ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11464236

VM-UNSSOR: UNSUPERVISED NEURAL SPEECH SEPARATION ENHANCED BY HIGHER-SNR VIRTUAL MICROPHONE ARRAYS

Shulin He and Zhong-Qiu Wang

Department of Computer Science and Engineering
Southern University of Science and Technology, Shenzhen, China
{he.shulin96, wang.zhongqiu41}@gmail.com

ABSTRACT

Blind speech separation (BSS) aims to recover multiple speech sources from multi-channel, multi-speaker mixtures under unknown array geometry and room impulse responses. In unsupervised setup where clean target speech is not available for model training, UNSSOR proposes a mixture consistency (MC) loss for training deep neural networks (DNN) on over-determined training mixtures to realize unsupervised speech separation. However, when the number of microphones of the training mixtures decreases, the MC constraint weakens and the separation performance falls dramatically. To address this, we propose VM-UNSSOR, augmenting the observed training mixture signals recorded by a limited number of microphones with several higher-SNR virtual-microphone (VM) signals, which are obtained by applying linear spatial demixers (such as IVA and spatial clustering) to the observed training mixtures. As linear projections of the observed mixtures, the virtual-microphone signals can typically increase the SNR of each source and can be leveraged to compute extra MC losses to improve UNSSOR and address the frequency permutation problem in UNSSOR. On the SMS-WSJ dataset, in the over-determined six-microphone, two-speaker separation setup, VM-UNSSOR reaches 17.1 dB SI-SDR, while UNSSOR only obtains 14.7 dB; and in the determined two-microphone, two-speaker case, UNSSOR collapses to -2.7 dB SI-SDR, while VM-UNSSOR achieves 10.7 dB.

***Index Terms*—** Unsupervised neural speech separation

1. INTRODUCTION

The cocktail party problem [1–3] arises when several people speak at the same time in the same environment, wherein the microphones inevitably record a mixture of all the concurrent speech. This problem is widely encountered in applications such as smart speakers, smart cockpit (in electric vehicles), and wearable devices such as smart glasses [**?**, 1]. The goal of speech separation is to separate the mixture and recover each individual speaker signal so that downstream speech understanding applications such as automatic speech recognition, speaker identification, and hearing assistance can work robustly. Supervised speech separation based on deep neural networks (DNN) obtains impressive performance nowadays when the training and test conditions are matched with each other [4–13], yet the performance often drops dramatically in unseen acoustic environments. Meanwhile, collecting paired clean source signals and mixtures for every scenario is costly and in many cases impractical. These challenges motivated recent research on unsupervised speech separation (USS), which directly train DNNs on unlabeled mixtures recorded in the target environment via unsupervised learning [14–20].

UNSSOR [14], a recent algorithm in this line of research, proposes a so-called mixture consistency (MC) loss afforded by over-determined training mixtures to realize USS. The insight is that the source estimates, after being properly linearly filtered, should be able to reconstruct the observed mixture at each microphone. In other words, the mixture signal at each microphone can be leveraged as a constraint to regularize the source estimates, thereby realizing separation. In detail, during training, a DNN is trained to produce an estimate for each speaker, and, for the mixture signal at each microphone, the estimates of the speakers are linear-filtered and summed up to minimize the distance between the summated signal and the mixture signal (i.e., MC loss). This procedure yields supervision without requiring clean reference signals to penalize the DNN estimates [14]. Clearly, the supervision would become stronger when microphones outnumber speakers, because each additional microphone introduces one more MC constraint that could benefit training. For training mixtures recorded by microphone arrays with a limited number of microphones, the constraints are weaker and separation quality would degrade dramatically.

To address these limitations, we propose to introduce virtual microphones that increase the microphone count without using additional hardware. We denote $\mathcal{V}$ as the set of virtual microphones, derived by applying linear spatial demixers such as independent vector analysis (IVA) [21–24] or spatial clustering (SC) [25–28] to the observed mixture signals. Here, the term "virtual microphones" refers to linear projections of the recorded mixtures at the existing microphone positions. They introduce no new physical sensors and do not alter the array geometry. Because each virtual microphone in $\mathcal{V}$ is a linear projection of the observed mixture signals, it follows the same acoustic mixing model and can be used to compute additional MC loss for unsupervised training. On the other hand, as the linear spatial demixers are often effective to some extent, $\mathcal{V}$ often exhibit higher SNR for the sources. Such higher-SNR signals could act as a pseudo-teacher to help the model learn to separate the observed lower-SNR mixture signals, and, in addition, could help solve the frequency permutation problem [21], a unique issue that needs to be addressed in USS. We leverage the virtual-microphone signals to compute additional MC losses to penalize the DNN estimates, instead of having the estimates to directly fit the virtual-microphone signals, considering that doing so would limit the DNN's separation capability by that of the linear spatial demixers.

