Deep Latent Fusion Layers for Binaural Speech Enhancement

Tom Gajecki & Waldo Nogueira

***Abstract*—This work addresses the issue of enhancing speech in binaural hearing scenarios. Specifically, we present a method to improve binaural noise reduction by integrating latent features produced by monaural speech enhancement algorithms through the use of "Fusion layers." These layers perform Hadamard products between latent spaces at specific processing stages. These fusion layers draw inspiration from multi-task learning techniques, which involve sharing model weights across various models aimed at handling interconnected tasks. The layers perform element-wise dot products between tensors representing latent representations at the same processing stage, mimicking the physiological excitatory and inhibitory mechanisms of the binaural hearing system. This study initially presents a general fusion model, demonstrating its ability to better fit synthetic data compared to 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. The proposed method shows promise for improved noise reduction compared to other feature-sharing approaches. The study also suggests that including fusion can enhance predicted speech intelligibility and quality, but too many fused features may have a negative impact on expected speech intelligibility. Furthermore, the results suggest that fusion layers should share parameterized latent representations to effectively utilize information from each listening side, rather than using deterministic representations. Overall, this study highlights the potential of sharing information between speech enhancement modules through deep fusion layers to improve binaural speech enhancement while maintaining constant trainable parameters 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 potentially 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 an even 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]–[30]). 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

---

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

has used feature concatenation at the input level to perform binaural speech enhancement (e.g., [2], [31]). These methods have been shown to improve speech enhancement performance when compared to independent models, however, they rely on explicit spectral feature extraction and are not necessarily motivated by the human binaural auditory system.

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 the physiological excitatory and inhibitory mechanisms that occur in the binaural hearing system [36], we investigate the influence that sharing the latent representations of two single-channel end-to-end speech enhancement DNNs has on the speech enhancement performance of binaural noisy speech signals. The latent representations are shared through fusion layers that apply element-wise dot product operations to each of the features contained in them. These layers are designed to introduce non-linearities to the learning model that will allow better fitting of 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. Here we extend a previous study[1] presented at the 2021 Clarity speech enhancement challenge [37] by formalizing the concept and by analyzing the effect of input data correlation, latent activation variance, and encoding methods.

This work proposes a method for improving binaural speech enhancement by combining latent representations generated by DNNs using "Fusion layers." These layers perform element-wise dot products between tensors representing latent representations at the same processing stage, inspired by the physiological excitatory and inhibitory mechanisms of the binaural hearing system. The proposed method shows potential for better noise reduction compared to other data merging methods like spectral feature concatenation, and for improving predicted speech intelligibility and quality. However, fusing too many features can have a negative impact on predicted speech intelligibility. This highlights the need for caution when using fusion to prevent excessive degradation of output signals.

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., $\mathbf{X}_{l,m} = \omega_{l,m}(\mathbf{X}_{l-1,m})$, $l = \{1, \ldots, L\}$ (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, and; 3) Act as a channel for the gradients to back-propagate through, to reduce the activation variance between learning modules and improve generalization on unseen data [38].

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

---

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

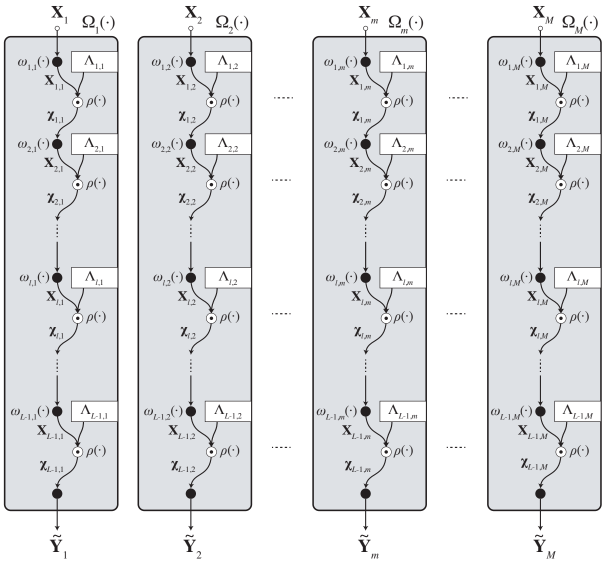
$\mathbf{X}_1$ $\Omega_1(\cdot)$
$\mathbf{X}_2$ $\Omega_2(\cdot)$
$\mathbf{X}_m$ $\Omega_m(\cdot)$
$\mathbf{X}_M$ $\Omega_M(\cdot)$
$\omega_{1,1}(\cdot)$ $\Lambda_{1,1}$ $\mathbf{X}_{1,1}$ $\rho(\cdot)$ $\chi_{1,1}$
$\omega_{2,1}(\cdot)$ $\Lambda_{2,1}$ $\mathbf{X}_{2,1}$ $\chi_{2,1}$
$\omega_{l,1}(\cdot)$ $\Lambda_{l,1}$ $\mathbf{X}_{l,1}$ $\chi_{l,1}$
$\omega_{L-1,1}(\cdot)$ $\Lambda_{L-1,1}$ $\mathbf{X}_{L-1,1}$ $\chi_{L-1,1}$
$\tilde{\mathbf{Y}}_1$ $\tilde{\mathbf{Y}}_2$ $\tilde{\mathbf{Y}}_m$ $\tilde{\mathbf{Y}}_M$

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.

