KERNEL MACHINES BEAT DEEP NEURAL NETWORKS ON MASK-BASED SINGLE-CHANNEL SPEECH ENHANCEMENT

Like Hui, Siyuan Ma, Mikhail Belkin

Department of Computer Science and Engineering, The Ohio State University, USA
hui.87@osu.edu, {masi, mbelkin}@cse.ohio-state.edu

ABSTRACT

We apply a fast kernel method for mask-based single-channel speech enhancement. Specifically, our method solves a kernel regression problem associated to a non-smooth kernel function (exponential power kernel) with a highly efficient iterative method (EigenPro). Due to the simplicity of this method, its hyper-parameters such as kernel bandwidth can be automatically and efficiently selected using line search with subsamples of training data. We observe an empirical correlation between the regression loss (mean square error) and regular metrics for speech enhancement. This observation justifies our training target and motivates us to achieve lower regression loss by training separate kernel model per frequency subband. We compare our method with the state-of-the-art deep neural networks on mask-based HINT and TIMIT. Experimental results show that our kernel method consistently outperforms deep neural networks while requiring less training time.

***Index Terms*—** large-scale kernel machines, deep neural networks, speech enhancement, exponential power kernel, automatic hyper-parameter selection

1. INTRODUCTION

The challenging problem of single-channel speech enhancement has received significant attention in research and applications. In recent years the dominant methodology for addressing single-channel speech enhancement has been based on neural networks of different architectures [1, 2]. Deep Neural Networks (DNNs) present an attractive learning paradigm due to their empirical success on a range of problems and efficient optimization.

In this paper, we demonstrate that modern large-scale kernel machines are a powerful alternative to DNNs, capable of matching and surpassing their performance while utilizing less computational resources in training. Specifically, we take the approach to speech enhancement based on the Ideal Binary Mask (IBM) and Ideal Ratio Mask (IRM) methodology. The first application of DNNs to this problem was presented in [3], which used a DNN-SVM (support vector machine) system to solve the classification problem corresponding to estimating the IBM. [4] compared different training targets including IRM. [5] proposed a regression-based approach to estimate speech log power spectrum. Recently, [6] applies recurrent neural networks to similar mask-based tasks and [7] applies convolutional networks to the spectrum-based tasks.

Kernel-based shallow models (which can be interpreted as two-layer neural networks with a fixed first layer), were also proposed to deal with speech tasks. In particular, [8] gave a kernel ridge regression method, which matched DNN on TIMIT. Inspired by this work, [9] applied an efficient one-vs-one kernel ridge regression for speech recognition. [10] developed kernel acoustic models for speech recognition. Notably, these approaches require large computational resources to achieve performance comparable to neural networks.

In our opinion, the computational cost of scaling to larger data has been a major factor limiting the success of these methods. In this work we apply a recently developed highly efficient kernel optimization method EigenPro [11], which allows kernel machines to handle large datasets.

We conduct experiments on standard datasets using mask-based training target. Our results show that, with EigenPro iteration, kernel methods can consistently outperform the performance of DNN in terms of the target mean square error (MSE) as well as the commonly used speech quality evaluation metrics including perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI).

The contributions of our paper are as follows:

1. Using modern kernel algorithms we show performance on mask-based speech enhancement surpassing that of neural networks and requiring less training time.
2. To achieve the best performance, we use exponential power kernel, which, to the best of our knowledge, has not been used for regression or classification tasks.
3. The simplicity of our approach allows us to develop a nearly automatic hyper-parameter selection procedure based on target speech frequency channels.

The rest of the paper is organized as follows. Section 2 introduces our proposed kernel-based speech enhancement system: kernel machines, exponential power kernel, automatic hyper-parameter selection for subband adaptive kernels. Experimental results and time complexity comparisons are discussed in Section 3. Section 4 gives the conclusion.

