

USDnet: Unsupervised Speech Dereverberation via Neural Forward Filtering

Zhong-Qiu Wang 

Abstract—In reverberant conditions with a single speaker, each far-field microphone records a reverberant version of the same speaker signal at a different location. In over-determined conditions, where there are multiple microphones but only one speaker, each recorded mixture signal can be leveraged as a constraint to narrow down the solutions to target anechoic speech and thereby reduce reverberation. Equipped with this insight, we propose USDnet, a novel deep neural network (DNN) approach for unsupervised speech dereverberation (USD). At each training step, we first feed an input mixture to USDnet to produce an estimate for target speech, and then linearly filter the DNN estimate to approximate the multi-microphone mixture so that the constraint can be satisfied at each microphone, thereby regularizing the DNN estimate to approximate target anechoic speech. The linear filter can be estimated based on the mixture and DNN estimate via neural forward filtering algorithms such as forward convolutive prediction. We show that this novel methodology can promote unsupervised dereverberation of single-source reverberant speech.

Index Terms—Unsupervised neural speech dereverberation (USD).

I. INTRODUCTION

ROOM reverberation, caused by signal reflections inside reverberant enclosures, is very detrimental to machine perception in tasks such as automatic speech recognition (ASR) and speaker recognition [1]. It also dramatically degrades human hearing, leading to much lower speech intelligibility and quality. Rereverberation results from a linear convolution between a room impulse response and the source signal, and speech dereverberation is known as a classic blind deconvolution problem [1], [2], [3], which is ill-posed in nature and difficult to be solved as only the convolved signal is observed but the filter and source are both unknown and need to be estimated. There are infinite solutions to the filter and source pair, where, in each solution, their linear-convolution result can well-approximate the observed reverberant speech. For example, even if the source estimate is a reverberant version of the source, there still exists a filter such that convolving the source estimate with the filter can roughly approximate the mixture. The key to successful speech dereverberation, we believe, is to cleverly narrow down the infinite solutions to the source and filter so that the models

or algorithms can have a clear idea about what target signals to predict.

A popular approach for speech dereverberation is based on inverse filtering [1], [4], [5], [6], mainly leveraging signal processing principles. Representative algorithms include weighted prediction error (WPE) [7], [8], [9], [10], [11], where a linear filter is computed to filter past observations with a positive prediction delay to estimate late reverberation to remove. The rationale is that the speech signals beyond the prediction delay in past frames is little linearly-correlated with the target anechoic signal at the current frame. WPE leverages this physical principle to narrow down the solutions to target speech in an unsupervised way. It has gradually become the most popular and successful dereverberation algorithm [6], often producing consistent improvement for modern ASR systems. Subsequent studies along this line of research combine WPE with beamformers in a sequential [6] or in a jointly-optimal [12], [13] way. It should be noted that the dereverberation results of WPE typically contain some early reflections due to its positive prediction delay. This is however often acceptable in many applications. For example, early reflections are found beneficial for speech intelligibility [14].

Another popular approach is based on supervised learning, where anechoic and reverberant speech pairs are synthesized via room simulation and a supervised DNN is trained to predict the anechoic speech from reverberant speech [15], [16], [17], [18], [19], [20], [21], [22], [23], [24]. This approach usually requires a large amount of simulated training data to avoid potential issues related to data scarcity [25]. The target anechoic speech serves as the training label, and the DNN can learn to find the true solution to target speech through supervised learning. Recent studies along this line of research leverage target speech estimated by supervised DNNs to improve signal processing based dereverberation [26], [27], [28], [29], [30], [31], and leverage generative modeling [32] to improve dereverberation. This approach is often effective as the DNN has strong capabilities at modeling speech patterns that can be very informative for dereverberation.

In this context, we propose to investigate unsupervised neural speech dereverberation. Different from signal processing based approaches, which are unsupervised in nature and usually not designed to learn speech patterns from massive training data, we leverage unsupervised deep learning to model speech patterns. Different from supervised learning based approaches, unsupervised neural dereverberation models can be trained directly on real-recorded reverberant speech, avoiding using simulated

Manuscript received 1 February 2024; revised 13 July 2024; accepted 4 August 2024. Date of publication 16 August 2024; date of current version 30 August 2024. The associate editor coordinating the review of this article and approving it for publication was Dr. Keisuke Kinoshita.

The author is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: wang.zhongqiu41@gmail.com).

Digital Object Identifier 10.1109/TASLP.2024.3445120

training data which could incur generalization issues [33], [34], [35], [36], [37].

The key to unsupervised neural speech dereverberation, we believe, is to cleverly narrow down the infinite solutions to the source and filter so that DNNs can receive a clear *supervision* on what target signals to predict, or a clear *regularization* on what estimates produced by the DNN are not good and should be avoided. This way, the strong modeling capability of DNNs on speech signals can be sufficiently utilized for dereverberation, rather than being wasted due to obscure supervision. Motivated by the UNSSOR algorithm [38], which deals with unsupervised neural speaker separation and does not perform any dereverberation, we assume the availability of over-determined training mixtures (i.e., multi-channel reverberant speech in single-speaker cases), and leverage each mixture signal, during training, as a *mixture constraint* to narrow down the solutions to the source and filter. This way, the DNN can (a) receive a clear *supervision* about what the target speech is by checking whether its prediction satisfies the mixture constraints during training; and (b) be trained directly on a set of collected reverberant signals to realize dereverberation. Our contributions are summarized as follows:

- We, for the first time, formulate unsupervised neural speech dereverberation as a blind deconvolution problem, which requires estimating both the source and the filter.
- We propose USDnet, which estimates target (source) speech by modeling speech patterns via unsupervised deep learning. Based on the source estimate and input mixture, filter estimation is formulated as a linear regression problem, which can be solved via a neural forward filtering algorithm named forward convolutive prediction (FCP) [31] in multiple ways.
- Following the blind deconvolution problem, we propose novel mixture-constraint loss functions. They leverage multi-microphone mixtures as constraints to regularize the DNN estimate. Together with several other regularizations (which we will introduce), they are found effective at leading the DNN estimate to approximate target speech (and realize dereverberation).

Evaluation results on two speech dereverberation tasks show that USDnet can effectively reduce reverberation. A sound demo page is provided in the link below.¹

II. RELATED WORK

Unsupervised neural speech separation, mainly being studied for speech de-noising, separating mixture of multiple anechoic speakers, and separating multi-speaker mixtures to reverberant speaker images without any dereverberation, has attracted consistent research interests in recent years. However, unsupervised neural dereverberation has been largely under-explored, although dereverberation is an important task in speech separation and enhancement.

MixIT [36], [39], [40], a representative algorithm, first synthesizes mixtures of mixtures (MoM), each created by mixing existing recorded mixtures, and then trains a DNN to separate

each MoM to a number of sources such that the separated sources can be partitioned into groups and, within each group, the separated sources can sum up to one of the mixtures used for synthesizing the MoM. Although MixIT is shown capable of separating mixtures of independent reverberant sources, it cannot reduce the reverberation of each source as the reflected signals of each source are dependent with each other. Noisy-target training [41], [42] is an unsupervised approach for speech de-noising. Similarly to MixIT, it synthetically mixes noisy speech with noise, and trains DNNs to predict the noisy speech based on the synthetic mixtures. This approach is found capable of reducing noise as speech and noise are independent. However, whether it would be effective at dereverberation is unknown, especially when dealing with reflection signals that are highly dependent with each other.

Another approach requires source prior models, which can be built based on clean anechoic speech [43], [44], [45], [46], or anechoic and reverberant speech with some annotations [47]. The prior models can, to some extent, narrow down the solutions to target speech and provide a supervision on whether DNN estimates are desired or not during training or inference.

There are approaches leveraging pseudo-labels produced by conventional algorithms such as spatial clustering and independent vector analysis (IVA) for training unsupervised neural source separation models [48], [49], [50], [51], leveraging pseudo-labels provided by DNN-supported WPE for training unsupervised dereverberation models [52], or training DNNs to optimize the objective functions of conventional separation algorithms [53], [54], [55]. Their systems heavily rely on conventional algorithms and their performance is often limited by conventional algorithms.

There are studies realizing unsupervised speech de-noising via positive and unlabeled learning [56], where noisy speech and noise signals are assumed available so that a binary classifier can be trained to determine whether each small patch of T-F units belongs to noisy speech that can be maintained or noise that should be removed. However, it is unclear whether this approach can be modified for dereverberation.

The proposed USDnet is motivated by UNSSOR [38], which formulates unsupervised neural speaker separation as a blind deconvolution problem, but UNSSOR is designed to maintain the reverberation of each speaker rather than reducing it. As we will show in this paper, unsupervised dereverberation has unique challenges and difficulties very different from the ones in unsupervised separation of mixed reverberant speakers, and therefore calls for innovative solutions.

During training, USDnet utilizes recorded mixtures as mixture constraints to narrow down target speech and realizes unsupervised dereverberation by solely leveraging recorded mixtures. It exploits implicit regularization afforded by multi-microphone signals, and does not require using synthetic data (e.g., MoM, and paired clean and corrupted speech), source prior models, or pseudo-labels provided by conventional algorithms. These differences distinguish USDnet from existing unsupervised neural methods, and, meanwhile, indicate that USDnet can complement many existing algorithms.

¹ See https://zqwang7.github.io/demos/USDnet_demo/index.html

III. PROBLEM FORMULATION

This section describes the hypothesized physical models, formulates speech dereverberation as a blind deconvolution problem, highlights the benefits of over-determined conditions, and overviews the proposed USDnet.

A. Physical Models

For a P -microphone mixture with a single speaker in noisy-reverberant conditions, the physical model can be formulated in the short-time Fourier transform (STFT) domain as follows.

At a reference microphone $q \in \{1, \dots, P\}$, we have

$$\begin{aligned} Y_q(t, f) &= X_q(t, f) + \varepsilon_q(t, f) \\ &= S_q(t, f) + R_q(t, f) + \varepsilon_q(t, f) \\ &= S_q(t, f) + \mathbf{g}_q(f)^H \tilde{\mathbf{S}}_q(t, f) + \varepsilon'_q(t, f), \end{aligned} \quad (1)$$

where t indexes T frames and f indexes F frequencies. In the first row, $Y_q(t, f)$, $X_q(t, f)$, and $\varepsilon_q(t, f)$ respectively denote the STFT coefficients of the noisy-reverberant mixture, reverberant speaker image, and a weak noise signal captured at time t , frequency f and microphone q . In the rest of this paper, when dropping indices t , f and q , we refer to the corresponding spectrograms. In the second row, we decompose X_q into direct-path signal S_q and reverberation R_q (i.e., $X_q = S_q + R_q$). In the third row, following narrowband approximation [5], [57] we model reverberation R_q as a linear convolution between a linear filter \mathbf{g}_q and the source S_q , i.e., $R_q(t, f) \approx \mathbf{g}_q(f)^H \tilde{\mathbf{S}}_q(t, f)$, where $\tilde{\mathbf{S}}_q(t, f) = [S_q(t - K + 1, f), \dots, S_q(t - \Delta, f)]^T \in \mathbb{C}^{K-\Delta}$ stacks a window of T-F units, Δ is a non-negative prediction delay, $\mathbf{g}_q(f) \in \mathbb{C}^{K-\Delta}$ is a relative transfer function relating direct-path $S_q(\cdot, f)$ to reverberation $R_q(\cdot, f)$, $(\cdot)^H$ computes Hermitian transpose, and ε'_q absorbs the modeling error.