by assuming that all learning modules (i.e., fully connected layers) are linear, and that input tensors are vectors $\mathbf{X}_m \in \mathbb{R}^{1\times T}$, where T can be interpreted as the number of time steps. 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 referred to as "independent" (i.e., $\Lambda_{l,m} = \{J_l\}\ \forall\ l, m$). 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 $\mathbf{w}_{l,m} \in \mathbb{R}^{(n_{l-1})\times n_l}$ 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 let us define a fused model where all layers are multiplied with each other for all learning modules, that is $\Lambda_{l,m} := \{\mathbf{X}_{l,m'}\ \forall\ l \land m' \neq m\}$. 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 as the size of the fusion set, that is, $|\Lambda_{l,m}|$ (where $|A|$ represents the cardinality of a set A). In general, the fusion output at layer l in a fully fused model (side-wise fusion level of $|\Lambda_{l,m}| = M-1$) is given by:

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

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:

$$\mathrm{LOT}\left(\frac{\partial \chi_{l,m}}{\partial \mathbf{X}_{l-1,m}}\right) \sim \mathcal{O}\Big(n^{M-1}\Big)\ \forall\ M \geq 1. \tag{3}$$

Depth-wise fusion level is here defined as the number of fusion operations that precede the deepest fusion layer. It 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 be not only dependent 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:

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

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 m such that the LOT is:

$$\mathrm{LOT}\left(\frac{\partial \chi_{L-1}}{\partial \mathbf{X}_{L-2}}\right) \sim \mathcal{O}\left(n^{(M-1)\cdot\left(2^{(L-1)}-1\right)}\right)\ \forall\ M \geq 1,\ L > 1. \tag{5}$$

It is important to note that the special case where $M = 1$ leads to a model with no fusion operations, where $|\Lambda_{l,1}| = 0\ \forall\ l$, and the output is reduced to a linear regression.

III. RESULTS

A. Experiment 1: Study on synthetic data

In this experiment, our objective is to examine the impact of the fusion operation on basic regression problems using a dataset generated artificially. We partition this experiment into two sub-experiments. The first one will demonstrate through empirical evidence that the operation presented in equation 2 introduces non-linearities. In the second sub-experiment, we explore the trade-off between the correlation of the input data in each sub-model and its fitting capabilities.

Model: In this experiment we will keep the number of sub-models $m = 2$ (as shown in Figure 2). All learning modules are fully connected layers with linear activation functions. 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 n_l, 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 (T = 500, see Figure 3, first panel). From the input data, we generated

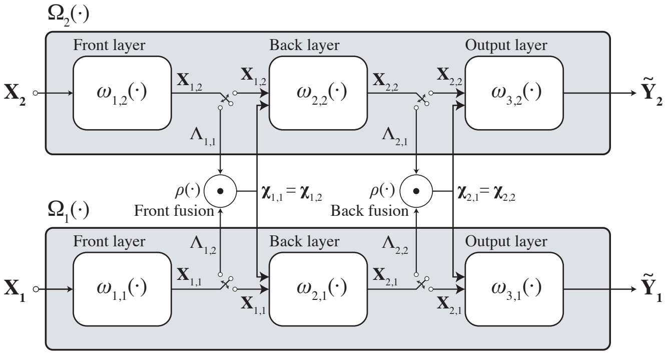

Fig. 2. Block diagram of a model comprised of two deep learning sub-models, each containing three learning modules. Fusion layers are included or bypassed using the switches depicted in the block diagram. Each sub-model is represented by the grey blocks and each of the learning modules is represented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

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:

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

where $\mathbf{z}_1 = \mathbf{X}_1$, $\mathbf{z}_2 = \mathbf{X}_1 \cdot (1-d) + d \cdot \mathbf{X}_2$. $\mathbf{X}_{n1}$ 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, i.e., identical input signals, 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 $\tilde{\mathbf{Y}}$. The MSE computed over n samples is defined as:

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

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 [39] was used to minimize the MSE (see equation 7) between the estimated and true outputs.

1) Visual intuition: An illustrative example of how the output of a model of size $n_l = 64$, for $l = \{2, 3\}$ (see Figure 2) is affected by the addition of fusion layers is shown in Figure 3. The first panel shows the raw data generated by equation 6. The second panel shows the data fitted by an independent model. 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 quartic polynomial regression.

a) Independent model: The data regressions obtained with this model can be seen to be linear for both predicted outputs (Figure 3, second panel). In this model, the two sub-models shown in Figure 2 are disconnected, that is, latent representations at any stage are independent of each other. This is equivalent to having a side-wise and depth-wise fusion level of zero. Because no fusion layers are present throughout the model, we can apply equation 5 for $M = 1$, obtaining outputs that satisfy $\partial \tilde{\mathbf{Y}}_{1,2} / \partial \mathbf{X} = C_{1,2} \sim \mathcal{O}\big(n^0\big)$, i.e., linear regressions.

b) Single fusion model: The regressions produced by this model display a quadratic trend in both predicted outputs, as depicted in the third panel of Figure 3. In this case, we fuse the latent representations of the model between two fully connected layers (note that it does not matter whether is between $l = 1$ and $l = 2$ or $l = 2$ and $l = 3$, because of the symmetry of the model, that is, all deep learning modules have the same dimensionality). The fusion operation performed in this case is a one-sided fusion and not a depth-wise fusion. Therefore, we apply equation 3 for $M = 2$, which satisfies $\partial \tilde{\mathbf{Y}}_{1,2} / \partial \mathbf{X} = f(\mathbf{X})_{1,2} \sim \mathcal{O}\big(n^1\big)$, which can be seen by the unique global minima in the third panel of Figure 3. Note that in this example both quadratic functions show a convex nature, caused by the fact that most raw target data points are located in the top half of the panel. One would expect the second derivative of the output regressions to change sign if the target data would be vertically flipped around $\text{Y} = 0.5$.

c) Double fusion model: In this case, the model's regressions are represented by quartic functions for both outputs, as illustrated in the fourth and last panel of Figure 3. This model presents a depth-fusion level of one, which in this particular model represents a fully fused model. For this reason, we can apply equation 5 for $M = 2$ and $L = 3$, which will satisfy $\partial \tilde{\mathbf{Y}}_{1,2} / \partial \mathbf{X} = f(\mathbf{X})_{1,2} \sim \mathcal{O}\big(n^3\big)$, which can be seen by the three function turning points in each regression.

