
256/64	9.23/9.26	9.84/**9.56**	15.88/16.02	17.90/16.99
256/128	9.37/9.42	9.91/9.45	15.95/16.01	18.01/16.90
256/256	9.42/9.51	9.79/9.54	15.95/15.86	18.13/**17.51**

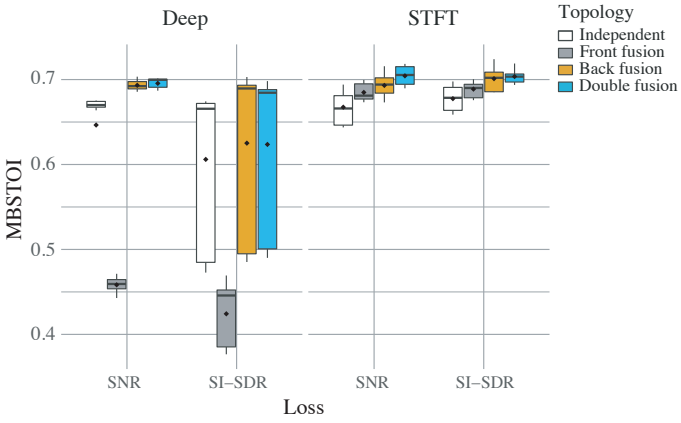

Fig. 9. Box plots indicating the MBSTOI scores on the testing set for the different tested models and encodings. The black horizontal bars within each of the boxes represent the median, the diamond-shaped marks indicate the mean improvement, and the top and bottom extremes of the boxes indicate the 75% and 25% quartiles.

decoding. This suggests that the shared information between sides is learned.

To investigate how the number of fused channels between the left and right speech enhancement models impact the testing error, we correlated the total amount of fused channels to the objective test loss, for the different encodings and loss functions. Figure 11 shows the relation of the performance difference between the fused and independent models as a function of the total number of fused latent channels and encoding type.

This plot corroborates that a deep encoding is necessary in order to take advantage of the fusion layers, as we can see that not only the STFT deterministic encoding is negatively correlated to the total number of fused channels (frequency bins when fusing the encoder outputs) but also that this encoding performs generally poorer than the independent model.

To further assess the effect of the fusion layers on speech enhancement we computed the modified binaural short-time objective intelligibility (MBSTOI) [44] for each of the deep learning topologies. Figure 9 shows the box plots depicting the MBSTOI for the independent and fused models for each tested encodings and loss functions. A visual analysis of this plot seems to suggest that the fused models obtain higher average MBSTOI scores when compared to the independent model.

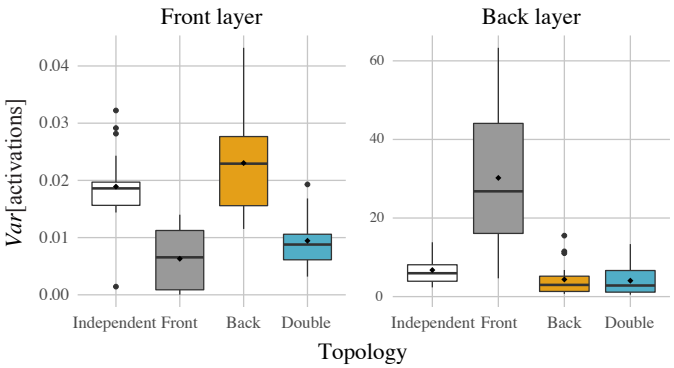

Fig. 10. Box plots indicating the activation variance on the testing set. The black horizontal lines within boxes represent the median, the diamond-shaped marks indicate the mean improvement, and the top and bottom extremes of the boxes indicate the 75% and 25% quartiles.

However, unlike the case of noise reduction amount (see Figure 8) the STFT encoding obtained higher performance than the deep encoding. This indicates that there may be a trade-off between the noise reduction amount and the preservation of binaural cues, which MBSTOI relies on to compute its intelligibility score.

2) Layer variance analysis: Figure 10 shows a box plot of the layer activation variances of the different speech enhancement algorithms tested in this study. The left panel shows the layer variance of the encoder output (note that this analysis is only applicable for the deep non-deterministic encoding) and the right panel shows the variance of the temporal convolution outputs. It can be seen that the activation variance is again affected by the fusion operation. For example, note how the single fusion models obtained an unbalanced variance being smaller where the fusion operation is performed.

The fusion operation causes a reduced layer activation variance. The double fusion model obtains activation values at the front and back layers that are numerically closer to each other, compared to the other three models. Fundamentally, this may indicate that the fusion operation causes the gradients two propagate between the left and right enhancement modules, acting as a channel that balances the learning rate.