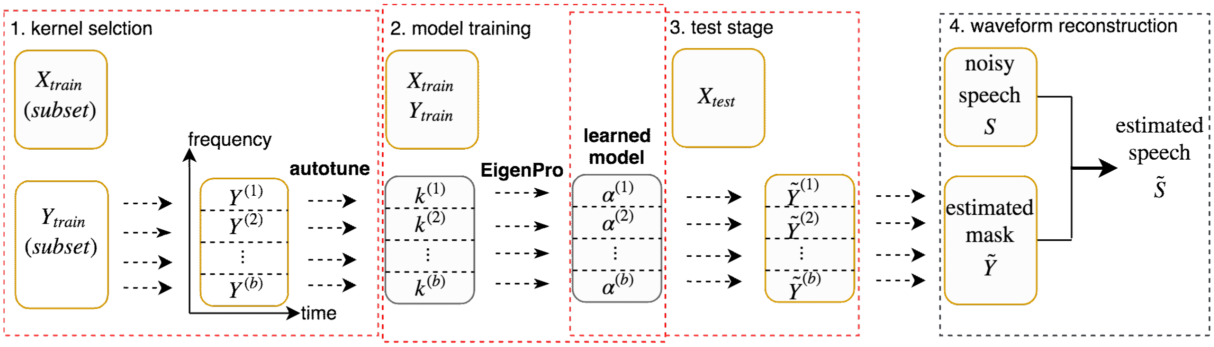


Fig. 1: Kernel-based speech enhancement framework

2. KERNEL-BASED SPEECH ENHANCEMENT

2.1. Kernel Machines

The standard kernel methods for classification/regression denote a function f that minimizes the discrepancy between $f(\boldsymbol{x}_j)$ and y_j, given labeled samples $(\boldsymbol{x}_j, y_j)_{j=1,\ldots,n}$ where $\boldsymbol{x}_j \in \mathbb{R}^d$ is a feature vector and $y_j \in \mathbb{R}$ is its label.

Specifically, the space of f is a Reproducing Kernel Hilbert Space $\mathcal{H}$ associated to a positive-definite kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. We typically seek a function $f^* \in \mathcal{H}$ for the following optimization problem:

$$f^\star = \text{argmin}_{f(\boldsymbol{x}_j)=y_j, j=1,2,\ldots,n} \|f\|_{\mathcal{H}}\,, \tag{1}$$

According to the Representer Theorem [12], f^* has the form

$$f(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j k(\boldsymbol{x}, \boldsymbol{x}_j), \tag{2}$$

To compute f^* is equivalent to solve the linear system,

$$K\boldsymbol{\alpha} = (y_1, \cdots, y_n)^T, \tag{3}$$

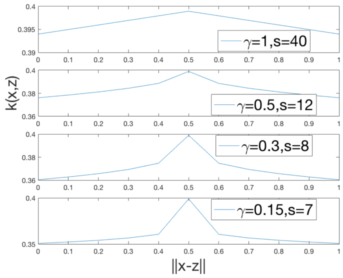
where the kernel matrix K has entry $[K]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{\alpha} \triangleq (\alpha_1, \cdots, \alpha_n)^T$ is the representation of f under basis $\{k(\cdot, \boldsymbol{x}_1), \cdots, k(\cdot, \boldsymbol{x}_n)\}$.

2.2. Exponential Power Kernel

We use an exponential power kernel of the form

$$k_{\gamma,\sigma}(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\frac{\|\boldsymbol{x} - \boldsymbol{z}\|^\gamma}{\sigma}) \tag{4}$$

for our kernel machine, where σ is the kernel bandwidth and γ is often called shape parameter. [13] shows that the exponential power kernel is positive definite, hence a valid reproducing kernel. This kernel also covers a large family of reproducing kernels including Gaussian kernel ($\gamma = 2$) and Laplacian kernel ($\gamma = 1$). We observe that in many noise settings of speech enhancement, the best performance is achieved using this kernel with shape parameter $\gamma \leq 1$, which is highly non-smooth. In the right side figure, we plot this kernel function with parameters that we use in our experiments. We have not seen any application of this kernel (with $\gamma < 1$) in supervised learning literature.