2) Experiment 1.1: In this experiment, we aim to investigate 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 compared to a completely linear model.

Figure 4 shows box plots of the MSE improvement given by the fused models with linear activations computed as $\delta\text{MSE} = \text{MSE}_{ind} - \text{MSE}_{\Lambda}$, 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.

3) Experiment 1.2: In this experiment, our goal is to explore the sensitivity of the proposed attention mechanism to variations between inputs in each sub-model. To achieve this, we will calculate the errors at the outputs of both the individual and fused models, based on the correlation of the input data. This investigation is crucial due to the motivation behind employing fusion layers in binaural speech enhancement systems, where there is a presence of correlation

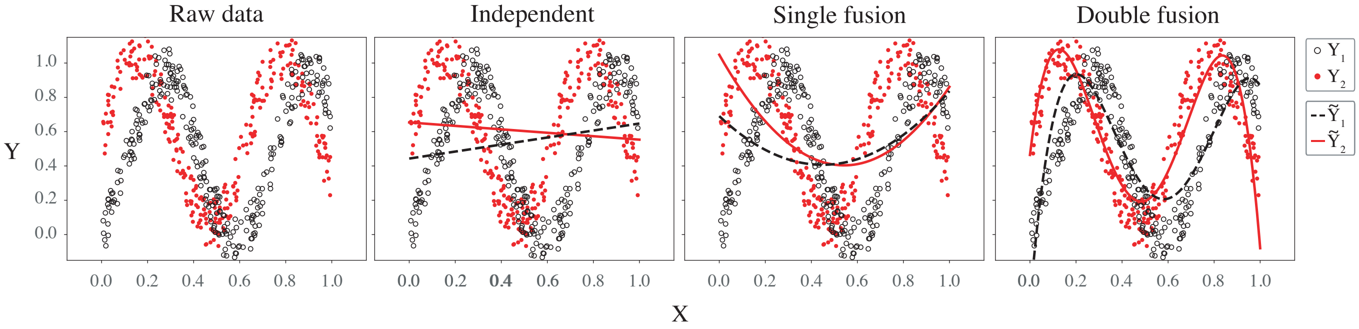

Fig. 3. Data regression plots obtained by the independent and fused models on generated synthetic data. Left most plot shows the raw output data Y as a function of the input data X for the left and right channels, and the remaining three plots show the obtained regressions on top of it.

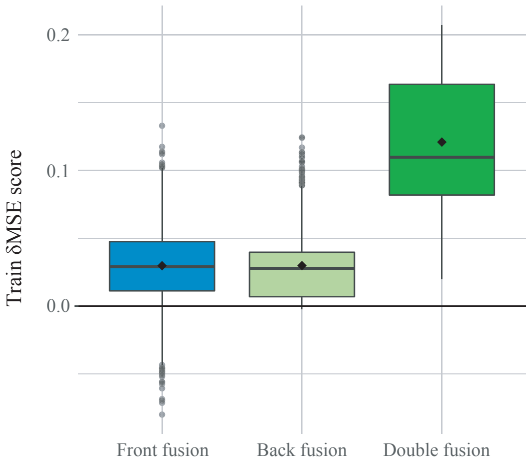

Fig. 4. Box plots showing the increment in MSE error of the different fused models w.r.t. the independent linear model ($\delta\text{MSE} = \text{MSE}_{ind} - \text{MSE}_{\Lambda}$), for the front, back, and double fusion. The black horizontal bars within each box represent the median for each condition, the circle-shaped marks indicate the mean improvement, and the top and bottom extremes of the boxes indicate the $Q_3 = 75\%$ and $Q_1 = 25\%$ quartiles, respectively. The box length is given by the interquartile range (IQR), used to define the whiskers that show the variability of the data above the upper and lower quartiles (the upper whisker is given by $Q_3 + 1.5{\cdot}\text{IQR}$ and the lower whisker is given by $Q_1 - 1.5{\cdot}\text{IQR}$ [40]). Black dots indicate observations that fall beyond the whisker range (outliers).

between hearing sides. However, our aim is to determine a potential threshold below which fusion might not provide significant benefits in fitting the training data distribution.

Figure 5 shows a dot plot together with its polynomial regression showing how the 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., $\delta\text{MSE} > 0$), indicating that the introduced non-linearities do help the model fit the input training data more accurately.

4) 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

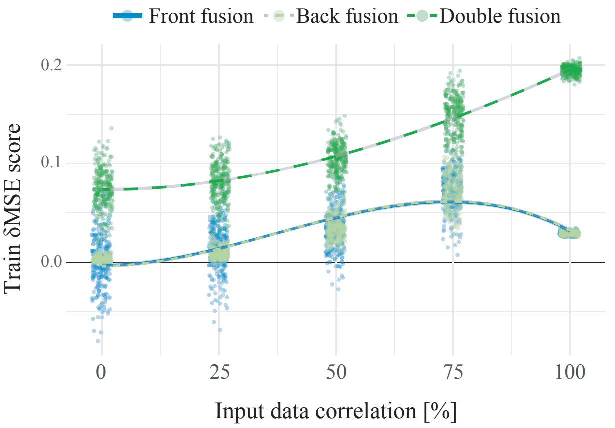

Fig. 5. Dot plot of the training error differences between the independent and fused models ($\delta\text{MSE} = \text{MSE}_{ind} - \text{MSE}_{\Lambda}$) as a function of input data correlation for generated synthetic data. A second-order polynomial regression is included to show the performance trend for each condition.

contained in each layer is defined as:

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

where $\mathbb{E}[\cdot]$ is the expected value operator, $\mathbf{w}_{l,m}$ is the tensor containing all of the learned weights in layer l in model m, and $\overline{\mathbf{w}}_{l,m}$ 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.