Linear spatial demixers such as IVA [21–24] often back-project separated signals to the reference microphone. This would skew the MC loss towards that microphone and degrades training stability.

This research was supported by National Key Research and Development Program of China (Grant No. 2025YFF0518003). *Corresponding author: Zhong-Qiu Wang*.

We mitigate this issue by back-projecting the separated estimates to every physical microphone. That is, for each physical microphone, the number of virtual microphones we create equals the number of sources. We then apply a re-weighted MC loss that balances the contributions of physical and virtual microphones. During training, we compute the MC loss on all microphones, physical and virtual. This increases the number of constraints. On the other hand, the separator takes as input the concatenated physical and virtual microphones.

We name the proposed system *VM-UNSSOR*. By injecting virtual microphones, determined mixtures become pseudo over-determined during training, and over-determined mixtures gain extra constraints. This way, we can keep the training of the system label-free and not requiring additional hardware microphones. The contributions of this work can be summarized as follows:

- We extend UNSSOR by introducing virtual microphones whose signals offer higher-SNR cues that strengthen the MC constraint.
- We show that a simple physical–virtual re-weighted MC loss enables unsupervised training on determined trainining mixtures by creating pseudo over-determined constraints.
- On the SMS-WSJ dataset [29], VM-UNSSOR achieves 17.1 dB SI-SDR in the 6-microphone, 2-speaker setup, compared with 14.7 dB obtained by UNSSOR; and in the determined 2-microphone, 2-speaker setup, UNSSOR fails to train (-2.7 dB SI-SDR) while VM-UNSSOR reaches 10.7 dB.

2. BACKGROUND

2.1. Notations and Speaker–Image Physical Model

We operate in the short-time Fourier transform (STFT) domain. Let $p \in \{1, \dots, P\}$ indexes P microphones, $c \in \{1, \dots, C\}$ indexes C speakers, t indexes T frames, and f indexes F frequency bins. UNSSOR models each microphone signal as a sum of speaker images:

$$Y_p(t,f) = \sum\nolimits_{c=1}^{C} X_p(c,t,f) + \varepsilon_p(t,f), \tag{1}$$

where $X_p(c)$ is the reverberant image of speaker c at microphone p, and ε_p absorbs residual noises. Without loss of generality, microphone 1 is designated as the reference microphone. The goal is to estimate the speaker images $\{X_1(c)\}_{c=1}^{C}$ at the reference microphone in an unsupervised manner while preserving their reverberation.

Let P denote the number of physical microphones, and we define the size of the full microphone index set as $\mathcal{P} = \{1, \dots, P\}$. Let $\mathcal{R} \subseteq \mathcal{P}$ denote the subset of physical microphones whose mixture signals are actually used as the input to the separator, and we denote its cardinality as P_r ($= |\mathcal{R}|$). Virtual microphones constructed by spatial demixers are collected in the set $\mathcal{V}$ with size Q ($= |\mathcal{V}|$). We use $\mathcal{U} = \mathcal{R} \cup \mathcal{V}$ as the combined input set, so $|\mathcal{U}| = P_r + Q$. In particular, we do not overload the notation P to mean $|\mathcal{U}|$ or P_r, and P always refers to the total number of physical microphones.

2.2. Relative RIR Constraint via Short Convolutions

For a small-aperture microphone array, the images of the same speaker at nearby microphones can be well-approximated by using a short linear filter between the microphones [30]. Let $p \in \{1, \dots, P\}$ index microphones and fix $p = 1$ as the reference. We first define a temporal context of the speaker image at the reference microphone

$$\tilde{X}_1(c,t,f) = \big[X_1(c,t-A,f), \dots, X_1(c,t+B,f)\big]^\top \in \mathbb{C}^E, \tag{2}$$

with $E = A + B + 1$. Then, for each non-reference microphone p (where $p \neq 1$) there exists a relative RIR $g_p(c,f) \in \mathbb{C}^E$ such that

$$X_p(c,t,f) \;\approx\; g_p(c,f)^{\mathsf{H}}\, \tilde{X}_1(c,t,f), \tag{3}$$

where $(\cdot)^{\mathsf{H}}$ computes Hermitian transpose. In other words, at each frequency f, $X_p(c,\cdot,f)$ can be modeled as a short convolution of the reference image $X_1(c,\cdot,f)$ and a relative RIR $g_p(c,f)$.