2.3. Automatic Subbands Adaptive Kernels

Algorithm 1 Automatic hyper-parameter selection[1]

```
Input: $\boldsymbol{D}_{\text{train}}, \boldsymbol{D}_{\text{val}}$: training and validation data, $\Gamma$: a set of $\gamma$ for the exponential power kernel, $\sigma_l, \sigma_h$: smallest and largest bandwidth
Output: selected kernel parameters $\gamma_{\text{opt}}, s_{\text{opt}}$ for $D_{\text{train}}$
procedure autotune($\boldsymbol{D}_{\text{train}}, \boldsymbol{D}_{\text{val}}, \Gamma, \sigma_l, \sigma_h$)
    define subprocedure cross-validate($\gamma, \sigma$) as: train one kernel model with $k_{\gamma,\sigma}$ on $\boldsymbol{D}_{\text{train}}$ using EigenPro iteration, return its loss on $D_{\text{val}}$.
    for $\gamma$ in $\Gamma$ do
        $\sigma_\gamma$ = search(cross-validate($\gamma, \cdot$), $\sigma_l, \sigma_h$)
    $\gamma_{\text{opt}}, \sigma_{\text{opt}} \leftarrow \text{argmin}_{\gamma\in\Gamma,\sigma_\gamma}$ cross-validate($\gamma, \sigma_\gamma$)
     return $\gamma_{\text{opt}}, \sigma_{\text{opt}}$
procedure search($f, \sigma_l, \sigma_h$)
    if ($\sigma_h - \sigma_l \leq 2$) then
        return $\sigma_l$
    select $\sigma_{m1}, \sigma_{m2} \in (\sigma_l, \sigma_h)$
    compute $f(\sigma_l), f(\sigma_{m1}), f(\sigma_{m2}), f(\sigma_h)$
    switch min$\{f(\sigma_l), f(\sigma_{m1}), f(\sigma_{m2}), f(\sigma_h)\}$ do
        case $f(\sigma_l)$: return search($f, \sigma_l, \sigma_{m1}$)
        case $f(\sigma_{m1})$: return search($f, \sigma_l, \sigma_{m2}$)
        case $f(\sigma_{m2})$: return search($f, \sigma_{m1}, \sigma_h$)
        case $f(\sigma_h)$: return search($f, \sigma_{m2}, \sigma_h$)
```

[1] We apply memoization technique for computing cross-validate$(\cdot,\cdot)$. We first attempt to set σ_{m1}, σ_{m2} as a value that is already used in (σ_l, σ_h), then we choose them to split (σ_l, σ_h) into three parts as equal as possible.

Table 1: Kernel & DNN on TIMIT: (MSE: lowest is best, STOI and PESQ: highest is best. Best results bolded.)

Noise Type	Metrics	5 dB			0 dB			-5 dB		
		Kernel	DNN	Noisy	Kernel	DNN	Noisy	Kernel	DNN	Noisy
Engine	MSE ($\cdot 10^{-2}$)	**1.10**	1.41	-	**1.34**	1.86	-	**1.17**	1.82	-
	STOI	**0.91**	0.90	0.80	**0.86**	0.85	0.68	**0.80**	0.77	0.57
	PESQ	**2.77**	**2.77**	1.97	**2.51**	2.45	1.66	**2.19**	2.16	1.41
Babble	MSE ($\cdot 10^{-2}$)	**3.34**	3.49	-	**4.18**	4.37	-	**4.94**	5.43	-
	STOI	**0.86**	**0.86**	0.77	**0.77**	**0.77**	0.66	**0.64**	**0.64**	0.55
	PESQ	**2.54**	2.52	2.08	**2.12**	2.10	1.73	**1.70**	1.61	1.42
SSN	MSE ($\cdot 10^{-2}$)	**1.35**	1.53	-	**1.48**	1.67	-	**1.60**	1.76	-
	STOI	**0.88**	**0.88**	0.81	**0.82**	**0.82**	0.69	**0.74**	**0.74**	0.57
	PESQ	**2.68**	2.66	2.05	**2.36**	2.32	1.75	**2.03**	2.00	1.48
Oproom	MSE ($\cdot 10^{-2}$)	**1.44**	1.85	-	**1.34**	1.86	-	**1.17**	1.82	-
	STOI	**0.88**	**0.88**	0.79	**0.84**	0.83	0.70	**0.79**	0.76	0.59
	PESQ	**2.80**	2.79	2.16	**2.50**	2.47	1.78	**2.23**	2.12	1.40
Factory1	MSE ($\cdot 10^{-2}$)	**2.51**	2.53	-	**2.52**	2.55	-	**2.71**	2.77	-
	STOI	**0.86**	**0.86**	0.77	0.78	**0.79**	0.65	**0.68**	**0.68**	0.54
	PESQ	**2.56**	2.51	1.99	2.20	**2.23**	1.62	**1.79**	1.77	1.29