At any non-reference microphone $p \in \{1, \dots, P\}$, where $p \neq q$, we formulate the physical model as

$$\begin{aligned} Y_p(t, f) &= X_p(t, f) + \varepsilon_p(t, f) \\ &= \mathbf{h}_p(f)^H \bar{\mathbf{S}}_q(t, f) + \varepsilon'_p(t, f), \end{aligned} \quad (2)$$

where $\bar{\mathbf{S}}_q(t, f) = [S_q(t - I + 1, f), \dots, S_q(t + J, f)]^T \in \mathbb{C}^{I+J}$ stacks a window of T-F units, $\mathbf{h}_p(f) \in \mathbb{C}^{I+J}$ is the relative transfer function relating S_q at the reference microphone q to the reverberant image at non-reference microphone p (i.e., X_p), and ε'_p absorbs the modeling error.

B. Formulating Dereverberation as Blind Deconvolution

We aim at reducing the reverberation of the target speaker in an unsupervised way. With the physical models in (1) and (2), unsupervised dereverberation can be realized by solving, e.g., the minimization problem below:

$$\begin{aligned} \argmin_{\mathbf{g}(\cdot), \mathbf{h}(\cdot), S_q(\cdot, \cdot)} & \left(\sum_{t, f} \left| Y_q(t, f) - S_q(t, f) - \mathbf{g}_q(f)^H \tilde{\mathbf{S}}_q(t, f) \right|^2 \right. \\ & \left. + \sum_{p=1, p \neq q}^P \sum_{t, f} \left| Y_p(t, f) - \mathbf{h}_p(f)^H \bar{\mathbf{S}}_q(t, f) \right|^2 \right), \end{aligned} \quad (3)$$

which finds source and filters that are most consistent with the physical models in (1) and (2) (i.e., most satisfied with the mixture constraints). This is a blind deconvolution problem [2], which is non-convex in nature and is not solvable if not assuming any prior knowledge about the filters or the source, because all of them are unknown and need to be estimated.

C. Benefits of Over-Determined Conditions

We point out that the more microphones we have, the more likely we could solve the problem in (3). To provide an intuitive interpretation, following UNSSOR [38] we can view the physical models in (1) and (2) as a system of linear equations with many unknowns, where each linear equation can be considered as a constraint to some of the unknowns. When certain conditions (which we will describe next) are satisfied, the equations can significantly outnumber unknowns, making it more likely to successfully estimate the unknowns.

In detail, suppose that the modeling error ε' is negligible. The linear system in (1) and (2) has in total $P \times T \times F$ linear equations, as we have a mixture observation for each $Y_p(t, f)$. Meanwhile, it has $T \times F + (K - \Delta) \times F + (I + J) \times (P - 1) \times F$ unknowns, where the first term, $T \times F$, is because we have an unknown for each $S_q(t, f)$, the second term, $(K - \Delta) \times F$, is because we have $K - \Delta$ unknowns for $\mathbf{g}_q(f) \in \mathbb{C}^{K-\Delta}$ at the reference microphone q and each frequency f , and the third term, $(I + J) \times (P - 1) \times F$, is because we have $I + J$ unknowns for each $\mathbf{h}_p(f) \in \mathbb{C}^{I+J}$ at each non-reference microphone p and each frequency f .

In many cases, the linear filters are usually shorter than one second (meaning that $K - \Delta$ and $I + J$ can be assumed small). If the recording is reasonably long (i.e., $T \gg K - \Delta$ and $T \gg I + J$) and when there are multiple microphones (i.e., $P > 1$), the number of equations can be much larger than the number of unknowns. That is, $P \times T \times F \gg T \times F + (K - \Delta) \times F + (I + J) \times (P - 1) \times F$ when $P > 1$. In other words, we consider each mixture observation as a constraint which the estimate of target speech has to satisfy. When there are many more constraints than unknowns, we can more likely narrow down the solutions to target speech.

On the other hand, if we only have one microphone (i.e., $P = 1$), the number of equations, $T \times F$, is smaller than the number of unknowns, $T \times F + (K - \Delta) \times F$. In this case, it would be more difficult to pinpoint the true solution to target speech using the approach in (3) as the number of constraints is insufficient.

Since our study assumes that there is only one target speaker, we refer to the case when multiple microphones are available as *over-determined condition*, following the conventions in source separation [58].

D. Overview of USDnet

With the problem formulation in Section III-B and understandings in Section III-C, in the next section, following UNSSOR [38] we propose USDnet, which tackles this problem by modeling speech patterns (i.e., source prior) via unsupervised deep learning. The high-level idea is to leverage unsupervised deep learning to first produce a source estimate. With the source

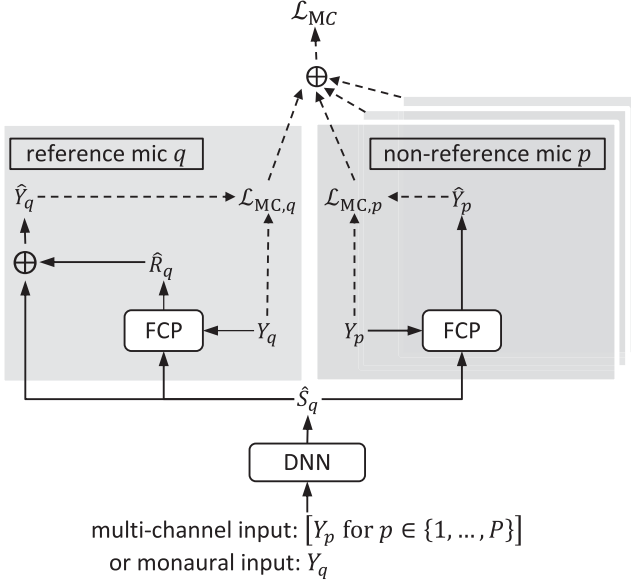


Fig. 1. Illustration of vanilla USDnet (described in first paragraph of Section IV). Several variants are described in Section IV.

estimated in (3), filter estimation then becomes a simple linear regression problem, which has a closed-form solution. Finally, an objective similar to (3), ideally computed based on multiple microphones, can be designed to regularize the source estimate to have it approximate S_q .

IV. USDNET

Fig. 1 illustrates the proposed USDnet in its vanilla case. It takes as input the mixtures at all the microphones or just at the reference microphone q , and produces an estimate \hat{S}_q at the reference microphone q . To encourage \hat{S}_q to approximate S_q , we regularize it during training by linearly filtering it via FCP [31] so that the filtered estimates can approximate the mixtures at different microphones. This section describes the DNN setup, loss functions, FCP filtering, an extension to monaural unsupervised dereverberation, and the rationale of various design choices of USDnet.

A. DNN Setup

The DNN estimate \hat{S}_q can be produced via complex spectral mapping [59], [60], where the real and imaginary (RI) parts of input mixtures are stacked as input features for the DNN to predict the RI parts of \hat{S}_q . Alternatively, it can be estimated via complex ratio masking [61], [62], where the RI parts of a complex mask \hat{M}_q is predicted by the DNN and \hat{S}_q is obtained by $\hat{S}_q = \hat{M}_q \odot Y_q$, with \odot denoting point-wise multiplication. In supervised speech separation [63], both masking and mapping are popular, and mapping usually produces better separation than masking [64]. In unsupervised dereverberation, however, we find that, for non-trivial reasons, training USDnet with masking is clearly better. We will discuss the reasons in Section IV-H. Other DNN configurations such as network architectures are provided later in Section V-C.

B. Mixture-Constraint Loss

Following UNSSOR [38], we design a mixture-constraint (MC) loss \mathcal{L}_{MC} to regularize the DNN estimate \hat{S}_q so that it can approximate S_q :

$$\mathcal{L}_{MC} = \mathcal{L}_{MC,q} + \alpha \times \sum_{p=1, p \neq q}^P \mathcal{L}_{MC,p}, \quad (4)$$

with $\alpha \in \mathbb{R}_{>0}$ denoting a microphone weighting term for non-reference microphones, and $\mathcal{L}_{MC,q}$ and $\mathcal{L}_{MC,p}$, defined by closely following the two terms in (3), as

$$\begin{aligned} \mathcal{L}_{MC,q} &= \sum_{t,f} \mathcal{F} \left(Y_q(t, f), \hat{Y}_q(t, f) \right) \\ &= \sum_{t,f} \mathcal{F} \left(Y_q(t, f), \hat{S}_q(t, f) + \hat{R}_q(t, f) \right) \\ &= \sum_{t,f} \mathcal{F} \left(Y_q(t, f), \hat{S}_q(t, f) + \hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f) \right), \end{aligned} \quad (5)$$

$$\begin{aligned} \mathcal{L}_{MC,p} &= \sum_{t,f} \mathcal{F} \left(Y_p(t, f), \hat{Y}_p(t, f) \right) \\ &= \sum_{t,f} \mathcal{F} \left(Y_p(t, f), \hat{\mathbf{h}}_p(f)^H \tilde{\mathbf{S}}_q(t, f) \right), \end{aligned} \quad (6)$$

where $\tilde{\mathbf{S}}_q(t, f) = [\hat{S}_q(t - K + 1, f), \dots, \hat{S}_q(t - \Delta, f)]^T \in \mathbb{C}^{K-\Delta}$ stacks a window of T-F units in \hat{S}_q with a non-negative prediction delay Δ , $\tilde{\mathbf{S}}_q(t, f) = [\hat{S}_q(t - I + 1, f), \dots, \hat{S}_q(t + J, f)]^T \in \mathbb{C}^{I+J}$, and $\hat{\mathbf{g}}_q(f) \in \mathbb{C}^{K-\Delta}$ and $\hat{\mathbf{h}}_p(f) \in \mathbb{C}^{I+J}$ respectively denote the corresponding estimated linear filters (to be described later). Notice that the filter taps, K , Δ , I and J , are unknown in unsupervised setups. They are hyper-parameters to tune. Their values are shared by all the training utterances in this study.

Following [38], [65], $\mathcal{F}(\cdot, \cdot)$ computes a loss on the estimated RI components and their magnitude:

$$\begin{aligned} \mathcal{F} \left(Y_a(t, f), \hat{Y}_a(t, f) \right) &= \frac{1}{\sum_{t',f'} |Y_a(t', f')|} \left(\left| \mathcal{R}(Y_a(t, f)) - \mathcal{R}(\hat{Y}_a(t, f)) \right| \right. \\ &\quad + \left| \mathcal{I}(Y_a(t, f)) - \mathcal{I}(\hat{Y}_a(t, f)) \right| \\ &\quad \left. + \left| |Y_a(t, f)| - |\hat{Y}_a(t, f)| \right| \right), \end{aligned} \quad (7)$$

where a indexes all the microphones, $|\cdot|$ computes magnitude, $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ respectively extract RI components, and the normalization term balances the losses at different microphones and across training mixtures.

To compute \mathcal{L}_{MC} and enforce the above-mentioned regularizations to \hat{S}_q , we need to first estimate the filters, $\hat{\mathbf{g}}_q(f)$ and $\hat{\mathbf{h}}_p(f)$. We describe the method in the next subsection.