2.3. UNSSOR Training Mechanism

A neural separator g_θ is trained to output a complex-valued estimate $\hat{Z}(c)$ for each speaker c. Given $\hat{Z}(c)$ and the observed mixture signal Y_p, UNSSOR estimates the relative RIR by forward convolutive prediction (FCP) [14, 31–33]:

$$\hat{g}_p(c,f) = \operatorname*{argmin}_{g_p(c,f)} \sum_t \frac{\big|Y_p(t,f) - g_p(c,f)^{\mathsf{H}} \tilde{\hat{Z}}(c,t,f)\big|^2}{\hat{\lambda}_p(c,t,f)}, \tag{4}$$

where $\hat{\lambda}_p$ is a weighting term balancing the importance of each T-F unit and it is defined as $\hat{\lambda}_p(c,t,f) = \xi \cdot \texttt{max}(|Y_p|^2) + |Y_p(t,f)|^2$, with ξ flooring the denominator and $\texttt{max}(\cdot)$ extracting the maximum value of a spectrogram. Next, the FCP-estimated speaker image at microphone p is computed via

$$\hat{X}_p^{\text{FCP}}(c,t,f) = \hat{g}_p(c,f)^{\mathsf{H}} \tilde{\hat{Z}}(c,t,f). \tag{5}$$

UNSSOR [14] defines a label-free reconstruction loss named MC loss, which enforces consistency between the mixture signal and the summation of the FCP-estimated speaker images at each microphone. That is,

$$\mathcal{L}_{\text{MC}} = \sum\nolimits_{p=1}^{P} \mathcal{L}_{\text{MC},p}, \tag{6}$$

with $\mathcal{L}_{\text{MC},p}$ denoting the MC loss at microphone p and defined as

$$\begin{aligned}\mathcal{L}_{\text{MC},p} = \sum\nolimits_{t,f} \Big(& w_r \cdot \Big|\mathcal{R}\Big(Y_p(t,f) - \sum\nolimits_c \hat{X}_p^{\text{FCP}}(c,t,f)\Big)\Big| \\ & + w_i \cdot \Big|\mathcal{I}\Big(Y_p(t,f) - \sum\nolimits_c \hat{X}_p^{\text{FCP}}(c,t,f)\Big)\Big| \\ & + w_m \cdot \Big||Y_p(t,f)| - |\sum\nolimits_c \hat{X}_p^{\text{FCP}}(c,t,f)|\Big|\Big),\end{aligned} \tag{7}$$

where $|\cdot|$ extracts magnitude, $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ respectively extract real and imaginary components, and (w_r, w_i, w_m) are weighting terms controlling the contributions of the losses on the real component, imaginary component, and magnitude. We implement per-microphone energy normalization following UNSSOR [14].

Since $\hat{g}_p(c,f)$ is estimated independently per frequency, cross-frequency permutations can occur. To address this, UNSSOR includes an intra-source magnitude scattering (ISMS) loss to promote consistent spectral patterns across frequencies [14]:

$$\mathcal{L}_{\text{ISMS}} = \frac{\sum_t \frac{1}{C} \sum_{c=1}^{C} \texttt{var}\Big(\log(|\hat{X}_p^{\text{FCP}}(c,t,\cdot)|)\Big)}{\sum_t \texttt{var}\Big(\log(|Y_p(t,\cdot)|)\Big)}, \tag{8}$$

where $\texttt{var}(\cdot)$ computes the variance of the values in a vector.

3. VM-UNSSOR

Fig. 1 shows VM-UNSSOR, where a linear spatial demixer computes higher-SNR VM signals. We feed the concatenated physical and virtual microphone signals as inputs to the neural separator, and leverage FCP to enforce per-microphone mixture consistency.