As empirically shown in Section 3.3, we see that models with lower MSE at every frequency channel consistently outperform other models in STOI. This motivates us to achieve lower MSE for all frequency channels by tuning kernel parameters for each of them. In practice, we split the band of frequency channels into several blocks , which we call *subband*s.

We propose a simple kernel-based framework as depicted in Fig. 1 to achieve automatic parameter selection and fast training for each subband. For i-th subband, the framework learns one model $f^{(i)}$ related to an exponential power kernel $k^{(i)}$ with parameters automatically tuned for this subband,

$$f^{(i)}(\boldsymbol{x}) = \sum_{j=1}^{n} \alpha_j^{(i)} k^{(i)}(\boldsymbol{x}, \boldsymbol{x}_j). \tag{5}$$

Our framework starts by splitting the training targets Y_{train} into subband targets $Y^{(1)}, \cdots, Y^{(b)}$. For training data related to the i-th subband $(X_{\text{train}}, Y^{(i)})$, we perform fast and automatic kernel parameter selection using *autotune* (Algorithm 1) on its subsamples, which selects one exponential power kernel $k^{(i)}$ for this subband. We then train a kernel model on $(X_{\text{train}}, Y^{(i)})$ with kernel $k^{(i)}$ using EigenPro iteration proposed in [11]. It learns an approximate solution $\alpha^{(i)}$ (or $f^{(i)}$) for the optimization problem (1). Our final kernel machine is then formed by $\{f^{(1)}, \cdots, f^{(b)}\}$.

For any unseen data $\boldsymbol{x}$, our kernel machine first computes estimated mask $f^{(i)}(\boldsymbol{x})$ for each subband. Then it combines the results of $\{f^{(1)}(\boldsymbol{x}), \cdots, f^{(b)}(\boldsymbol{x})\}$ to obtain the estimated mask for all frequency channels. Applying this mask to the noisy speech produces the estimated clean speech.

3. EXPERIMENTAL RESULTS

We use kernel machines with 4 subbands (block of frequencies) for speech enhancement. For fair comparison, we train both kernel machines and DNNs from scratch using the same features and targets. We halt the training for any model when error on validation set stops decreasing. Experiments are run on a server with 128GB main memory, two Intel Xeon(R) E5-2620 CPUs, and one GTX Titan Xp (Pascal) GPU.

3.1. Regression Task

We compare kernel machines and DNNs on a speech enhancement task described in [4] which is based on TIMIT corpus [14] and uses real-valued masks (IRM). We follow the description in [4] for data preprocessing and DNN construction/training. We consider five background noises: SSN, babble, a factory noise (factory1), a destroyer engine room (engine), and an operation room noise (oproom). Every noise is mixed to speech at $-5, 0, 5$dB Signal-Noise-Ratio (SNR).

Table 1 reports the MSE, STOI, and PESQ on test set for kernel machines and DNNs. We also present the STOI and PESQ of the noisy speech without enhancement. For all noise settings, we see that kernel machines consistently produce better MSE, in many cases significantly lower than that from DNNs, which is also the training objective for both models. We also see that STOI and PESQ of kernel machines are consistently better than or comparable to that from DNNs with only one exception (Factory1 0dB).