Figure 6 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 6 how the violin plots show less deviation from the

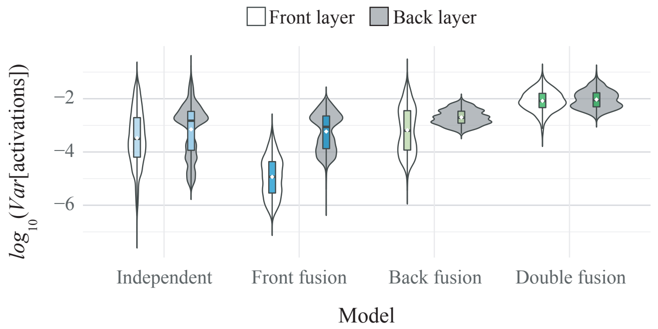

Fig. 6. Violin plots indicating the activation variance across predictions for the front and back fusion layers (see Figure 2) in the different models for generated synthetic data. Data are plotted on a logarithmic scale for visualization purposes. The black horizontal bars within each box represent the median for each condition, the circle-shaped marks indicate the mean improvement, and the top and bottom extremes of the boxes indicate the $Q_3 = 75\%$ and $Q_1 = 25\%$ quartiles, respectively. The box length is given by the interquartile range (IQR), used to define the whiskers that show the variability of the data above the upper and lower quartiles (the upper whisker is given by $Q_3 + 1.5{\cdot}\text{IQR}$ and the lower whisker is given by $Q_1 - 1.5{\cdot}\text{IQR}$ [40]). Black dots indicate observations that fall beyond the whisker range (outliers).

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 [38].

B. Experiment 2: Ablation study

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 investigated fusion method will be investigated in the context of a well-known fully-convolutional time-domain audio separation network (Conv-TasNet [9]; which will we be referring to as TasNet for simplicity). In this ablation experiment we analyze the effect of introducing and/or removing fusion layers between specific latent representations of the input signals. The TasNet relies on two end-to-end audio speech enhancement models; each consisting of three processing stages, as shown in Figure 7: an encoder, a separator (a temporal convolution module (TCN), 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 enhanced speech is resynthesized by the decoder. The utilized range of hyperparameters is presented in detail in Table I. The implementation was done in TensorFlow 2.0 [41] and the code for training and evaluating can be found online[2].

TABLE I

HYPERPARAMETERS USED TO TRAIN THE DEEP LEARNING MODELS

Parameter	Value
N (Encoding size)	Variable (*N*)
L (Filter length)	64
B (Bottleneck size)	64
S (Skip-connection size)	Variable (*S*)
H (Number of convolution channels)	256
P (Convolution kernel size)	256
X (Number of convolutions in each repeat)	2
R (Number of repeats)	2
Model size range (per listening side)	[430k, 600k]

Dataset: The speech material used for the evaluation of the speech enhancement models was obtained from the TIMIT acoustic-phonetic Continuous Speech Corpus [42] (consisting of a set dedicated to training and another set for testing). TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences. The TIMIT corpus includes time-aligned orthographic, phonetic, and word transcriptions as well as a 16-bit, 16-kHz speech waveform file for each utterance. The speech data contained in this corpus consists of fluent spoken sentences with a total duration of 18 hours.

The interfering noisy signals were all obtained from the DEMAND collection of multi-channel recordings of acoustic noise in diverse environments [43]. The environmental noises recorded to create this dataset are split into six categories; four are indoor noises and the other two are outdoor recordings. The indoor environments are further divided into domestic, office, public, and transportation; the open-air environments are divided into streets and nature. There are 3 environment recordings per category.

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. As a preprocessing stage, all audio material was ensured to be stereo and sampled at 16 kHz.

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) [44] 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 II.

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 explore the impact of fusion operations on the performance of models when using different encodings of the input signals. Our investigation focuses on comparing

---

[2] https://github.com/tomgajecki/FusionLayers

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

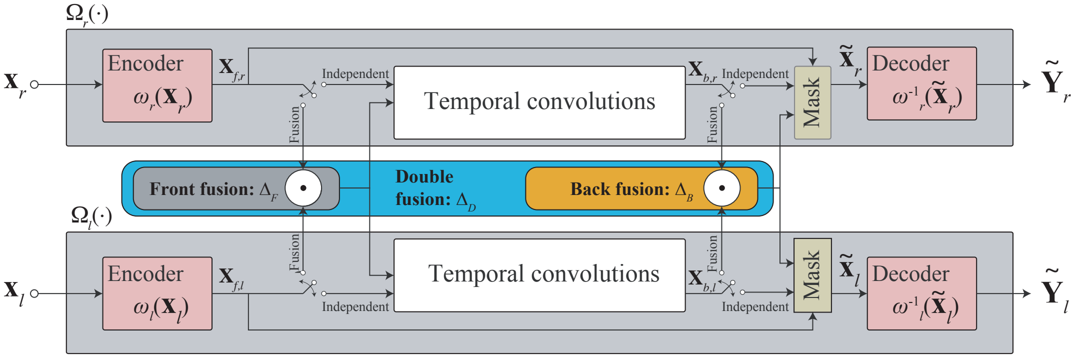

Fig. 7. Block diagram of the evaluated algorithms. "Independent" model bypasses both fusion layers. "Front fusion (Δ_F)" model, "Back fusion (Δ_B)" model, and "Double fusion (Δ_D)" model.