3.1. Virtual Microphones from Linear Spatial Demixers

Let $\mathcal{R} = \{1, \dots, P_r\}$ denote the set of physical microphones used as the input to the DNN. We synthesize virtual microphones in the STFT domain by applying linear spatial demixers to the physical-array mixture signals. Concretely, we estimate a frequency-wise

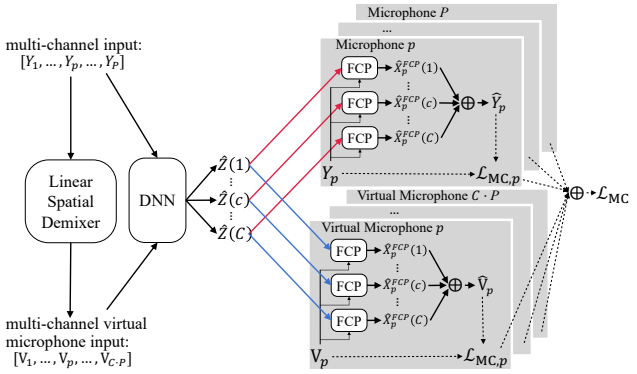

Fig. 1: Overview of VM-UNSSOR. A linear spatial demixer derives $\mathcal{V}$ via back-projection. The separator DNN takes the physical and virtual channels as input, using FCP and MC losses to enforce per-channel consistency.

demixing matrix $W(f) \in \mathbb{C}^{C \times P_r}$ using IVA on the raw mixtures and obtain separated components:

$$\hat{S}_c(t,f) = w_c(f)^{\mathsf{H}} Y_{\mathcal{R}}(t,f), \text{ for } c = 1, \ldots, C, \tag{9}$$

where $Y_{\mathcal{R}}(t,f) \in \mathbb{C}^{P_r}$ stacks the P_r physical microphones and $w_c(f)$ is the c-th row of $W(f)$. We then form a mixing estimate $A(f) \in \mathbb{C}^{P_r \times C}$ (i.e., the pseudo-inverse of $W(f)$) and back-project each separated component to every physical microphone:

$$V_{p,c}(t,f) = A_{p,c}(f)\, \hat{S}_c(t,f), \text{ for } p = 1, \ldots, P_r,\ c = 1, \ldots, C. \tag{10}$$

Each $V_{p,c}$ is a linear combination of the physical microphones. Therefore, it is consistent with the same acoustic mixing model as Y_p. This construction yields $Q = C \cdot P_r$ virtual microphones.

Let $\mathcal{V} = \{(p,c) : p \in \mathcal{R}, c \in \{1, \ldots, C\}\}$ denotes the set of virtual microphones, and let $\mathcal{U} = \mathcal{R} \cup \mathcal{V}$ be the augmented observation stack. We define

$$O_k(t,f) = \begin{cases} Y_k(t,f), & k \in \mathcal{R}, \\ V_{p,c}(t,f), & k = (p,c) \in \mathcal{V}, \end{cases} \tag{11}$$

and feed the physical and virtual (mixture) signals $\{O_k\}_{k \in \mathcal{U}}$ to the separator g_θ. The total number of input signals is $P_u = P_r + Q = P_r \times (1 + C)$.

3.2. Mixture Consistency Loss on Virtual Microphones

Let $\hat{Z}(c)$ be the estimate for source c produced by the neural separator g_θ based on the augmented mixture signals. For each microphone $k \in \mathcal{U}$ and frequency f, we estimate a microphone-specific relative filter $\hat{g}_k(c,f) \in \mathbb{C}^E$ by FCP:

$$\hat{g}_k(c,f) = \underset{g_k(c,f)}{\operatorname{argmin}} \sum_t \frac{\left| O_k(t,f) - g_k(c,f)^{\mathsf{H}} \tilde{\hat{Z}}(c,t,f) \right|^2}{\hat{\lambda}_k(c,t,f)}, \tag{12}$$

where $\hat{\lambda}_k(c,t,f) = \xi \cdot \max(|O_k|^2) + |O_k(t,f)|^2$ is a weighting term balancing the importance of each T-F unit, and $\tilde{\hat{Z}}(c,t,f)$ is the length-E temporal context of $\hat{Z}(c,t,f)$, constructed by following Eq. (2). The FCP-estimated image at microphone k is

$$\hat{X}_k^{\text{FCP}}(c,t,f) = \hat{g}_k(c,f)^{\mathsf{H}}\, \tilde{\hat{Z}}(c,t,f). \tag{13}$$