In our experiments, we observe that minimizing \mathcal{L}_{MC} can promote unsupervised dereverberation. A plot of its loss curve is described later in Section VI-B and shown later in Fig. 2.

Notice that we want \hat{S}_q to approximate direct-path S_q , and $\hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f)$ to approximate reverberation $R_q(t, f)$. To achieve this, in (5) we constrain their summation to approximate the mixture (i.e., $\hat{Y}_q(t, f) = \hat{S}_q(t, f) + \hat{R}_q(t, f) = \hat{S}_q(t, f) + \hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f)$), and in (6) we constrain \hat{S}_q such that it can be linearly filtered to approximate the mixture at another microphone (i.e., $\hat{Y}_p(t, f) = \hat{\mathbf{h}}_p(f)^H \tilde{\mathbf{S}}_q(t, f)$). However, naively using (5) and (6) alone is not sufficiently effective at leading \hat{S}_q to approximate S_q , as the loss is defined on the reconstructed mixture rather than on \hat{S}_q itself. We will propose more regularizations later in Sections IV-E, IV-G, IV-H and IV-I so that even if the loss is defined on the reconstructed mixtures, the resulting \hat{S}_q would still approximate S_q .

C. FCP for Filter Estimation

Following UNSSOR [38], we compute the relative transfer functions $\hat{\mathbf{g}}_q(f)$ and $\hat{\mathbf{h}}_p(f)$ via forward convolutive prediction (FCP) [31], [66], a neural forward filtering algorithm that linearly filters the DNN estimate \hat{S}_q to approximate the mixture:

$$\hat{\mathbf{g}}_q(f) = \underset{\mathbf{g}_q(f)}{\operatorname{argmin}} \sum_t \frac{|Y_q(t, f) - \hat{S}_q(t, f) - \mathbf{g}_q(f)^H \tilde{\mathbf{S}}_q(t, f)|^2}{\hat{\lambda}_q(t, f)}, \quad (8)$$

$$\hat{\mathbf{h}}_p(f) = \underset{\mathbf{h}_p(f)}{\operatorname{argmin}} \sum_t \frac{|Y_p(t, f) - \mathbf{h}_p(f)^H \tilde{\mathbf{S}}_q(t, f)|^2}{\hat{\lambda}_p(t, f)}, \quad (9)$$

where $\hat{\lambda}_a$, with a indexing all the microphones, is a weighting term balancing the importance of various T-F units. Following [31], [38], [66], we define it as

$$\hat{\lambda}_a(t, f) = \frac{1}{P} \sum_{p=1}^P |Y_p(t, f)|^2 + \xi \times \max \left(\frac{1}{P} \sum_{p=1}^P |Y_p|^2 \right), \quad (10)$$

where ξ (tuned to 10^{-4} in this paper) floors the weighting term to avoid placing too much weights on low-energy T-F units, and $\max(\cdot)$ extracts the maximum value of a spectrogram.

Both (8) and (9) are weighted linear regression problems, where closed-form solutions can be readily computed. For (9), the closed-form solution is

$$\hat{\mathbf{h}}_p(f) = \left(\sum_t \frac{\tilde{\mathbf{S}}_q(t, f) \tilde{\mathbf{S}}_q(t, f)^H}{\hat{\lambda}_p(t, f)} \right)^{-1} \sum_t \frac{\tilde{\mathbf{S}}_q(t, f) (Y_p(t, f))^*}{\hat{\lambda}_p(t, f)},$$

where $(\cdot)^*$ computes complex conjugate. For (8), it is

$$\hat{\mathbf{g}}_q(f) = \left(\sum_t \frac{\tilde{\mathbf{S}}_q(t, f) \tilde{\mathbf{S}}_q(t, f)^H}{\hat{\lambda}_q(t, f)} \right)^{-1}$$

$$\sum_t \frac{\tilde{\mathbf{S}}_q(t, f) (Y_q(t, f) - \hat{S}_q(t, f))^*}{\hat{\lambda}_q(t, f)}.$$

We then plug the closed-form solutions to (5) and (6), compute the \mathcal{L}_{MC} loss in (4), and optimize the DNN.

D. Interpretation of MC Loss and FCP Filtering

To facilitate understanding, we give an interpretation of the proposed \mathcal{L}_{MC} loss.

Suppose that $\mathcal{F}(\cdot, \cdot)$ is defined as $\mathcal{F}(Y_r(t, f), \hat{Y}_r(t, f)) = |Y_r(t, f) - \hat{Y}_r(t, f)|^2$ (rather than as the one in (7), no weighting is used in FCP (i.e., $\hat{\lambda}_a(t, f) = 1$), and $\alpha = 1$, \mathcal{L}_{MC} in (4) can be alternatively formulated as

$$\begin{aligned} \mathcal{L}_{\text{MC,alt}} &= \sum_f \min_{\mathbf{g}_q(f)} \sum_t |Y_q(t, f) - \hat{S}_q(t, f) - \mathbf{g}_q(f)^H \tilde{\mathbf{S}}_q(t, f)|^2 \\ &\quad + \sum_{p=1, p \neq q}^P \sum_f \min_{\mathbf{h}_p(f)} \sum_t |Y_p(t, f) - \mathbf{h}_p(f)^H \tilde{\mathbf{S}}_q(t, f)|^2, \end{aligned} \quad (11)$$

which, given a DNN estimate \hat{S}_q , first computes the relative transfer functions $\hat{\mathbf{g}}_q(f)$ and $\hat{\mathbf{h}}_p(f)$ that best relate the DNN estimate \hat{S}_q to the mixtures (i.e., best satisfy the physical mixture constraints) by minimizing a quadratic objective per frequency at each microphone, and then uses the minimum as the loss for DNN training. The proposed approach can be considered as a *minimizing the minimum* approach. This echoes the idea of permutation invariant training [67] in supervised speech separation, where the minimum to minimize is first computed by finding the best speaker permutation and then used as the loss to train DNNs.

In our study, we use \mathcal{L}_{MC} in (4) rather than $\mathcal{L}_{\text{MC,alt}}$ for DNN training, as we find that using $\mathcal{F}(\cdot, \cdot)$ defined in (7) and the weighting term in (10) produces better performance.

E. Alternative FCP Filtering at Reference Microphone

An alternative way to compute $\hat{\mathbf{g}}_q(f)$ is as follows:

$$\hat{\mathbf{g}}_q(f) = \underset{\mathbf{g}_q(f)}{\operatorname{argmin}} \sum_t \frac{|Y_q(t, f) - \mathbf{g}_q(f)^H \tilde{\mathbf{S}}_q(t, f)|^2}{\hat{\lambda}_q(t, f)}, \quad (12)$$

where the difference from (8) is that, in the numerator, \hat{S}_q is not subtracted from Y_q . Although this change appears minor, its benefits are non-trivial. We find it a useful regularization that can (a) help USDnet avoid the trivial solution of copying the input mixture as the output (i.e., $\hat{S}_q = Y_q$); (b) enable USDnet to be trained on monaural mixtures where only one microphone can be used in loss computation; and (c) lead to an \hat{S}_q that better approximates S_q and a $\hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f)$ that better approximates R_q , rather than an \hat{S}_q and a $\hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f)$ whose summation in (5) can better approximate the mixture Y_q (and hence better minimize $\mathcal{L}_{\text{MC},q}$) but not respectively better approximating S_q and R_q . We will later detail the first two benefits in Section IV-G

and the third in Section IV-I, after describing the input to USDnet in the next subsection.

F. Number of Microphones in Input and Loss

We can use a subset of microphones in the input or loss to train USDnet. This section describes three cases.

1) *Multi-Channel Input and Loss*: So far, we have mainly presented the loss functions of USDnet, which are computed based on multiple microphones. On the input side, we can feed all the microphone mixtures (i.e., same as the ones used in loss computation) to USDnet. The resulting trained USDnet performs multi-channel unsupervised dereverberation at run time.

2) *Single-Channel Input and Multi-Channel Loss*: USDnet can be trained to realize monaural dereverberation at run time by only feeding in the mixture at the reference microphone (i.e., Y_q) to the DNN while still optimizing \mathcal{L}_{MC} computed based on multi-channel mixtures. Fig. 1 shows the idea. The loss computed on multiple microphones could guide USDnet to model the spectro-temporal patterns in reverberant speech and realize monaural unsupervised dereverberation. Notice that multi-channel mixtures are only required for training. At run time, the trained model performs monaural dereverberation.

3) *Single-Channel Input and Loss*: If the training mixtures are all monaural, we find that we can still train USDnet (although with very limited improvement) if the FCP filtering in (12) is used, but the training would fail if (8) is used for filter estimation. We will describe the rationale in Section IV-G.

G. Necessity of Over-Determined Training Mixtures

This subsection discusses the necessity of over-determined training mixtures, depending on whether $\hat{\mathbf{g}}_q(f)$ is computed using (8) or (12).

1) *When $\hat{\mathbf{g}}_q(f)$ is Computed Using (8)*: In this case, the mixtures for training USDnet need to be over-determined. If the training mixtures are instead all monaural (i.e., using single-channel input and loss), \mathcal{L}_{MC} would only contain the first term $\mathcal{L}_{MC,q}$, which can be optimized to zero by the DNN by simply copying the input mixture as the output (i.e., $\hat{S}_q = Y_q$ and, in this case, $\hat{\mathbf{g}}_q(f)$ computed via (8) would be all-zero since in the numerator $Y_q - \hat{S}_q = 0$). This trivial solution would not yield any dereverberation. By leveraging extra microphone recordings as constraints, the second term in \mathcal{L}_{MC} defined in (4) could help avoid the trivial solution.

2) *When $\hat{\mathbf{g}}_q(f)$ is Computed Using (12)*: In this case, we find that the mixtures for training USDnet can also be monaural (i.e., by using single-channel input and loss). Different from the previous case, in this case if the DNN simply copies the input as output (i.e., $\hat{S}_q = Y_q$), (12) becomes identical to the optimization problem in WPE [9], a classic dereverberation algorithm. Following WPE, the filtering result $\hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f) = \hat{\mathbf{g}}_q(f)^H \tilde{Y}_q(t, f)$ would inevitably, and largely only, approximate late reverberation if the prediction delay Δ used in defining $\tilde{\mathbf{S}}_q(t, f)$ is

reasonably large². As a result, the reconstructed mixture used for computing $\mathcal{L}_{MC,q}$ in (5), $\hat{Y}_q(t, f) = \hat{S}_q(t, f) + \hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f) = Y_q(t, f) + \hat{\mathbf{g}}_q(f)^H \tilde{Y}_q(t, f)$, would produce a large loss, due to the approximated late reverberation $\hat{\mathbf{g}}_q(f)^H \tilde{Y}_q(t, f)$. Therefore, using (12) and minimizing $\mathcal{L}_{MC,q}$ could help USDnet avoid the trivial solution. For similar reasons, using (12) could yield better dereverberation than (8) when multiple microphones are used in loss computation.