IV. Conclusion

In this manuscript, we propose deep fusion layers to improve speech enhancement in binaural listening. We first introduce and formalize the concept of the general fused model, defining its basic notation and describing its properties. Specifically, we prove that fusion layers introduce non-linearities in the model allowing it to fit the input data distribution better. We also show empirically that fused models are susceptible to input decorrelation. Finally, we analyze the effect of the fusion layers on binaural speech enhancement. Results indicate that fused models may be promising in terms of noise reduction when compared to independent models. In fact, based on our experiments, the model using the largest double fusion layers performs the best with respect to the other topologies on unseen data. However, based on the MBSTOI measure, we also show that there is a trade-off between noise reduction and

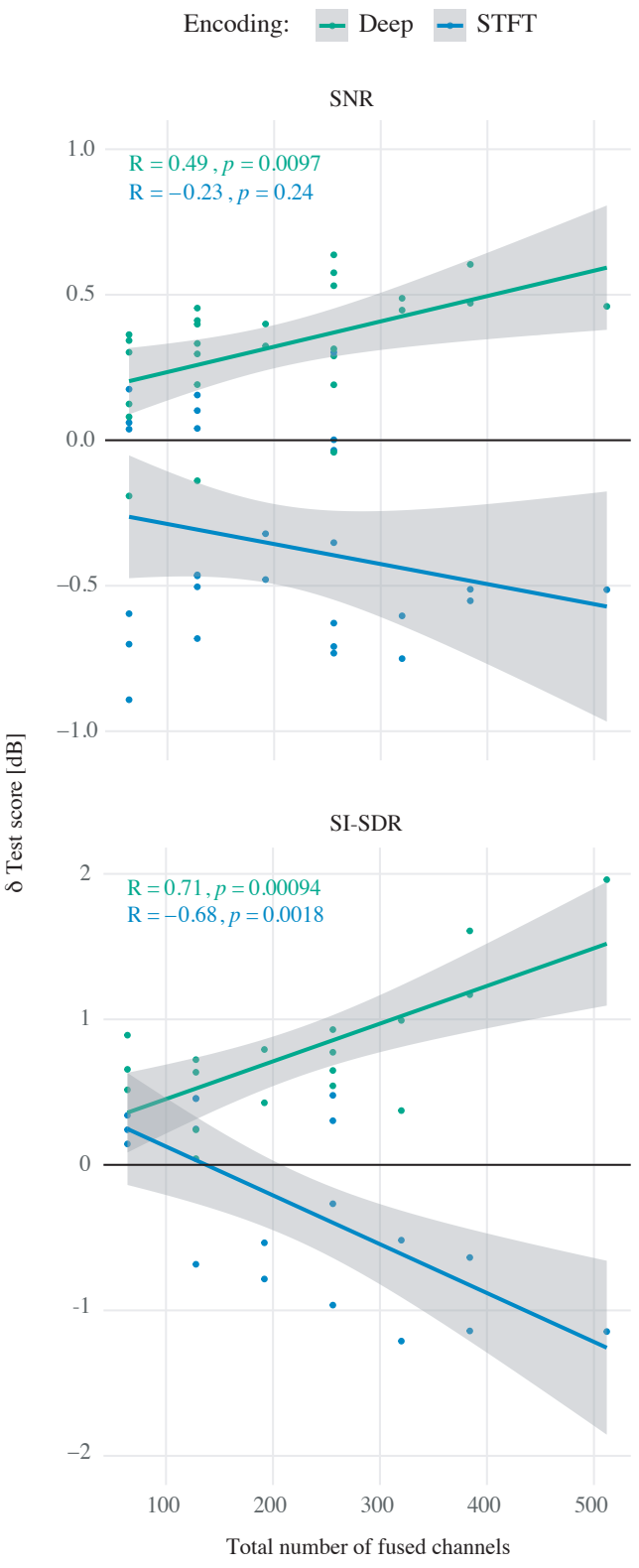

Fig. 11. Regression of the testing error difference between the fused and independent models as a function of the number of the total number of fused channels for each of the investigated encoders. Shaded areas represent a point-wise 95% confidence interval on the fitted values. Correlation analysis is expressed as the adjusted-R and p-value, and it is considered to be significant when $p < 0.05$.

predicted speech intelligibility, potentially due to distortions introduced by largely fused models. Based on these results we think that this approach could potentially be beneficial for future binaural speech processing systems.

It is important to notice that in this work we assume the transmission of information between listening sides is instantaneous. It should be pointed out that in real-life applications this would not be the case. A relevant aspect to investigate is how the latency and the bitrate reduction required for the transmission of the latent spaces affect the performance of the fused models.

REFERENCES

[1] Q. Liu, Y. Xu, P. J. B. Jackson, W. Wang, and P. Coleman, "Iterative deep neural networks for speaker-independent binaural blind speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 541–545.

[2] C. Han, Y. Luo, and N. Mesgarani, "Real-time binaural speech separation with preserved spatial cues," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6404–6408.

[3] J. Deng, W. Dong, R. Socher, L. Li, L. Kai, and F. Li, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[4] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[5] W. Kang, Q. Yang, and R. Liang, "The comparative research on image segmentation algorithms," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 2, 2009, pp. 703–707.