Aggregating constraints across $\mathcal{U}$ strengthens the overall MC constraint, and in addition leads to more stable FCP estimation. The training loss is defined as

$$\mathcal{L}_{\text{VM}} = \alpha \times \sum\nolimits_{k \in \mathcal{R}} \mathcal{L}_{\text{MC},k} + \beta \times \sum\nolimits_{k \in \mathcal{V}} \mathcal{L}_{\text{MC},k}, \tag{14}$$

where α and β are tunable weighting terms balancing the MC losses on the physical and virtual microphones.

4. EXPERIMENTAL SETUP

4.1. Dataset and Evaluation Setup

All experiments are based on the 2-speaker SMS-WSJ corpus [29]. We use the same training, validation, and test sets, room simulation settings, and STFT setup as in UNSSOR [14]. The UNSSOR baseline is trained and evaluated with 6 physical microphones. For VM-UNSSOR, we adopt $P_r = 6$ physical microphones and synthesize $Q = C \times P_r$ virtual microphones per mixture by frequency-wise linear spatial demixers, which yields $Q = 12$ and an augmented observation stack of $P_u = P_r + Q = 18$ signals for the 2-speaker case. We report averaged SDR [35], SI-SDR [36], NB-PESQ [37], STOI [38], and eSTOI [39] scores on the official test set. For the 2-microphone experiments, we use channels 0 and 3 from the 6-microphone recordings.

4.2. Baseline Systems

UNSSOR [14] trains the separator using a combination of the MC loss in Eq. (6) and ISMS loss in (8). It estimates per-frequency FCP filters from the mixtures and separator's outputs. We adopt its original setup for SMS-WSJ proposed in [14].

We utilize IVA and spatial clustering (SC) as the demixers. For IVA, we implement it with a Gaussian source model using the *torchiva* toolkit [40]. On over-determined arrays, we run a 3-source IVA and drop the lowest-energy estimate. On determined arrays we run a 2-source IVA. The STFT uses 256 ms window and 32 ms hop size. For SC, we use a public CACGMM implementation with inter-frequency correlation for frequency alignment [41]. Similarly to the IVA setup, on over-determined arrays we estimate 3 sources and drop the source with the lowest energy, and on determined arrays we estimate 2. The STFT uses 128 ms window and 16 ms hop size. Unless otherwise stated, VM-UNSSOR adopts IVA to form $Q = C \cdot P_r$ virtual microphones, which we found yields stable demixing on SMS-WSJ.

ArrayDPS [34] is a generative approach based on diffusion posterior sampling for USS. It combines a diffusion prior to leverage speech priors and a likelihood based on mixture consistency to satisfy signal regularizations enforced by observed mixtures. It uses IVA results to initialize its sampling process and reports results on SMS-WSJ under the same metrics used here. We cite its published SMS-WSJ configurations and scores for comparison in our tables.

The training procedure of VM-UNSSOR (e.g., learning-rate scheduling, optimizer, gradient clipping, and data augmentation) follows the UNSSOR recipe [14]. For VM-UNSSOR, in the reweighted loss described in Eq. (14), we use $\alpha = 1.0$ for physical microphones and $\beta = 0.02$ for virtual, unless otherwise noted. ξ in Eq. (4) and (12) is set to 10^{-4}.

5. EVALUATION RESULTS

Table 1 reports results with six physical microphones. Rows 0a and 1a report the results of the unprocessed mixtures and the demixer-only IVA output, and row 2a is the UNSSOR baseline. Adding only the virtual-microphone MC loss while keeping the separator input at six physical microphones (rows 2b/2c) increases SI-SDR from 14.7 to 14.9/15.3 dB and SDR from 15.5 to 15.7/16.2 dB (with/without using the ISMS loss, respectively). Feeding the physical and virtual microphone signals as inputs but without VM-loss (row 3a) further increases SI-SDR to 16.6 dB and SDR to 17.6 dB. Combining VM-input and VM-loss with the ISMS loss enabled (row 3b) yields 16.7 dB SI-SDR and 17.7 dB SDR. The best configuration is in row 3f,

Table 1: Results on SMS-WSJ (6-microphone, 2-speaker setup). "Input ch." are microphones fed to the separator. "VM-loss (ch.)" counts the number of channels used in the loss. "-" means no VM (β=0). "VM-input" uses $\mathcal{V}$ as additional inputs to the separator. α/β weight physical/virtual microphones. "ISMS" shows whether the ISMS loss is enabled. Virtual microphones are formed by IVA (Gaussian). In this result, C=2 and P_r=6, Q=12 and P_u=18.