H. Difficulties of Time Alignment, and Masking vs. Mapping

In supervised speech separation, the availability of target speech signals can provide a strong supervision at each sample for DNNs to learn to produce an estimate that is strictly time-aligned with target speech. In WPE, a time-aligned estimate can be obtained by only removing estimated late reverberation. In unsupervised neural dereverberation, however, we find it particularly difficult to produce an estimate time-aligned with target speech, simply due to the lack of an explicit sample-level supervision (e.g., oracle target speech). For $\mathcal{L}_{MC,q}$ in (5), even if \hat{S}_q produced by the DNN is not time-aligned with the oracle target speech S_q , $\mathcal{L}_{MC,q}$ can still be small because the filter $\hat{\mathbf{g}}_q(f)$, which is multi-tap, could compensate the time-misalignment through multi-frame filtering. This compensation can also happen when we compute $\mathcal{L}_{MC,p}$ in (6).

To partially address this issue, we propose to use complex masking instead of mapping to obtain target estimate \hat{S}_q , and observe that masking obtains better performance. This could be because, in masking, \hat{S}_q is produced by masking the mixture Y_q , while, in mapping, \hat{S}_q is produced directly by the DNN and could hence more likely be misaligned with target speech. For example, when using complex mapping, at each T-F unit the DNN could just output an estimated signal not time- and gain-aligned with the mixture, e.g., $\hat{S}_q(t, f) = \beta \times Y_q(t + \Delta, f)$ where $\beta \in \mathbb{C}$ denotes an arbitrary scaling factor with $|\beta|$ being small. In this case, in (12), $\hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f)$ can perfectly approximate $Y_q(t, f)$ since now $\tilde{\mathbf{S}}_q(t, f) = [\hat{S}_q(t - K, f), \dots, \hat{S}_q(t - \Delta, f)]^T = \beta \times [Y_q(t + \Delta - K, f), \dots, Y_q(t, f)]^T$ has $Y_q(t, f)$ in the last element; and in (5), the resulting reconstructed mixture $\hat{Y}_q(t, f) = \hat{S}_q(t, f) + \hat{\mathbf{g}}_q(f)^H \tilde{\mathbf{S}}_q(t, f) = \beta \times Y_q(t + \Delta, f) + Y_q(t, f)$ would be very close to the mixture $Y_q(t, f)$ if $|\beta|$ is small, yielding a very small $\mathcal{L}_{MC,q}$ even though the solution is trivial. By using masking, this trivial solution could be avoided, as masking applies a complex-valued filter to the mixture at each T-F unit and it is difficult to modify the phase and magnitude of the mixture in a way such that the resulting estimates would be multiple frames ahead or behind.

I. Discussion on Why \hat{S}_q Could Approximate S_q

So far, we have described the benefits of over-determined conditions in Section III-C, and various regularizations such as

²This is because when the prediction delay Δ is set reasonably large, the target speech in the current T-F unit becomes much less linearly-correlated with the T-F units that are beyond Δ frames in the past [9].

using (12) rather than (8) for FCP filtering and using masking rather than mapping to obtain target estimate \hat{S}_q . One more important regularization we have not discussed is the non-negative prediction delay Δ . We find that these regularizations combined can lead \hat{S}_q to approximate S_q . This subsection provides more intuitions and discussions.

In (5), although the loss is defined on the reconstructed mixture $\hat{Y}_q(t, f) = \hat{S}_q(t, f) + \hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$, our goal is to have \hat{S}_q approximate direct-path S_q and $\hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$ approximate reverberation $R_q(t, f)$. In this case, the prediction delay Δ has to be positive. If it is zero, \hat{S}_q can be absorbed into $\hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$, and (5) would have the same form as (6). This would not lead \hat{S}_q to approximate S_q , as there are infinite solutions to the estimated filter $\hat{\mathbf{g}}_q$ and source \hat{S}_q that can minimize the losses in (5) and (6).

Suppose that the prediction delay Δ is positive, masking is used, and noise ε is negligible. The loss in (5) would encourage \hat{S}_q to approximate S_q . Our interpretation consists of three parts:

- When masking is used, \hat{S}_q would be constrained to be approximately time-aligned with S_q (see our discussion in Section IV-H).
- Due to the positive prediction delay, $\hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$, with $\hat{\mathbf{g}}_q(f)$ computed via (12), largely can only approximate late reverberation.
- To minimize (5), \hat{S}_q and $\hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$ need to add up to the mixture. Since $\hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$ can only approximate (or *explain*) the late reverberation component of the mixture, \hat{S}_q has to *explain* the rest components, which mainly include (a) the direct-path signal S_q ; and (b) some early reflections that cannot be explained by $\hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$ when the prediction delay Δ is set to a large positive value.

This interpretation highlights the importance of using masking together with a non-negative prediction delay Δ . If mapping is used instead, it could predict multiple frames ahead and break the regularizations enforced by the positive prediction delay.

The third point above suggests that \hat{S}_q would inevitably contain some early reflections due to the positive prediction delay. Similarly to that in WPE [9], this could be fine in many applications. For example, including some early reflections can improve speech intelligibility [14].

On the other hand, the third point also suggests that we may need a noise estimate (besides the speech estimate) to *explain* the noise component in the mixture. We will describe this later in Section IV-K.

J. Input Microphone Dropout

When multiple microphone signals are used as the network input and in loss computation, we find that USDnet often optimizes the loss well but the dereverberation result is poor. This could be because USDnet sees, from the input, exactly the same mixtures used in loss computation and, therefore, could figure out a way to optimize the loss well but not dereverberate well.

To deal with this issue, we propose to apply dropout [68] to input non-reference microphone signals, where, during training,

each non-reference microphone signal in the input is entirely dropped out with a probability of θ .

K. Introducing Garbage Sources

To model environmental noises (assumed weak) and the modeling error caused by narrow-band approximation (i.e., ε' in (1) and (2)), we propose to use a garbage source to absorb them.

We modify $\mathcal{L}_{MC,q}$ in (5) and $\mathcal{L}_{MC,p}$ in (6) as follows:

$$\mathcal{L}_{MC,q} = \sum_{t,f} \mathcal{F}(Y_q(t, f), \hat{S}_q(t, f) + \hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f) + \hat{\mathbf{w}}_q(f)^H \check{\tilde{\mathbf{V}}}_q(t, f)), \quad (13)$$

$$\mathcal{L}_{MC,p} = \sum_{t,f} \mathcal{F}(Y_p(t, f), \hat{\mathbf{h}}_p(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f) + \hat{\mathbf{w}}_p(f)^H \check{\tilde{\mathbf{V}}}_q(t, f)), \quad (14)$$

where the DNN is trained to output one more spectrogram $\check{\tilde{\mathbf{V}}}_q$ in addition to \hat{S}_q , $\check{\tilde{\mathbf{V}}}_q(t, f) = [\hat{V}_q(t-L, f), \dots, \hat{V}_q(t+L, f)]^T \in \mathbb{C}^{L+1+L}$ stacks a window of $L+1+L$ T-F units in \hat{V}_q , and $\hat{\mathbf{g}}_q(f)$ and $\hat{\mathbf{h}}_p(f)$ are respectively computed in the same ways as in (8) or (12), and (9). $\hat{\mathbf{w}}_q(f)$ and $\hat{\mathbf{w}}_p(f)$ are computed as follows, similarly to (9):

$$\hat{\mathbf{w}}_a(f) = \underset{\mathbf{w}_a(f)}{\operatorname{argmin}} \sum_t \frac{|Y_a(t, f) - \mathbf{w}_a(f)^H \check{\tilde{\mathbf{V}}}_q(t, f)|^2}{\hat{\lambda}_a(t, f)}, \quad (15)$$

where a indexes all the microphones.

We point out that using garbage sources to absorb modeling errors is a widely-adopted technique in conventional separation algorithms such as IVA [58], and CAGMM- and MESSL-based spatial clustering [69], [70]. Our novelty here is adapting this idea for DNN-based unsupervised dereverberation.

L. Run-Time Inference

At run time, we use \hat{S}_q as the final prediction. We just need to run feed-forwarding for inference, without needing to compute FCP filters.

In other words, USDnet is trained to optimize the \mathcal{L}_{MC} loss in (4) so that it can learn to model signal patterns in reverberant speech for dereverberation. At run time, it can be used to dereverberate signals in the same way as models trained in a supervised way. This novel design makes USDnet possible to be readily integrated with semi-supervised learning, where, for example, (a) USDnet trained on massive unlabeled data can be fine-tuned on labeled target-domain data via supervised learning; and (b) pre-trained supervised models can be adapted in an unsupervised way on target-domain unlabeled data by initializing USDnet with the pre-trained model.

Following WPE [9], we can alternatively use $Y_q(t, f) - \hat{\mathbf{g}}_q(f)^H \hat{\tilde{\mathbf{S}}}_q(t, f)$ as the final prediction, where the subtracted term is obtained by filtering past DNN estimates with a

prediction delay Δ (in $\tilde{\hat{S}}_q(t, f)$) and hence can be viewed as an estimate of reverberation to remove. However, in our experiments the performance is worse than directly using \hat{S}_q .

V. EXPERIMENTAL SETUP

We train USDnet using an existing speech dereverberation dataset, and evaluate it on its simulated test set and the REVERB dataset. This section describes the datasets, miscellaneous setup, baseline systems, and evaluation metrics.

A. WSJ0CAM-DEREVERB Dataset

WSJ0CAM-DEREVERB is simulated and has been used in several supervised dereverberation studies [31], [64], [71], which reported strong results. The dry source signals are from the WSJ0CAM corpus, which has 7,861, 742 and 1,088 utterances respectively in its training, validation and test sets. Based on them, 39,293 (~ 77.7 h), 2,968 (~ 5.6 h) and 3,262 (~ 6.4 h) noisy-reverberant mixtures are respectively simulated as the training, validation and test sets. For each utterance, a room with random room characteristics and speaker and microphone locations is sampled. The simulated microphone array has 8 microphones uniformly placed on a circle with a diameter of 20 cm. The speaker-to-array distance is drawn from the range $[0.75, 2.5]$ m and the reverberation time (T60) from $[0.2, 1.3]$ s. For each utterance, an 8-channel diffuse air-conditioning noise is sampled from the REVERB dataset [72], and added to the reverberant speech at an SNR (between the direct-path signal and the noise) sampled from the range $[5, 25]$ dB. The sampling rate is 16 kHz.

B. REVERB Dataset

We evaluate USDnet, trained on WSJ0CAM-DEREVERB, directly on the REVERB dataset [72], which has real near- and far-field reverberant speech recorded in rooms with air-conditioning noises by an 8-microphone circular array with a diameter of 20 cm. The rooms have a T60 at around 0.7 s and the speaker to microphone distances are around 1 m in the near-field case and 2.5 m in the far-field case. We feed dereverberated signals produced by USDnet to an ASR backend for recognition. The ASR backend is trained by using the default recipe in Kaldi,³ which trains a TDNN-based acoustic model based on the default multi-condition training data consisting of the noisy-reverberant speech of REVERB. The sampling rate is 16 kHz.

We emphasize that the purpose of this evaluation is to show the effectiveness of USDnet on real-recorded data and compare its performance with WPE, not to obtain state-of-the-art ASR performance on REVERB.

C. Miscellaneous System Setup

The STFT window size is 32ms, hop size 8ms, and the square root of Hann window is used as the analysis window. We use

512-point discrete Fourier transform to extract 257-dimensional complex spectra at each frame.