[6] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.

[7] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8599–8603.

[8] U. Kamath, J. Liu, and J. Whitaker, *Deep learning for NLP and speech recognition*. Springer, 2019, vol. 84.

[9] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, pp. 1256–1266, 2019.

[10] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5069–5073.

[11] N. Zeghidour and D. Grangier, "Wavesplit: End-to-end speech separation by speaker clustering," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2840–2849, 2021.

[12] J. Lin, A. van A. J. Wijngaarden, K. Wang, and M. C. Smith, "Speech enhancement using multi-stage self-attentive temporal convolutional networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3440–3450, 2021.

[13] J. Hou, S. Wang, Y. Lai, J. Lin, Y. Tsao, H. Chang, and H. Wang, "Audio-visual speech enhancement using deep neural networks," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1–6.

[14] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[15] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[16] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, no. 1, pp. 30–43, 2018.

[17] M. Yang, W. Zhao, W. Xu, Y. Feng, Z. Zhao, X. Chen, and K. Lei, "Multitask learning for cross-domain image captioning," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1047–1061, 2019.

[18] C. Wang, H. Yang, and C. Meinel, "Image captioning with deep bidirectional lstms and multi-task learning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2s, pp. 1–20, 2018.

[19] X. Tan and X. Zhang, "Speech enhancement aided end-to-end multi-task learning for voice activity detection," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6823–6827.

[20] T. G. Kang and N. S. Kim, "Dnn-based voice activity detection with multi-task learning," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 2, pp. 550–553, 2016.

[21] J. Liang, E. Meyerson, and R. Miikkulainen, "Evolutionary architecture search for deep multitask networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Association for Computing Machinery, 2018, p. 466–473.

[22] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2019.

[23] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[24] Z. Hu, Z. Su, Y. Li, and J. Ma, "Adaptive cross-stitch graph convolutional networks," in *ACM Multimedia Asia*, 2021, pp. 1–7.

[25] P. Avan, F. Giraudet, and B. Büki, "Importance of binaural hearing," *Audiology and Neurotology*, vol. 20, no. Suppl. 1, pp. 3–6, 2015.

[26] B. C. J. Moore, *An introduction to the psychology of hearing*. Brill, 2012.

[27] T. Lotter and P. Vary, "Dual-channel speech enhancement by superdirective beamforming," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, pp. 1–14, 2006.

[28] T. Rohdenburg, V. Hohmann, and B. Kollmeier, "Robustness analysis of binaural hearing aid beamformer algorithms by means of objective perceptual quality measures," in *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 2007, pp. 315–318.

[29] B. Cornelis, S. Doclo, T. V. dan Bogaert, M. Moonen, and J. Wouters, "Theoretical analysis of binaural multimicrophone noise reduction techniques," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 342–355, 2009.

[30] S. Chakrabarty, D. Wang, and E. A. P. Habets, "Time-frequency masking based online speech enhancement with multi-channel data using convolutional neural networks," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 476–480.

[31] K. Tan, B. Xu, A. Kumar, E. Nachmani, and Y. Adi, "SAGRNN: Self-attentive gated rnn for binaural speaker separation with interaural cue preservation," *IEEE Signal Processing Letters*, vol. 28, pp. 26–30, 2021.

[32] D. R. Moore, "Anatomy and physiology of binaural hearing," *Audiology*, vol. 30, no. 3, pp. 125–134, 1991.

[33] J. Pickles, "An introduction to the physiology of hearing," in *An Introduction to the Physiology of Hearing*. Brill, 1998.

[34] N. I. Durlach, "Equalization and cancellation theory of binaural masking level differences," *The Journal of the Acoustical Society of America*, vol. 35, no. 8, pp. 1206–1218, 1963.

[35] J. F. Culling, "Evidence specifically favoring the equalization-cancellation theory of binaural unmasking," *The Journal of the Acoustical Society of America*, vol. 122, no. 5, pp. 2803–2813, 2007.

[36] S. Graetzer, J. Barker, T. J. Cox, M. Akeroyd, G. N. J. F. Culling, E. Porter, and R. V. Muñoz, "Clarity-2021 challenges: Machine learning challenges for advancing hearing aid processing," in proceedings of the annual conference of the international speech communication association," in *INTERSPEECH 2021*, Brno, Czech Republic, 2021.

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

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

[40] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993.

[41] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (DEMAND): A database of multi-channel environmental noise recordings," *Proceedings of Meetings on Acoustics*, vol. 19, no. 1, p. 035081, 2013.

[42] C. Hummersone, R. Mason, and T. Brookes, "Dynamic precedence effect modeling for source separation in reverberant environments," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1867–1871, 2010.

[43] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 626–630.

[44] A. H. Andersen, J. Mark, Z. Tan, and J. Jensen, "Refinement and validation of the binaural short time objective intelligibility measure for spatially diverse conditions," *Speech Communication*, vol. 102, pp. 1–13, 2018.