3.2. Classification Task

We train kernel machines and DNNs for a speech enhancement task in [15] which is based on HINT dataset and adopts binary masks (IBM) as targets. We follow the same procedure described in [15] to preprocess the data and construct/train DNNs. Specifically, we use two background noises, SSN and multi-talker babble. SSN is mixed to speech at -2, -5, -8dB SNR, and babble is mixed to speech at $0, -2, -5$dB SNR. As our kernel machine is designed for regression task, we use a threshold 0.5 to map its real-value prediction to binary target $\{0, 1\}$.

Table 2: Kernel & DNN on HINT

Metrics	Model	Babble			SSN		
		0dB	-2dB	-5dB	-2dB	-5dB	-8dB
Acc	DNN	0.90	0.91	0.90	0.91	**0.91**	**0.92**
	Kernel	**0.92**	**0.92**	**0.91**	**0.92**	0.90	0.89
STOI	DNN	0.83	0.80	0.76	0.79	**0.76**	**0.74**
	Kernel	**0.86**	**0.83**	**0.78**	**0.81**	0.75	0.71

In Table 2, we compare the classification accuracy (Acc) and STOI of kernel machine and DNNs under different noise settings. We see that our kernel machines outperform DNNs on noise settings with babble and perform worse than DNN on noise settings with SSN. In all, the proposed kernel machines match the performance of DNNs on this classification task.

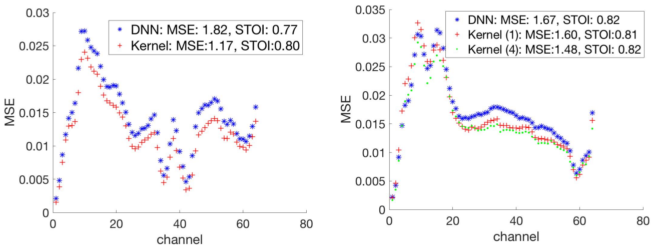
3.3. Single Kernel and Subband Adaptive Kernels

We start by analyzing the performance of kernel machines that use a single kernel for all frequency channels on the regression task in Section 3.1. The training of such kernel machine (1 subband) is significantly faster than that of our default kernel machine (4 subbands). Remarkably, its performance is also quite competitive. It consistently outperforms DNNs in MSE in all noise settings. In 8 out of 15 noise settings, it produces STOI the same as that from the kernel machine with 4 subbands (it also produces nearly same PESQ).

Table 3: Comparison of kernel machines with 1 subband and 4 subbands

Noise setting	Metrics	Kernel (1 subband)	Kernel (4 subbands)	DNN
SSN 0dB	MSE	1.60	**1.48**	1.67
	STOI	0.81	**0.82**	**0.82**
	PESQ	2.35	**2.36**	2.32
SSN -5dB	MSE	1.67	**1.60**	1.76
	STOI	0.73	**0.74**	**0.74**
	PESQ	2.01	**2.03**	2.00
Factory1 -5dB	MSE	2.76	**2.71**	2.77
	STOI	0.67	**0.68**	**0.68**
	PESQ	1.78	**1.79**	1.77

However, in other noise settings, kernel (1 subband) has smaller training loss (MSE) than DNNs, but no better STOI (we show three cases in Table 3) [16, 17]. To improve desired metrics (STOI/PESQ), we first compare the MSE of every frequency channel of DNNs and kernel machines.



(a) Engine -5dB

(b) SSN 0dB

Fig. 2: MSE along per frequency channel

As shown in Fig. 2a, for cases that kernels have much smaller overall MSE and smaller MSE on each frequency channel, kernels also achieve better STOI. For cases like SSN 0dB, as shown in Fig. 2b, even though single kernel (1 subband) has smaller overall MSE, its STOI is not as good as DNNs. Multiple kernels (4 subbands) decrease MSE further and also achieve better STOI. This shows that having smaller MSE along all frequency channels leads to better STOI. This reveals a correlation between MSE and STOI/PESQ associated with frequency channels.