We employ TF-GridNet [64] as the DNN architecture. Using symbols defined in [64, Table I], we set its hyper-parameters to $D = 48$, $B = 4$, $I = 4$, $J = 4$, $H = 192$, $L = 4$ and $E = 2$. Please do not confuse the symbols of TF-GridNet with the ones in this paper. When the DNN is trained for complex ratio masking, we truncate the RI parts of the estimated masks into the range $[-5, 5]$ before multiplying it with the mixture.

We consider 1-, 2-, 4- and 8-microphone dereverberation. The first microphone is always considered as the reference microphone. For 1-channel dereverberation, we use the mixture signal captured by microphone 1; for 2-channel dereverberation, we use microphone 1 and 4; for 4-channel dereverberation, we use microphone 1, 3, 5 and 7; and for 8-channel dereverberation, all the 8 microphones are used.

D. Baseline Systems

We consider WPE [9] as the baseline. It is unsupervised and is so far the most popular and successful dereverberation algorithm. We use the implementation in the *nara_wpe* toolkit [73]. The STFT configuration for WPE is the same as that in USDnet. Following [12], [73], the filter tap is tuned to 37 in monaural cases, to 10 in 4-channel cases, to 5 in 8-channel cases, the prediction delay is 3, and 3 iterations is performed. Building upon the configurations of WPE, we additionally provide the results of DNN-WPE [26], where the target power spectral density used in WPE is provided by a supervised dereverberation DNN. The DNN has the same architecture as that in USDnet. It is trained via complex masking in the same way as USDnet, but the loss is defined on the RI components and magnitude of estimated target speech \hat{S}_q and is weighted by the summation of mixture magnitude, following the distance metric in (7).

We considered comparing USDnet with unsupervised neural dereverberation models such as [44], [46], [47]. However, they only deal with monaural cases and require separate source prior models trained on anechoic speech or DNN-based metric models (such as DNSMOS) trained on annotated anechoic and reverberant speech, and cannot be directly trained solely on a set of reverberant signals. Differently, USDnet exploits implicit regularizations afforded by multiple microphones and can be trained directly on a set of reverberant signals. In other words, they utilize very different signal principles, which can be likely integrated with USDnet in future studies. We therefore do not consider them as baselines for comparison, and mainly focus on showing the effectiveness of leveraging mixture constraints in this paper.

E. Evaluation Metrics

For evaluation on the test set of WSJ0CAM-DEREVERB, we use the direct-path signal at the first microphone (designated as the reference microphone) as the reference signal for metric computation. We report scores of perceptual evaluation of speech quality (PESQ) [74] and extended short-time objective intelligibility (eSTOI) [75]. They are widely-adopted objective metrics of speech quality and intelligibility. We use the

³commit 61637e6c8ab01d3b4c54a50d9b20781a0aa12a59

TABLE I
MAIN HYPER-PARAMETERS OF USDNET

Symbols	Description
K, Δ	$\tilde{\mathbf{S}}_q(t, f) = [\hat{S}_q(t - K + 1, f), \dots, \hat{S}_q(t - \Delta, f)]^T \in \mathbb{C}^{K-\Delta}$
I, J	$\tilde{\mathbf{S}}_q(t, f) = [\hat{S}_q(t - I + 1, f), \dots, \hat{S}_q(t + J, f)]^T \in \mathbb{C}^{I+J}$
L	$\tilde{\mathbf{V}}_q(t, f) = [\hat{V}_q(t - L, f), \dots, \hat{V}_q(t + L, f)]^T \in \mathbb{C}^{L+1+L}$ for garbage source
α	Microphone weight for non-reference microphones
θ	Dropout ratio for input non-reference microphones

TABLE II
RESULTS ON TEST SET OF WSJOCAM-DEREVERB USING VARIOUS NUMBER OF MICROPHONE SIGNALS IN INPUT AND LOSS

Row	Systems	#Mics in input/loss	$\hat{\mathbf{g}}_q(f)$	$I/J/K/\Delta/L$	Masking/Mapping	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow
1	Mixture	1 / -	-	-	-	-	-	1.64	0.494	-3.6
2a	USDnet	1 / 1	(12)	40 / 0 / 40 / 3 / -	Masking	1	-	1.76	0.561	-2.1
2b	USDnet	1 / 2	(12)	40 / 0 / 40 / 3 / -	Masking	1	-	2.34	0.735	1.6
2c	USDnet	1 / 4	(12)	40 / 0 / 40 / 3 / -	Masking	1	-	2.51	0.760	2.5
2d	USDnet	1 / 8	(12)	40 / 0 / 40 / 3 / -	Masking	1	-	2.35	0.742	2.2
3b	USDnet	2 / 2	(12)	40 / 0 / 40 / 3 / -	Masking	1	0	2.17	0.696	1.1
3c	USDnet	4 / 4	(12)	40 / 0 / 40 / 3 / -	Masking	1	0	2.28	0.716	1.6
3d	USDnet	8 / 8	(12)	40 / 0 / 40 / 3 / -	Masking	1	0	2.33	0.730	2.0
4a	USDnet	1 / 1	(12)	40 / 0 / 40 / 3 / -	Mapping	1	-	1.61	0.493	-3.6
4b	USDnet	1 / 8	(12)	40 / 0 / 40 / 3 / -	Mapping	1	-	1.89	0.279	-36.0
4c	USDnet	8 / 8	(12)	40 / 0 / 40 / 3 / -	Mapping	1	0	1.87	0.507	-28.5
5a	USDnet	1 / 1	(8)	40 / 0 / 40 / 3 / -	Masking	1	-	1.64	0.494	-3.6
5b	USDnet	1 / 8	(8)	40 / 0 / 40 / 3 / -	Masking	1	-	2.08	0.671	-0.1
5c	USDnet	8 / 8	(8)	40 / 0 / 40 / 3 / -	Masking	1	0	2.08	0.678	0.0

python-pesq toolkit to report narrow-band MOS-LQO scores for PESQ, and the *pystoi* toolkit for eSTOI. Additionally, we report scale-invariant signal-to-distortion ratio (SI-SDR) [76] but do not consider it as a primary evaluation metric due to the difficulties, and in many applications unnecessary, of time alignment. For the ASR evaluation on REVERB, we report word error rates (WER).

All of these metrics, except WER, favor estimated signals that are time-aligned with reference signals.

VI. EVALUATION RESULTS

This section validates USDnet and its design choices based on the WSJOCAM-DEREVERB and REVERB datasets. Table I lists the main hyper-parameters of USDnet. We start with $I = 40$, $J = 0$, $K = 40$, $\Delta = 3$, no garbage sources (i.e., the filter tap L is shown as “-” in the tables), no microphone weight (i.e., $\alpha = 1$), and no dropout to input microphones (i.e., $\theta = 0$).

A. Effects of Using More Microphones in Loss and Input

In row 2a of Table II, the training mixtures are all monaural. The performance is not good, since only one microphone signal can be used as the mixture constraint and this is likely not enough for the DNN to solve the ill-posed problem and learn to figure out what the target signal is. On the other hand, through the formulation of $\mathcal{L}_{MC,q}$ in (5) and using (12) for filter estimation, USDnet does not fail completely and shows some effectiveness.

In row 2b-2d of Table II, only one microphone signal is used in input but multiple microphone signals are used for loss computation. We observe that the dereverberation performance gets much better, compared with row 2a. This supports our idea

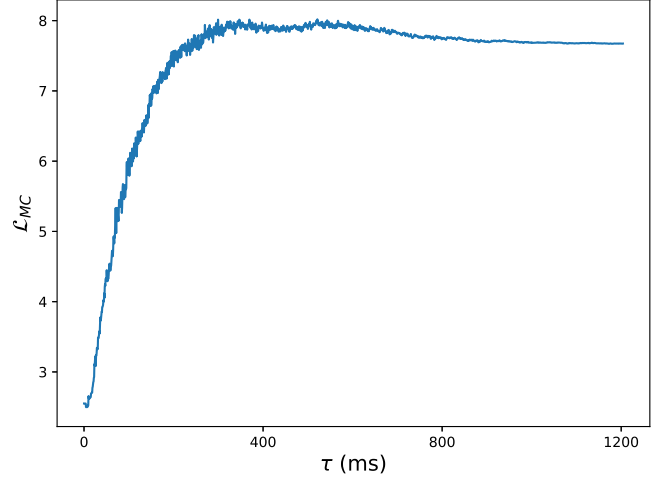


Fig. 2. Loss curve of \mathcal{L}_{MC} against truncated length of relative RIR, τ . For this example utterance, the T60 is 1.204 seconds. The truncation length is enumerated every 10 samples, starting from one sample long all the way up to the length of T60. See Section VI-B for more details.

that using extra microphone signals as mixture constraints in the loss can help USDnet better figure out the target signal.

In row 3b-3d, multiple microphone signals are used in input and all of them are used for loss computation. We observe better performance than row 2a.

Next, we use the results in 2a-2d and 3b-3d as baselines, and validate the effects of various design choices in USDnet.

B. Illustration of Loss Curve of \mathcal{L}_{MC}

To show that minimizing \mathcal{L}_{MC} can promote unsupervised dereverberation, we plot the loss curve of \mathcal{L}_{MC} in Fig. 2, by using a simulated noisy-reverberant speech signal sampled from WSJOCAM-DEREVERB. Given the simulated direct-path RIR $o_d \in \mathbb{R}^{M_d}$ and reverberant-speech RIR $o_i \in \mathbb{R}^{M_i}$, both at the reference microphone q , we first compute the relative RIR o_r relating the direct-path signal to reverberant speech as follows:

$$o_r = \text{iFFT} \left(\frac{\text{FFT}(o_i, M_r)}{\text{FFT}(o_d, M_r)}, M_r \right) \in \mathbb{R}^{M_r}, \quad (16)$$

where $M_r = M_i + M_d - 1$, $\text{FFT}(\cdot, M_r)$ computes an M_r -point fast Fourier transform (FFT) of the RIR, and $\text{iFFT}(\cdot, M_r)$ computes an M_r -point inverse FFT (iFFT). We then compute a hypothesized dereverberation result as

$$\hat{S}_q = \text{STFT} (s_q * \text{T}_0^T(o_r)), \quad (17)$$

where s_q is the time-domain direct-path signal corresponding to S_q , the operator $*$ denotes linear convolution, and $\text{T}_0^T(\cdot)$ truncates the RIR to length τ so that we can control how reverberant \hat{S}_q is. We then use \hat{S}_q to compute \mathcal{L}_{MC} based on the hyper-parameters shown in row 2d of Table II, and plot the loss curve against τ . From Fig. 2, we observe much smaller \mathcal{L}_{MC} when \hat{S}_q is configured less reverberant. This partially supports our claim that minimizing \mathcal{L}_{MC} can promote dereverberation.