TABLE II
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	$\Delta_D = \begin{pmatrix} \{\mathbf{X}_{f,r}\} & \{\mathbf{X}_{f,l}\} \\ \{\mathbf{X}_{b,r}\} & \{\mathbf{X}_{b,l}\} \end{pmatrix}$

a non-deterministic learned representation and a deterministic representation. The objective of this analysis is to examine whether these fusion layers effectively utilize redundant binaural data by sharing underlying representations among models through the inclusion of adaptable non-linearities that align with the input 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). These encoding-decoding stages are represented by the encoder-decoder blocks shown in Figure 7.

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) [45]. 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:

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

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 5-epoch patience was applied as a regularization method, and only the best-performing model was saved. For the model optimization, Adam [39] 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.

1) Absolute speech denoising performance with no fusion layers: Table III 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 the 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 details refer to [9]).

2) Relative denoising performance with fusion layers: To assess the generalization capabilities of the fusion layers, we will be reporting on the test score difference (δ) of the different fused models concerning the values shown in Table III. Figure 9 shows bar plots of the increment in the validation and testing error (δ Test score = $\mathrm{Loss}_{\Lambda} - \mathrm{Loss}_{ind}$) of the different fused models (see Table II) 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 deterministic STFT encoding, the fusion mechanisms may blur or distort the signal and fail to produce final faithful decoding. This suggests that the shared information between sides is learned.

3) Speech denoising performance as a function of the number of fused channels: To investigate how the number of fused channels between the left and right speech enhancement

TABLE III

ABSOLUTE VALIDATION AND TESTING SNR (FIRST TWO COLUMNS) AND SI-SDR (LAST TWO COLUMNS) VALUES OBTAINED BY THE INDEPENDENT MODELS FOR THE DIFFERENT TESTED LOSS FUNCTIONS AND ENCODINGS. BOLD VALUES SHOW THE BEST PERFORMANCE FOR EACH LOSS.

Validation Loss/Test Loss [dB]

Loss	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**

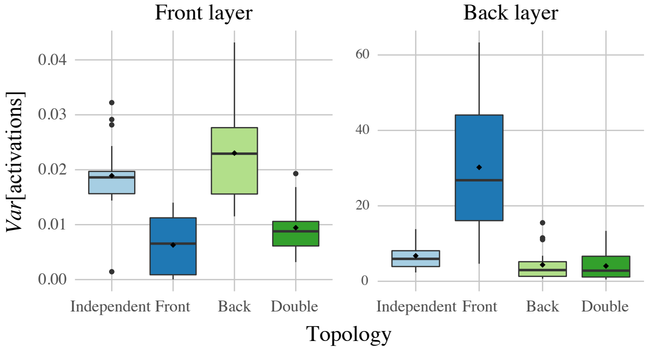

Fig. 8. Box plots indicating the activation variance on the testing set. The black horizontal bars within each box represent the median for each condition, the circle-shaped marks indicate the mean improvement, and the top and bottom extremes of the boxes indicate the $Q_3 = 75\%$ and $Q_1 = 25\%$ quartiles, respectively. The box length is given by the interquartile range (IQR), used to define the whiskers that show the variability of the data above the upper and lower quartiles (the upper whisker is given by $Q_3 + 1.5{\cdot}\text{IQR}$ and the lower whisker is given by $Q_1 - 1.5{\cdot}\text{IQR}$ [40]). Black dots indicate observations that fall beyond the whisker range (outliers).

models impacts the testing error, we correlated the total amount of fused channels to the objective test loss, for the different encodings and loss functions. Figure 10 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 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 generally performs poorer than the independent model.

4) Layer variance analysis of the different speech denoising topologies: Figure 8 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 gradient to propagate between the left and right enhancement modules, acting as a channel that balances the learning rate.

C. Experiment 3: Comparative study

In this section, we assess the effect of fusion compared with other baseline models. We also extend the baseline models by introducing fusion layers to assess their efficacy in improving binaural speech enhancement. All the tested models based on TCN separation share the same hyper-parameters shown in Table I with $N = S = 256$, and all with deep encoders and decoders. For all the other models, the number of trainable parameters was set to be roughly the same as the rest. To further assess the effect of the fusion layers on speech enhancement we computed the modified binaural short-time objective intelligibility (MBSTOI metric [46]); for each deep learning topology. MBSTOI is an extension of STOI [47] that includes a modified version of the equalization-cancellation model and enables predictions including binaural advantages, while also maintaining the monaural performance of the STOI measure.

To support this last analysis we include the averaged STOI [47] across listening sides and to monitor the quality of the separated speech we also include the PESQ [48] measure, also averaged across listening sides.

1) Final tested models:

a) Independent [9]: This is the simplest baseline model and it is comprised of two TasNets performing single-channel speech enhancement on each listening side independently.

b) CDNN [49]: In this model, binaural speech enhancement is performed by means of a fully connected complex DNN. Here, signals in the left and right channels are considered as the real and imaginary components of a monaural complex signal. Unlike alternative models, this architecture undergoes the challenge of estimating a complex ideal ratio mask.

c) Front fusion: This model uses two TasNets connected with a fusion layer after the encoding blocks, as defined in Table II, second row.

d) Concat [2]: This model uses spectral feature concatenation after the encoding stage.

e) Concat + Fusion: This model extends the "Concat" model by introducing a back fusion layer (defined in Table II, third row) after the TCN outputs, as shown in Figure 7 (see back fusion block).