3.4. Time Complexity

Table 4: Running time/epochs of Kernel & DNN

Dataset	Time (minutes)			Epochs	
	Kernel		DNN	Kernel	DNN
	1 subband	4 subbands			
HINT	0.8	3.2	6.6	10	50
TIMIT	18	65	124	5	93

In Table 4, we compare the training time of DNNs and kernel machine on both HINT and TIMIT. Note that the training of kernel machines in all experiments typically completes in no more than 10 epochs, significantly less than the number of epochs required for DNNs. Furthermore, the training time of kernel machines is also less than that of DNNs. Notably, training kernel machine with 1 subband takes much less time than DNNs.

4. CONCLUSION

In this paper, we have shown that kernel machines using exponential power kernels show strong performance on speech enhancement problems. Notably, our method needs no parameter tuning for optimization and employs nearly automatic tuning for kernel hyper-parameter selection. Moreover, we show that the training time and computational requirement of our method are comparable or less than those needed to train neural networks. We expect that this highly efficient kernel method will be useful for other problems in speech and signal processing.

5. REFERENCES

[1] DeLiang Wang and Jitong Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726., 2018.

[2] Zixing Zhang, Jürgen Geiger, Jouni Pohjalainen, Amr El-Desoky Mousa, Wenyu Jin, and Björn Schuller, "Deep learning for environmentally robust speech recognition: An overview of recent developments," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, pp. 49, 2018.

[3] Yuxuan Wang and DeLiang Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, 2013.

[4] Yuxuan Wang, Arun Narayanan, and DeLiang Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[5] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[6] Zhong-Qiu Wang and DeLiang Wang, "Recurrent deep stacking networks for supervised speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 71–75.

[7] Ashutosh Pandey and Deliang Wang, "A new framework for supervised speech enhancement in the time domain," *Proc. Interspeech 2018*, pp. 1136–1140, 2018.

[8] Po-Sen Huang, Haim Avron, Tara N Sainath, Vikas Sindhwani, and Bhuvana Ramabhadran, "Kernel methods match deep neural networks on timit.," in *Acoustics, Speech and Signal Processing (ICASSP), 2014*, 2014, pp. 205–209.

[9] Jie Chen, Lingfei Wu, Kartik Audhkhasi, Brian Kingsbury, and Bhuvana Ramabhadrari, "Efficient one-vs-one kernel ridge regression for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2454–2458.

[10] Zhiyun Lu, Dong Quo, Alireza Bagheri Garakani, Kuan Liu, Avner May, Aurélien Bellet, Linxi Fan, Michael Collins, Brian Kingsbury, Michael Picheny, and Fei Sha, "A comparison between deep neural nets and kernel acoustic models for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5070–5074.

[11] Siyuan Ma and Mikhail Belkin, "Learning kernels that adapt to gpu," *arXiv preprint arXiv:1806.06144*, 2018.

[12] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al., *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.

[13] BG Giraud and R Peschanski, "On positive functions with positive fourier transforms," *Acta Physica Polonica B*, vol. 37, pp. 331, 2006.

[14] John S Garofolo, Lori F Lamel, William M Fisher, Jonathon G Fiscus, and David S Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom," *NIST speech disc*, vol. 1-1.1, 1993.

[15] Eric W Healy, Sarah E Yoho, Yuxuan Wang, and DeLiang Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners," *The Journal of the Acoustical Society of America*, vol. 134, no. 4, pp. 3029–3038, 2013.

[16] Leo Lightburn and Mike Brookes, "A weighted stoi intelligibility metric based on mutual information," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5365–5369.

[17] Hui Zhang, Xueliang Zhang, and Guanglai Gao, "Training supervised speech separation system to improve stoi and pesq directly," in *Acoustics, Speech and Signal Processing (ICASSP), 2018*, 2018, pp. 5374–5378.