TABLE III
RESULTS ON TEST SET OF WSJ0CAM-DEREVERB USING VARIOUS
PREDICTION DELAYS Δ

Row	Systems	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow
1	USDnet	8 / 8	40 / 0 / 40 / 1 / -	1	0	2.34	0.717	2.3
2	USDnet	8 / 8	40 / 0 / 40 / 2 / -	1	0	2.34	0.734	2.8
3	USDnet	8 / 8	40 / 0 / 40 / 3 / -	1	0	2.33	0.730	2.0
4	USDnet	8 / 8	40 / 0 / 40 / 4 / -	1	0	2.32	0.726	1.3

C. Effects of Masking vs. Mapping

In row 4a-4c of Table II, we use complex spectral mapping instead of complex masking. The results are clearly worse than those in 2a, 2d and 3d, respectively, and, in particular, the SI-SDR scores are much lower. This is because, in complex spectral mapping, the DNN has fewer restrictions in producing target estimates, and this would result in trivial solutions where reverberation is not reduced (see our discussions in Section IV-H). During training, we indeed observe that, when using mapping, the training loss often suddenly decreases to values very close to zero.

In the rest of this paper, we use masking in default.

D. Effects of (12) vs. (8) for FCP Filtering at Reference Mic

In row 5a-5c of Table II, we use (8) instead of (12) to compute the filter $\hat{g}_q(f)$ at the reference microphone q . The results are respectively much worse than the ones in row 2a, 2d and 3d, indicating the effectiveness of using (12).

In 5a, no improvement is observed over the unprocessed mixture. This is because the network produces a trivial solution by copying the input mixture as the output. See the discussions in the first paragraph of Section IV-G. Differently, in 2a, we observe some improvement over the mixture by using (12) instead of (8) for FCP filtering. The rationale is provided in the second paragraph of Section IV-G.

In the rest of this paper, we use (12) in default.

E. Effects of Non-Negative Prediction Delay Δ

The prediction delay Δ in defining $\tilde{\hat{S}}_q(t, f)$ (see the text below (6)) is a hyper-parameter to tune. Ideally, for $\mathcal{L}_{MC,q}$ in (5), we want $\hat{S}_q(t, f)$ to approximate the direct-path signal and $\hat{g}_q(f)^H \tilde{\hat{S}}_q(t, f)$ to approximate all the reflections so that their summation can add up to the mixture Y_q . When using (12) to compute $\hat{g}_q(f)$, a large Δ would make $\hat{g}_q(f)^H \tilde{\hat{S}}_q(t, f)$ in the numerator not capable of approximating early reflections, and a small Δ could result in a $\hat{g}_q(f)^H \tilde{\hat{S}}_q(t, f)$ that cancels out target direct-path speech. In Table III, we sweep Δ based on the set of $\{1, 2, 3, 4\}$, and observe that setting Δ to 3 or 2 results in better performance. However, the performance differences are not very large.

F. Effects of Filter Length K and I

The filter length K and I in defining $\tilde{\hat{S}}_q(t, f)$ and $\bar{\hat{S}}_q(t, f)$ (see the text below (6)) are hyper-parameters to tune. We sweep

TABLE IV
RESULTS ON TEST SET OF WSJ0CAM-DEREVERB USING VARIOUS FILTER
LENGTH K AND I

Row	Systems	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow
1	USDnet	8 / 8	20 / 0 / 20 / 3 / -	1	0	2.30	0.715	1.3
2	USDnet	8 / 8	40 / 0 / 40 / 3 / -	1	0	2.33	0.730	2.0
3	USDnet	8 / 8	60 / 0 / 60 / 3 / -	1	0	2.35	0.739	2.3
4	USDnet	8 / 8	80 / 0 / 80 / 3 / -	1	0	2.27	0.727	2.8

TABLE V
RESULTS ON TEST SET OF WSJ0CAM-DEREVERB USING VARIOUS FILTER
DELAY J

Row	Systems	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow
1	USDnet	1 / 8	40 / 0 / 40 / 3 / -	1	-	2.35	0.742	2.2
2	USDnet	1 / 8	39 / 1 / 40 / 3 / -	1	-	2.36	0.730	1.6
3	USDnet	8 / 8	40 / 0 / 40 / 3 / -	1	0	2.33	0.730	2.0
4	USDnet	8 / 8	39 / 1 / 40 / 3 / -	1	0	2.31	0.719	1.6

TABLE VI
RESULTS ON TEST SET OF WSJ0CAM-DEREVERB USING VARIOUS
MICROPHONE WEIGHTS α

Row	Systems	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow
1	USDnet	1 / 8	40 / 0 / 40 / 3 / -	1	-	2.35	0.742	2.2
2	USDnet	1 / 8	40 / 0 / 40 / 3 / -	$\frac{3}{8-1}$	-	2.48	0.749	2.2
3	USDnet	8 / 8	40 / 0 / 40 / 3 / -	1	0	2.33	0.730	2.0
4	USDnet	8 / 8	40 / 0 / 40 / 3 / -	$\frac{3}{8-1}$	0	2.40	0.745	2.2

them based on the set of $\{20, 40, 60, 80\}$. Given that the STFT window and hop sizes are respectively 32 and 8ms, this set of filter taps roughly samples filter lengths in the range of $[0.2, 0.7]$ seconds. From Table IV, we observe that using $I = 60$ together with $K = 60$ produces the best performance, but the differences in performance are not very large.

G. Effects of Filter Delay J for Non-Reference Microphones

In the text below (6), $\tilde{\hat{S}}_q(t, f)$ is configured to only contain past T-F units, while $\bar{\hat{S}}_q(t, f)$ is configured to have J future T-F units. In earlier experiments, J is set to 0, while, in this subsection, we increase it to 1 to account for the fact that the reference microphone is not always the closest microphone to the target speaker. Notice that we do not need more future taps, since the microphone array is a compact array with an aperture size of only 20 cm, which is much small than the speed of sound. From Table V, we observe worse performance in almost all the entries.

H. Effects of Weighting Microphones

Table VI presents the results of using a smaller microphone weight α in (4) for non-reference microphones. The motivation is that when setting $\alpha = 1.0$ and many microphones are used to compute \mathcal{L}_{MC} in (4), the loss on non-reference microphones would dominate the overall loss, causing the loss on the reference microphone, $\mathcal{L}_{MC,q}$, which is more relevant to the target signal we aim to estimate, not optimized well. In our experiments, if

TABLE VII
RESULTS ON TEST SET OF WSJ0CAM-DEREVERB USING INPUT
MICROPHONE DROPOUT

Row	Systems	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow
1a	USDnet	2 / 2	40 / 0 / 40 / 3 / -	1	0.0	2.17	0.696	1.1
1b	USDnet	4 / 4	40 / 0 / 40 / 3 / -	1	0.0	2.28	0.716	1.6
1c	USDnet	8 / 8	40 / 0 / 40 / 3 / -	$\frac{3}{8-1}$	0.0	2.40	0.745	2.2
2a	USDnet	2 / 2	40 / 0 / 40 / 3 / -	1	0.7	2.37	0.745	2.1
2b	USDnet	4 / 4	40 / 0 / 40 / 3 / -	1	0.7	2.53	0.751	2.3
2c	USDnet	8 / 8	40 / 0 / 40 / 3 / -	$\frac{3}{8-1}$	0.7	2.42	0.746	2.1

TABLE VIII
RESULTS ON TEST SET OF WSJ0CAM-DEREVERB USING GARBAGE
SOURCES

Row	Systems	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow
1	Mixture	1 / -	-	-	-	1.64	0.494	-3.6
2a	USDnet	1 / 8	60 / 0 / 60 / 3 / -	$\frac{3}{8-1}$	-	2.45	0.757	2.5
2b	USDnet	8 / 8	60 / 0 / 60 / 3 / -	$\frac{3}{8-1}$	0.7	2.42	0.760	2.6
3a	USDnet	1 / 8	60 / 0 / 60 / 3 / 1	$\frac{3}{8-1}$	-	2.53	0.772	2.9
3b	USDnet	8 / 8	60 / 0 / 60 / 3 / 1	$\frac{3}{8-1}$	0.7	2.45	0.761	2.7
4a	WPE [9]	1 / -	-	-	-	1.78	0.579	-1.7
4b	WPE [9]	8 / -	-	-	-	2.02	0.690	2.0
5a	DNN-WPE [26]	1 / -	-	-	-	1.81	0.611	-1.2
5b	DNN-WPE [26]	8 / -	-	-	-	2.07	0.724	2.8

there are P (> 4) microphones for loss computation, we set $\alpha = \frac{3}{P-1}$, and still set $\alpha = 1.0$ otherwise. In Table VI, better performance is observed.

I. Effects of Input Microphone Dropout

In Table VII, we report the results of applying input microphone dropout to input non-reference microphone signals during training. The dropout probability θ of each non-reference microphone is 0.7 in default. Better performance is observed for all the 2-, 4- and 8-channel cases.

J. Effects of Using Garbage Sources

In Table VIII, we increase the filter taps I and K to 60 and report the results of using one garbage source to absorb environmental noises and modeling errors. We tune the FCP filter for filtering the garbage source to 3-tap, by setting L to 1 (see the text below (14) for the definitions). Comparing row 2a with 3a, and 2b with 3b, we observe noticeable improvements.

K. Comparison With WPE and DNN-WPE on WSJ0CAM-DEREVERB

Table VIII also reports the results of monaural and eight-channel WPE and supervised DNN-WPE. USDnet obtains clearly better performance than WPE, especially in monaural input cases (e.g., 2.53vs. 1.78in PESQ). DNN-WPE obtains better results than WPE. USDnet obtains better performance than DNN-WPE in almost all the entries, except on SI-SDR in the eight-channel-input case (i.e., 2.7vs. 2.8dB SI-SDR).

L. Illustration of Dereverberation Results

Fig. 3 illustrates and compares the dereverberation results of USDnet and WPE, based on a simulated reverberant speech

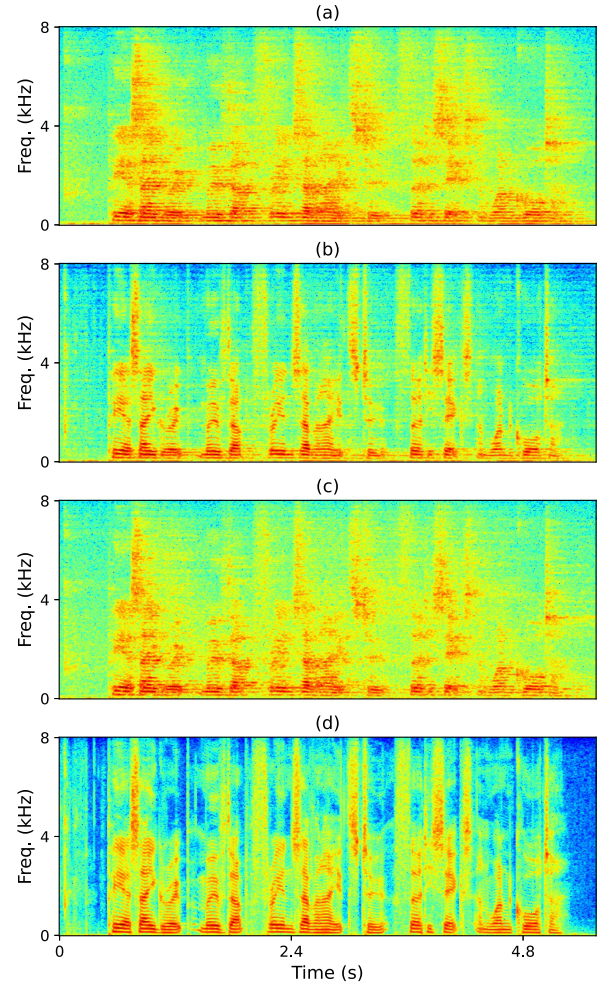


Fig. 3. Illustration of (a) unprocessed mixture; (b) Dereverberation results of USDnet (configured in row 3a of Table VIII); (c) Dereverberation results of WPE (row 4a of Table VIII); And (d) clean anechoic speech. For this example, the T60 is 1.204 seconds and direct-to-reverberation energy ratio is around -5.9 dB, the PESQ scores are respectively 1.44, 2.41, 1.51 and 4.55, and eSTOI scores are respectively 0.322, 0.735, 0.423 and 1.0. Best in color.

signal sampled from WSJ0CAM-DEREVERB. We observe that USDnet produces clear suppression of reverberation and clear reconstruction of target speech patterns, while WPE only reduces reverberation slightly and the speech patterns are still very smeared. This comparison suggests that USDnet can better reduce reverberation than WPE. A sound demo is provided (see the end of Section I).