f) Stitch [23]: In this configuration we substitute the front fusion operation with a cross-stitch network. The inputs to the TCN module at each side ($X'_{\{f,b\},\{r,l\}}$, following the notation shown in Figure 7) are given by:

$$\begin{bmatrix} X'_{\{f,b\},r} \\ X'_{\{f,b\},l} \end{bmatrix} = \begin{bmatrix} \alpha_{rr} & \alpha_{rl} \\ \alpha_{lr} & \alpha_{ll} \end{bmatrix} \begin{bmatrix} X_{\{f,b\},r} \\ X_{\{f,b\},l} \end{bmatrix}, \tag{11}$$

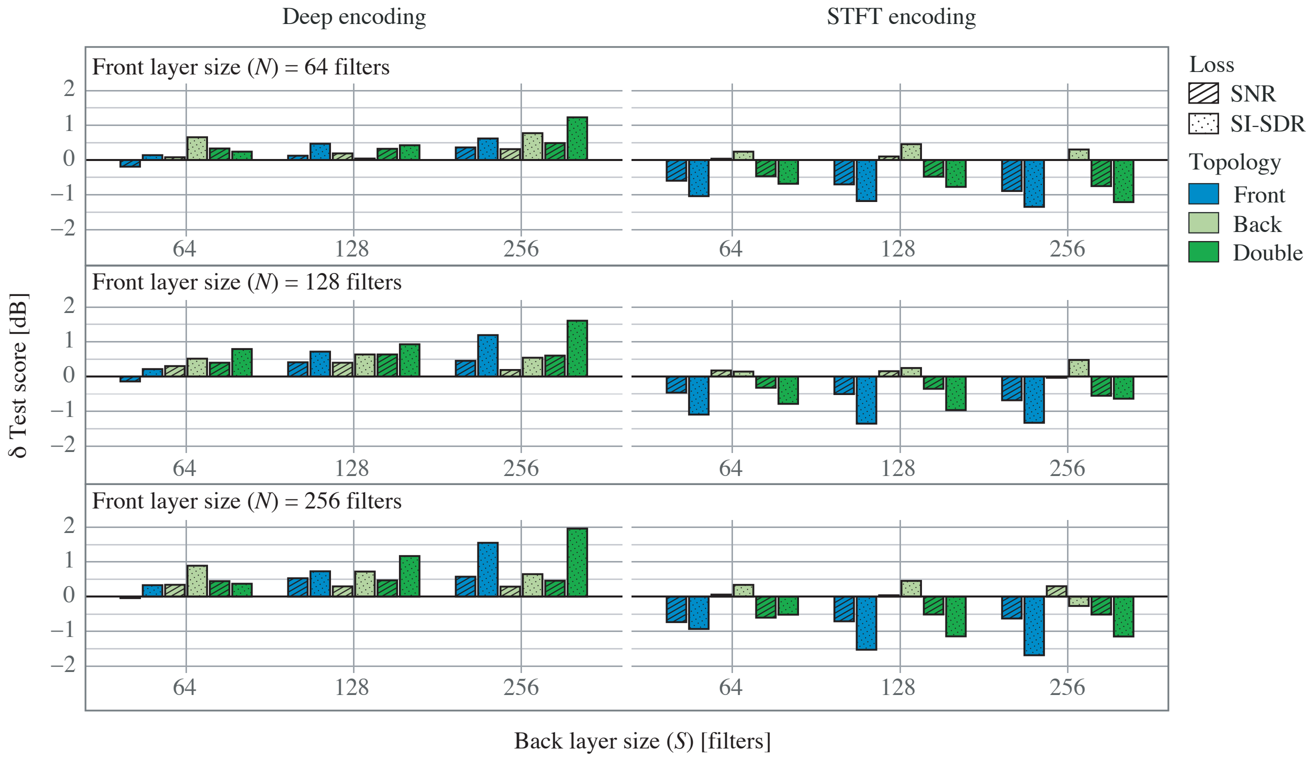

Fig. 9. Bar plots of the increment in the speech enhancement testing error with respect to the independent model (δ test score = Loss$_{ind}$ − Loss$_{\Lambda}$) of the different fused models as a function of fusion size, loss function, and encoding type.

where α_{ij} for $i, j \in \{r, l\}$ are trainable parameter tensors of adequate dimensionality.

g) Stitch+Fusion [23]: This model extends the "Stitch" model by introducing a back fusion layer (defined in Table II, third row) after the TCN outputs, as shown in Figure 7 (see back fusion block).

h) Parallel Concat [2]: This model is described in [2] as "parallel encoder + sum & mask," and here we concatenate the encoded spectral features obtained from the encoders.

i) Parallel Fusion: This model is architecturally identical to the model "Parallel Concat," but we replace the intermediate feature concatenation layer with a fusion layer.

j) Parallel Cross [31]: This model is based on two TasNets using two encoders per channel and shares cross-domain features. Specifically, cross-channel features are concatenated to the encoder output using interaural time and level differences as spatial features. An implementation of this model can be found online[4].

k) Parallel Cross+Fusion: This model adds a back fusion layer (defined in Table II, third row) to model "Parallel Cross."

l) Double Fusion: This model fuses the latent representations after the decoder and TCN outputs, as shown in Figure 7 and defined in Table II, last row.

2) Final performance results: Table IV shows the objective measures for each of the tested models using the SNR loss and Table V the results using the SI-SDR loss. It can be seen that the proposed fusion operation improves noise reduction performance for all baseline models. However, it can also be seen that in general, the fusion operation causes a slight drop in predicted speech intelligibility and quality, which may be related to the potential presence of artifacts and distortions introduced by the operation. This observation aligns with the findings of [50] where they demonstrate that increased noise reduction leads to a corresponding loss of spatial information and added distortions, which is a key factor in determining speech intelligibility as predicted by MBSTOI, and quality, as predicted by PESQ. Interestingly, there is one fused model that improves noise reduction and also speech intelligibility indexes and quality, namely the "Concat+Fusion" model. While the exact causes remain uncertain and require additional investigation, it can be inferred that the utilization of fusion may yield advantages by considering the balance between noise reduction and potential distortions it may introduce. Consequently, integrating fusion with other feature-sharing techniques has the potential to enhance both noise reduction and speech quality.