M. ASR Results on REVERB

In Tables IX and X, we evaluate USDnet trained on WSJ0CAM-DEREVERB directly on the ASR tasks of REVERB. Although WSJ0CAM-DEREVERB is simulated and is mismatched with the real-recorded REVERB dataset, we observe that USDnet can significantly improve ASR performance in both near- and far-field cases over unprocessed mixtures. In both tables, USDnet outperforms WPE and DNN-WPE.

In Table X, we observe that applying input microphone dropout (see the θ column), although producing better enhancement scores, is detrimental to ASR.

TABLE IX
WER (%) ON REAL DATA OF REVERB (1-CHANNEL INPUT)

Row	Systems	Cross-reference	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow	WER (%) \downarrow on val. set			WER (%) \downarrow on test set		
										Near	Far	Avg.	Near	Far	Avg.
1	Mixture	-	1 / -	-	-	-	1.64	0.494	-3.6	15.4	16.9	16.1	17.1	17.3	17.2
2a	USDnet	2a of Table II	1 / 1	40 / 0 / 40 / 3 / -	1	-	1.76	0.561	-2.1	21.1	24.2	22.6	20.7	22.1	21.4
2b	USDnet	2d of Table II	1 / 8	40 / 0 / 40 / 3 / -	1	-	2.35	0.742	2.2	11.7	15.0	13.4	11.6	12.6	12.1
2c	USDnet	2 of Table VI	1 / 8	40 / 0 / 40 / 3 / -	$\frac{3}{8-1}$	-	2.48	0.749	2.2	10.3	12.4	11.4	10.6	11.8	11.2
2d	USDnet	2a of Table VIII	1 / 8	60 / 0 / 60 / 3 / -	$\frac{3}{8-1}$	-	2.45	0.757	2.5	10.3	13.3	11.8	10.2	11.5	10.9
2e	USDnet	3a of Table VIII	1 / 8	60 / 0 / 60 / 3 / 1	$\frac{3}{8-1}$	-	2.53	0.772	2.9	10.4	12.4	11.4	10.1	10.7	10.4
3a	WPE [9]	4a of Table VIII	1 / -	-	-	-	1.78	0.579	-1.7	15.6	16.0	15.8	13.2	15.1	14.2
3b	DNN-WPE [26]	5a of Table VIII	1 / -	-	-	-	2.02	0.690	2.0	14.3	16.1	15.2	12.8	14.7	13.7

The bold values mean the best score in each column.

TABLE X
WER (%) ON REAL DATA OF REVERB (8-CHANNEL INPUT)

Row	Systems	Cross-reference	#CH in input/loss	$I/J/K/\Delta/L$	α	θ	PESQ \uparrow	eSTOI \uparrow	SI-SDR (dB) \uparrow	WER (%) \downarrow on val. set			WER (%) \downarrow on test set		
										Near	Far	Avg.	Near	Far	Avg.
1	Mixture	-	8 / -	-	-	-	1.64	0.494	-3.6	15.4	16.9	16.1	17.1	17.3	17.2
2a	USDnet	3d of Table II	8 / 8	40 / 0 / 40 / 3 / -	1	0.0	2.33	0.730	2.0	9.0	11.1	10.0	8.9	9.5	9.2
2b	USDnet	4 of Table VI	8 / 8	40 / 0 / 40 / 3 / -	$\frac{3}{8-1}$	0.0	2.40	0.745	2.2	9.2	11.9	10.5	8.5	8.8	8.6
2c	USDnet	2c of Table VII	8 / 8	40 / 0 / 40 / 3 / -	$\frac{3}{8-1}$	0.7	2.42	0.746	2.1	10.4	12.0	11.2	10.3	9.3	9.8
2d	USDnet	2b of Table VIII	8 / 8	60 / 0 / 60 / 3 / -	$\frac{3}{8-1}$	0.7	2.42	0.760	2.6	9.9	12.0	10.9	9.7	9.8	9.7
2e	USDnet	3b of Table VIII	8 / 8	60 / 0 / 60 / 3 / 1	$\frac{3}{8-1}$	0.7	2.45	0.761	2.7	10.7	11.4	11.0	9.5	9.6	9.5
3a	WPE [9]	4b of Table VIII	8 / -	-	-	-	2.02	0.690	2.0	15.5	19.3	17.4	11.8	13.9	12.9
3b	DNN-WPE [26]	5b of Table VIII	8 / -	-	-	-	2.07	0.724	2.8	14.7	18.7	16.7	11.7	12.8	12.3

The bold values mean the best score in each column.

Comparing the results in Tables IX and X, we observe that USDnet with eight-channel input obtains better recognition performance but worse PESQ, eSTOI and SI-SDR scores than the one with monaural input.

VII. MISCELLANEOUS DISCUSSIONS

This section discusses USDnet in aspects other than dereverberation performance, putting USDnet in a broader context.

1) *Algorithmic Complexity*: As described in Section IV-L, at run time the trained model performs inference in the same way as existing supervised dereverberation algorithms (i.e., just running a forward pass to obtain target estimate \hat{S}_q). During training, the extra amount of computation compared to supervised algorithms stems from estimating FCP filters and computing the \mathcal{L}_{MC} loss, which are typically much less costly than modern DNN modules.

2) *Amenability to Online Dereverberation*: USDnet can be easily configured for online dereverberation. We only need to change the DNN to be causal. We leave this investigation to future research.

3) *Array-Geometry-Invariant Modeling*: USDnet can be modified to be trained on mixtures recorded by arrays with diverse microphone geometries. Clearly, the \mathcal{L}_{MC} loss is invariant to microphone geometry. We only need to use a DNN architecture that can model mixtures with a diverse number of input microphones, by following, e.g., the VarArray approach [77]. We leave this to future research.

VIII. LIMITATIONS

A limitation of USDnet stems from the assumption that there is only one target speaker source and the considered noises are weak and stationary. In realistic conditions (such as in conversational applications), there could be an unknown number of target

speaker sources and an unknown number of directional and diffuse sources, and we need to address simultaneous separation, dereverberation and enhancement. Future research will modify USDnet for these more challenging cases. Our recent studies such as SuperM2M [78] and cross-talk reduction [79] have made initial investigation along this direction.

Another limitation is that, in our experiments, multi-channel USDnet trained with, e.g., eight-channel input and loss shows worse performance than single-channel USDnet trained with, e.g., single-channel input and eight-channel loss. This issue is likely because in the case of eight-channel input and loss, USDnet sees the same mixtures used in loss computation from the input, and hence could figure out a way to optimize the loss well but not dereverberate well. This issue needs to be addressed in future research to build better multi-channel models.

IX. CONCLUSION

We have formulated unsupervised neural speech dereverberation as a blind deconvolution problem, which requires estimating both the source and filter, and proposed a novel algorithm named USDnet, which leverages over-determined training mixtures as constraints to narrow down the solutions to target speech. Evaluation results on two dereverberation tasks show that USDnet, even if only trained on a set of reverberant mixtures in an unsupervised way, can learn to effectively reduce reverberation. Future research will modify USDnet and evaluate it on real-recorded conversational speech dereverberation and separation tasks with multiple sound sources.

A contribution of this paper, we highlight, is our finding that DNNs can be trained directly on a set of reverberant mixtures, in an unsupervised way, to reduce reverberation, by solely exploiting mixture constraints afforded by multi-microphone

mixtures. The novel way of modeling reverberation and the novel concept of leveraging mixture constraints exploit signal cues and physical principles very different from the ones utilized by existing algorithms. The proposed algorithms, hence, we think, would in many ways complement existing dereverberation and separation algorithms, and motivate new ones.

In closing, another contribution of this paper, we point out, is our unsupervised deep learning based approach for solving blind deconvolution problems, which exist not only in speech dereverberation but also in many other engineering applications. By modeling source priors in an unsupervised way, filter estimation becomes a differentiable operation given a source estimate. This enables us to train DNNs on signal patterns in a discriminative way to optimize mixture-constraint loss and realize unsupervised deconvolution. This novel methodological contribution, we think, could generate broader impact beyond speech dereverberation.