IV. CONCLUSION

In this manuscript, we have proposed the utilization of deep fusion layers as an approach to enhance speech in binaural listening scenarios. First, we introduce and establish the concept of the general fused model, elucidating its fundamental notation and describing its characteristics. With this work, we have demonstrated that fusion layers introduce non-

---

[4] https://github.com/speechbrain

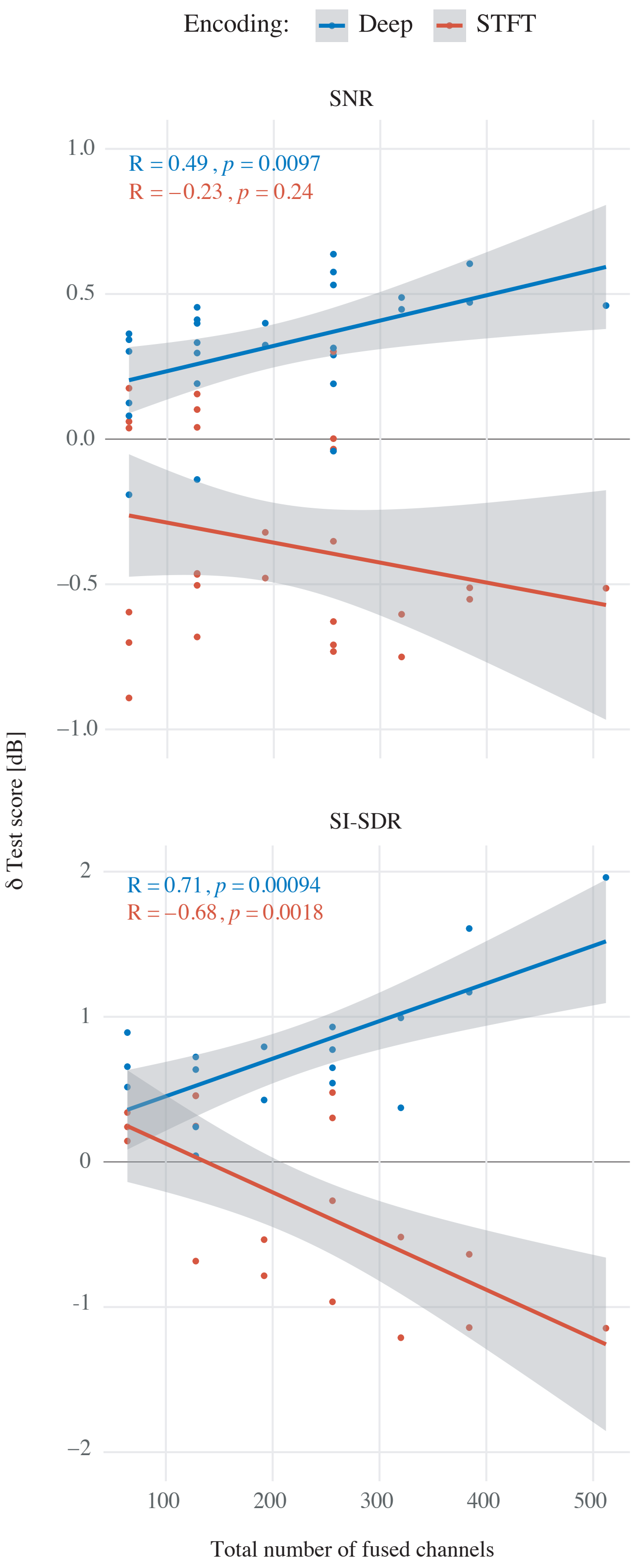

Fig. 10. Regression of the testing error difference between the fused and independent models as a function of the number of the total number of fused channels for each of the investigated encoders. Shaded areas represent a point-wise 95% confidence interval on the fitted values. Correlation analysis is expressed as the adjusted-R and p-value, and it is considered to be significant when $p < 0.05$.

TABLE IV

Absolute testing objective instrumental scores for the different final tested models trained using the SNR loss.

Model	SNR	MBSTOI	STOI	PESQ
Independent [9]	9.21	0.62	0.78	1.64
CDNN [49]	8.10	0.54	0.80	1.37
Front Fusion	9.62	0.58	0.75	1.33
Concat [2]	9.34	0.62	0.78	1.81
Concat + Fusion	9.45	0.63	0.80	2.34
Stitch [23]	9.50	0.58	0.76	1.40
Stitch + Fusion	9.94	0.58	0.76	1.36
Parallel Concat [2]	9.32	0.62	0.76	1.49
Parallel Fusion	9.64	0.59	0.74	1.35
Parallel Cross [31]	9.23	0.63	0.78	1.55
Parallel Cross + Fusion	9.56	0.58	0.75	1.38
Double Fusion	9.71	0.53	0.75	1.39

TABLE V

Absolute testing objective instrumental scores for the different final tested models trained using the SI-SDR loss.

Model	SI-SDR	MBSTOI	STOI	PESQ
Independent [9]	16.04	0.54	0.7	1.21
CDNN [49]	14.05	0.54	0.79	1.26
Front Fusion	17.14	0.51	0.68	1.14
Concat [2]	16.04	0.56	0.71	1.30
Concat + Fusion	16.62	0.56	0.71	1.40
Stitch [23]	16.87	0.50	0.67	1.16
Stitch + Fusion	17.75	0.47	0.65	1.13
Parallel Concat [2]	16.12	0.49	0.64	1.15
Parallel Fusion	18.21	0.53	0.67	1.2
Parallel Cross [31]	15.70	0.53	0.67	1.17
Parallel Cross + Fusion	17.23	0.51	0.68	1.17
Double Fusion	17.62	0.47	0.65	1.12

linearities to the model, improving its ability to accurately represent the distribution of input data. Also, our empirical analysis has shown that fused models are susceptible to input decorrelation, highlighting the importance of considering this aspect. Additionally, we observe that the fusion layers act as a channel through which the gradients through, reducing the variance between learning modules.

Furthermore, we have conducted an analysis of the impact of fusion layers on binaural speech enhancement. Our findings indicate that fused models exhibit promising capabilities in reducing noise compared to independent models. Among various topologies explored, we have discovered that the model incorporating the largest double fusion layers yields the best performance on unseen data.

Importantly, our results have demonstrated that the fusion operation leads to enhanced noise reduction performance when compared to all investigated baseline models. Nonetheless, we recognize the trade-off between the extent of noise reduction and the MBSTOI and PESQ scores, which are important metrics for evaluating noise reduction quality. It is worth noting that the fusion layers we propose not only improve noise reduction but also maintain a constant number of parameters. This aspect becomes particularly relevant when there is a necessity to share large latent representations between the listening sides.

Based on these findings, we firmly believe that our approach holds potential for enhancing future binaural speech processing systems. However, it is crucial to acknowledge that our work assumes instantaneous transmission of information between the listening sides, which may not hold true in real-life applications. Therefore, an important avenue for further investigation is evaluating how latency and the necessary reduction in bitrate for transmitting latent spaces impact the performance of fused models.

Overall, this study may help advance binaural speech processing techniques, and we anticipate that future research will build upon these insights to further refine and optimize the fusion-based approach.

ACKNOWLEDGEMENTS

The authors express their gratitude to Simon Doclo and Marvin Tammen for their invaluable assistance in collecting certain baseline algorithms employed in this study.

REFERENCES

[1] Q. Liu, Y. Xu *et al.*, "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 *et al.*, "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 *et al.*, "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 *et al.*, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[5] W. Kang, Q. Yang *et al.*, "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 *et al.*, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.

[7] L. Deng, G. Hinton *et al.*, "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 *et al.*, *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 *et al.*, "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 *et al.*, "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 *et al.*, "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 *et al.*, "Multitask learning for cross-domain image captioning," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 1047–1061, 2019.

[18] C. Wang, H. Yang *et al.*, "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 *et al.*, "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 *et al.*, "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 *et al.*, "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 *et al.*, "Adaptive cross-stitch graph convolutional networks," in *ACM Multimedia Asia*, 2021, pp. 1–7.

[25] P. Avan, F. Giraudet *et al.*, "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] S. Thaleiser and G. Enzner, "Cue-preserving MMSE filter with bayesian snr marginalization for binaural speech enhancement," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6124–6128.

[29] T. Rohdenburg, V. Hohmann *et al.*, "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.

[30] B. Cornelis, S. Doclo *et al.*, "Theoretical analysis of binaural multi-microphone noise reduction techniques," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 342–355, 2009.

[31] R. Gu, J. Wu *et al.*, "End-to-end multi-channel speech separation," *arXiv preprint arXiv:1905.06286*, 2019.

[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] G. Stecker and F. Gallun, "Binaural hearing, sound localization, and spatial hearing," *Translational perspectives in auditory neuroscience: Normal aspects of hearing*, vol. 383, pp. 383–433, 2012.

[37] S. Graetzer, J. Barker *et al.*, "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.

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

[39] 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.

[40] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria,

2012, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org/

[41] M. Abadi, A. Agarwal *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[42] J. S. Garofolo, L. F. Lamel *et al.*, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993.

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

[44] C. Hummersone, R. Mason *et al.*, "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.

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

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

[47] C. H. Taal, R. C. Hendriks *et al.*, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[48] I.-T. Recommendation, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *Rec. ITU-T P. 862*, 2001.

[49] X. Sun, R. Xia *et al.*, "A deep learning based binaural speech enhancement approach with spatial cues preservation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5766–5770.

[50] X. Leng, J. Chen *et al.*, "On the compromise between noise reduction and speech/noise spatial information preservation in binaural speech enhancement," *The Journal of the Acoustical Society of America*, vol. 149, no. 5, pp. 3151–3162, 2021.

Tom Gajecki obtained his Bachelor of Science degrees in Physics and Electronic Engineering from the University of the Basque Country (UPV), Spain, in 2013 and 2015, respectively. He further pursued a Master of Science in Sound and Music Computing at the University Pompeu Fabra (UPF) in Barcelona, Spain, which he completed in 2017. Since 2018, he has been working towards a Ph.D. in auditory sciences within the Auditory Prosthetic Group at the Hannover Medical School (MHH) and the Cluster of Excellence "Hearing4all." His primary research interests revolve around binaural audio signal processing for auditory devices, deep learning, and music perception in individuals with hearing impairments.

Waldo Nogueira received his Dipl-Ing and Dr.-Ing from the Universitat Politècnica de Catalunya (Barcelona Tech) – UPC and the Leibniz University of Hannover (LUH) in 2003 and 2008, respectively. Subsequently, in 2008, he became part of the R&D labs of Advanced Bionics in Belgium and Germany. In 2011, he was a post-Doc at the Music Technology Group of the Pompeu Fabra University in Barcelona. In 2013, he became W1-Junior professor at the Hannover Medical School (MHH) and the Cluster of Excellence "Hearing4all." Since 2019, he is W2-Professor in Auditory Prostheses at MHH. His primary research interests are audio signal processing for hearing devices, auditory computational models, psychoacoustics, and electrophysiology of the acoustically and electrically stimulated auditory system.