REFERENCES

- [1] E. A. P. Habets and P. A. Naylor, "Dereverberation," in *Audio Source Separation and Speech Enhancement*, E. Vincent, T. Virtanen, and S. Gannot, Eds., Hoboken, NJ, USA: Wiley, 2018, pp. 317–343.
- [2] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding blind deconvolution algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2354–2367, Dec. 2011.
- [3] "Blind deconvolution." [Online]. Available: https://en.wikipedia.org/wiki/Blind_deconvolution
- [4] T. Yoshioka et al., "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 114–126, Nov. 2012.
- [5] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multi-microphone speech enhancement and source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 4, pp. 692–730, Apr. 2017.
- [6] R. Haeb-Umbach, J. Heymann, L. Drude, S. Watanabe, M. Delcroix, and T. Nakatani, "Far-field automatic speech recognition," *Proc. IEEE*, vol. 109, no. 2, pp. 124–148, Feb. 2021.
- [7] T. Yoshioka, T. Nakatani, and M. Miyoshi, "Integrated speech enhancement method using noise suppression and dereverberation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 2, pp. 231–246, Feb. 2009.
- [8] T. Yoshioka, "Speech enhancement in reverberant environments," Ph.D. dissertation, 2010.
- [9] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 7, pp. 1717–1731, Sep. 2010.
- [10] A. Jukiać, T. V. Waterschoot, T. Gerkmann, and S. Doclo, "Multi-channel linear prediction-based speech dereverberation with sparse priors," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 9, pp. 1509–1520, Sep. 2015.
- [11] A. Jukić, T. V. Waterschoot, T. Gerkmann, and S. Doclo, "A general framework for incorporating time-frequency domain sparsity in multichannel speech dereverberation," *J. Audio Eng. Soc.*, vol. 65, no. 1/2, pp. 17–30, 2017.
- [12] T. Nakatani and K. Kinoshita, "A unified convolutional beamformer for simultaneous denoising and dereverberation," *IEEE Signal Process. Lett.*, vol. 26, no. 6, pp. 903–907, Jun. 2019.
- [13] C. Boeddeker, T. Nakatani, K. Kinoshita, and R. Haeb-Umbach, "Jointly optimal dereverberation and beamforming," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 216–220.
- [14] B. C. J. Moore, *An Introduction to the Psychology of Hearing*. Leiden, The Netherlands: Brill, 2012.
- [15] K. Han, Y. Wang, D. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 6, pp. 982–992, Jun. 2015.
- [16] T. May, "Robust speech dereverberation with a neural network-based post-filter that exploits multi-conditional training of binaural cues," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 2, pp. 406–414, Feb. 2018.
- [17] O. Ernst, S. E. Chazan, S. Gannot, and J. Goldberger, "Speech dereverberation using fully convolutional networks," in *Proc. IEEE Eur. Signal Process. Conf.*, 2018, pp. 390–394.
- [18] Y. Zhao, D. Wang, B. Xu, and T. Zhang, "Late reverberation suppression using recurrent neural networks with long short-term memory," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5434–5438.
- [19] Y. Zhao, D. Wang, B. Xu, and T. Zhang, "Monaural speech dereverberation using temporal convolutional networks with self attention," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1598–1607, 2020.
- [20] Z.-Q. Wang and D. Wang, "Deep learning based target cancellation for speech dereverberation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 941–950, 2020.
- [21] Z.-Q. Wang and D. Wang, "Multi-microphone complex spectral mapping for speech dereverberation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 486–490.
- [22] V. Kothapally and J. H. Hansen, "SkipConvGAN: Monaural speech dereverberation using generative adversarial networks via complex time-frequency masking," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1600–1613, 2022.
- [23] J. M. Lemerrier, J. Thiemann, R. Koning, and T. Gerkmann, "A neural network-supported two-stage algorithm for lightweight dereverberation on hearing devices," *EURASIP J. Audio, Speech, Music Process.*, vol. 2023, Art. no. 18.
- [24] V. Kothapally and J. H. L. Hansen, "Monaural speech dereverberation using deformable convolutional networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 1712–1723, 2024.
- [25] L. Alzubaidi et al., "A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications," *J. Big Data*, vol. 10, 2023, Art. no. 46.
- [26] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, and T. Nakatani, "Neural network-based spectrum estimation for online WPE dereverberation," in *Proc. Interspeech*, 2017, pp. 384–388.
- [27] J. Heymann, L. Drude, R. Haeb-Umbach, K. Kinoshita, and T. Nakatani, "Frame-online DNN-WPE dereverberation," in *Proc. IEEE 16th Int. Workshop Acoustic Signal Enhancement*, 2018, pp. 466–470.
- [28] J. Heymann, L. Drude, R. Haeb-Umbach, K. Kinoshita, and T. Nakatani, "Joint optimization of neural network-based WPE dereverberation and acoustic model for robust online ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6655–6659.
- [29] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *Proc. IEEE 13th ITG Symp.*, 2018, pp. 1–5.
- [30] W. Zhang, A. S. Subramanian, X. Chang, S. Watanabe, and Y. Qian, "End-to-end far-field speech recognition with unified dereverberation and beamforming," in *Proc. Interspeech*, 2020, pp. 324–328.
- [31] Z.-Q. Wang, G. Wichern, and J. L. Roux, "Convolutional prediction for monaural speech dereverberation and noisy-reverberant speaker separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3476–3490, 2021.
- [32] J. Richter, S. Welker, J. M. Lemerrier, B. Lay, and T. Gerkmann, "Speech enhancement and dereverberation with diffusion-based generative models," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 2351–2364, 2023.
- [33] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Interspeech*, 2018, pp. 1561–1565.
- [34] S. Watanabe et al., "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *Proc. CHiME*, 2020, pp. 1–7.
- [35] W. Zhang, J. Shi, C. Li, S. Watanabe, and Y. Qian, "Closing the gap between time-domain multi-channel speech enhancement on real and simulation conditions," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2021, pp. 146–150.
- [36] E. Tzinis, Y. Adi, V. K. Ithapu, B. Xu, P. Smaragdis, and A. Kumar, "RemixIT: Continual self-training of speech enhancement models via bootstrapped remixing," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1329–1341, Oct. 2022.
- [37] S. Leglaive et al., "The CHiME-7 UDASE task: Unsupervised domain adaptation for conversational speech enhancement," in *Proc. CHiME*, 2023, pp. 7–12.
- [38] Z.-Q. Wang and S. Watanabe, "UNSSOR: Unsupervised neural speech separation by leveraging over-determined training mixtures," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2023, pp. 34021–34042.
- [39] S. Wisdom, E. Tzinis, H. Erdogan, R. J. Weiss, K. Wilson, and J. R. Hershey, "Unsupervised sound separation using mixture invariant training," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2020.

- [40] K. Saijo and T. Ogawa, "Self-remixing: Unsupervised speech separation VIA separation and remixing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
- [41] T. Fujimura, Y. Koizumi, K. Yatabe, and R. Miyazaki, "Noisy-target training: A training strategy for DNN-based speech enhancement without clean speech," in *Proc. IEEE 29th Eur. Signal Process. Conf.*, 2021, pp. 436–440.
- [42] T. Fujimura and T. Toda, "Analysis of noisy-target training for DNN-Based speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
- [43] Y. Nakagome, M. Togami, T. Ogawa, and T. Kobayashi, "Efficient and stable adversarial learning using unpaired data for unsupervised multichannel speech separation," in *Proc. Interspeech*, 2021, pp. 3051–3055.
- [44] X. Bie, S. Leglaive, X. Alameda-Pineda, and L. Girin, "Unsupervised speech enhancement using dynamical variational autoencoders," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 2993–3007, 2022.
- [45] X. Hao, C. Xu, and L. Xie, "Neural speech enhancement with unsupervised pre-training and mixture training," *Neural Netw.*, vol. 158, pp. 216–227, 2023.
- [46] K. Saito et al., "Unsupervised vocal dereverberation with diffusion-based generative models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
- [47] S.-W. Fu, C. Yu, K.-H. Hung, M. Ravanelli, and Y. Tsao, "MetricGAN-U: Unsupervised speech enhancement / dereverberation based only on noisy / reverberated speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 7412–7416.
- [48] E. Tzinis, S. Venkataramani, and P. Smaragdis, "Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 81–85.
- [49] L. Drude, D. Hasenklever, and R. Haeb-Umbach, "Unsupervised training of a deep clustering model for multichannel blind source separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 695–699.
- [50] P. Seetharaman, G. Wichern, J. L. Roux, and B. Pardo, "Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 356–360.
- [51] M. Togami, Y. Masuyama, T. Komatsu, and Y. Nakagome, "Unsupervised training for deep speech source separation with Kullback-Leibler divergence based probabilistic loss function," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 56–60.
- [52] P. N. Petkov, V. Tsiaras, R. Doddipatla, and Y. Stylianou, "An unsupervised learning approach to neural-net-supported WPE dereverberation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 5761–5765.
- [53] L. Drude, J. Heymann, and R. Haeb-Umbach, "Unsupervised training of neural mask-based beamforming," in *Proc. Interspeech*, 2019, pp. 1253–1257.
- [54] Y. Bando, K. Sekiguchi, Y. Masuyama, A. A. Nugraha, M. Fontaine, and K. Yoshii, "Neural full-rank spatial covariance analysis for blind source separation," *IEEE Signal Process. Lett.*, vol. 28, pp. 1670–1674, 2021.
- [55] P. Wang and X. Li, "RVAE-EM: Generative speech dereverberation based on recurrent variational auto-encoder and convolutive transfer function," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2024, pp. 496–500.
- [56] N. Ito and M. Sugiyama, "Audio signal enhancement with learning from positive and unlabeled data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.
- [57] R. Talmon, I. Cohen, and S. Gannot, "Relative transfer function identification using convolutive transfer function approximation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 546–555, May 2009.
- [58] H. Sawada, N. Ono, H. Kameoka, D. Kitamura, and H. Saruwatari, "A review of blind source separation methods: Two converging routes to ILRMA originating from ICA and NMF," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–14, 2019.
- [59] Z.-Q. Wang, P. Wang, and D. Wang, "mComplex spectral mapping for single-and multi-channel speech enhancement and robust ASR," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1778–1787, 2020.
- [60] Z.-Q. Wang, P. Wang, and D. Wang, "Multi-microphone complex spectral mapping for utterance-wise and continuous speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2001–2014, 2021.
- [61] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 3, pp. 483–492, Mar. 2016.
- [62] D. S. Williamson and D. Wang, "Speech dereverberation and denoising using complex ratio masks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 5590–5594.
- [63] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, Oct. 2018.
- [64] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, "TF-GridNet: Integrating full- and sub-band modeling for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 3221–3236, 2023.
- [65] Z.-Q. Wang, G. Wichern, and J. L. Roux, "On the compensation between magnitude and phase in speech separation," *IEEE Signal Process. Lett.*, vol. 28, pp. 2018–2022, 2021.
- [66] Z.-Q. Wang, G. Wichern, and J. L. Roux, "Convolutional prediction for reverberant speech separation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2021, pp. 56–60.
- [67] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multi-talker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 10, pp. 1901–1913, Oct. 2017.
- [68] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 648–656.
- [69] M. I. Mandel, R. J. Weiss, and D. P. W. Ellis, "Model-based expectation-maximization source separation and localization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 2, pp. 382–394, Feb. 2010.
- [70] C. Boeddeker, F. Rautenberg, and R. Haeb-Umbach, "A comparison and combination of unsupervised blind source separation techniques," in *Proc. IEEE 14th ITG Conf. Speech Commun.*, 2021, pp. 1–5.
- [71] Z.-Q. Wang, G. Wichern, and J. L. Roux, "Leveraging low-distortion target estimates for improved speech enhancement," 2021, *arXiv:2110.00570*.
- [72] K. Kinoshita et al., "A summary of the REVERB challenge: State-of-the-art and remaining challenges in reverberant speech processing research," *Eurasip J. Adv. Signal Process.*, vol. 2016, no. 1, pp. 1–19, 2016.
- [73] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *Proc. IEEE ITG-Fachtagung Sprachkommunikation*, 2018, pp. 1–5.
- [74] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)-A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, pp. 749–752.
- [75] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep. 2011.
- [76] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR-half-baked or well done?," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 626–630.
- [77] T. Yoshioka et al., "VarArray: Array-geometry-agnostic continuous speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 6027–6031.
- [78] Z.-Q. Wang, "SuperM2M: Supervised and mixture-to-mixture co-learning for speech enhancement and robust ASR," 2024, *arXiv:2403.10271v2*.
- [79] Z.-Q. Wang, A. Kumar, and S. Watanabe, "Cross-talk reduction," in *Proc. IJCAI*, 2024, pp. 5171–5180.



Zhong-Qiu Wang received the B.E. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2013, and the M.S. and Ph.D. degrees in computer science and engineering from The Ohio State University, Columbus, OH, USA, in 2017 and 2020, respectively. From 2020 to 2021, he was a Visiting Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. From 2021 to 2024, he was a Postdoctoral Research Associate with Carnegie Mellon University, Pittsburgh, PA, USA. He is currently

a tenure-track Associate Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include speech separation, robust automatic speech recognition, microphone array processing, and deep learning, aiming at solving the cocktail party problem.