Row	Systems	Input ch.	VM-loss (ch.)	α	β	ISMS	SI-SDR(dB)↑	SDR(dB)↑	NB-PESQ↑	STOI↑	eSTOI↑
0a	Mixture (unprocessed)	-	-	-	-	-	0.0	0.1	1.87	0.603	0.722
1a	Demixer-only baseline [24]	6	-	-	-	-	13.4	14.8	3.08	0.866	0.948
1b	ArrayDPS [34]	6	-	-	-	-	16.2	16.9	3.49	0.884	-
2a	UNSSOR [14]	6	-	1.0	-	✓	14.7	15.5	3.42	0.887	0.956
2b	UNSSOR + VM-loss	6	18	1.0	0.02	✓	14.9	15.7	3.50	0.893	0.958
2c	UNSSOR + VM-loss	6	18	1.0	0.02	×	15.3	16.2	3.49	0.902	0.963
3a	UNSSOR + VM-input	18	-	1.0	-	✓	16.6	17.6	3.55	0.912	0.966
3b	UNSSOR + VM-input + VM-loss	18	18	1.0	0.02	✓	16.7	17.7	3.57	0.914	0.967
3c	UNSSOR + VM-input + VM-loss	8	8	1.0	0.02	✓	15.5	16.4	3.52	0.906	0.965
3d	VM-UNSSOR	18	18	1.0	1.00	×	14.3	15.9	3.36	0.885	0.954
3e	VM-UNSSOR	18	18	1.0	0.06	×	16.8	17.8	3.58	0.915	0.967
3f	VM-UNSSOR	18	18	1.0	0.02	×	**17.1**	**18.0**	**3.59**	**0.918**	**0.969**

Table 2: Demixing method for virtual microphones on SMS-WSJ (6-microphone, 2-speaker setup). "Demixer-only" means using demixers alone.

Systems	Demixer	Input ch.	SI-SDR(dB)↑
Demixer-only baseline [24]	SC (6 mics)	-	7.4
Demixer-only baseline [24]	IVA (6 mics)	-	13.4
VM-UNSSOR	SC	18	16.9
VM-UNSSOR	IVA	18	**17.1**

where ISMS is disabled and $\beta = 0.02$, reaching 17.1 dB SI-SDR and 18.0 dB SDR. In comparison, ArrayDPS in row 1b attains 16.2 dB SI-SDR and 16.9 dB SDR. VM-UNSSOR surpasses its performance in the same six-microphone setting. All virtual microphones in Table 1 are formed by IVA with a Gaussian source model.

Row 2b vs. 2a shows that adding virtual microphones only to the loss brings gains without changing the inference input. Row 3a vs. 2a shows that providing virtual microphones as inputs is also helpful. Row 3b vs. 3a shows a further improvement when VM-loss is enabled after VM-input is already in place. Overall, virtual microphones enlarge the set of MC losses and can offer higher-SNR observations that the separator can benefit.

With six physical inputs and VM-loss, turning off ISMS (row 2c vs. 2b) gives a small improvement. With 18 inputs (physical + virtual) and VM-loss, turning off ISMS (row 3f vs. 3b) improves SI-SDR from 16.7 to 17.1 dB and SDR from 17.7 to 18.0 dB. A possible explanation is that virtual microphones can already provide source dominance and alignment cues that could help resolve the frequency permutation problem. In this case, including the ISMS loss could make the estimated magnitudes too uniform across frequencies, leaving the separator less able to correct demixer artifacts.

In row 3c, IVA on six microphones yields two source estimates that are both back-projected to the reference microphone 1, which biases the MC loss and hurts performance. In 3f, the same IVA outputs are back-projected to all the six microphones, preserving MC balance and giving better results.

Setting $\beta = 0.02$ (in row 3f) leads to the best performance. At $\beta = 0.06$ (in row 3e), both SI-SDR and SDR decline, and at $\beta = 1$ (in row 3d), they fall further. As β increases, the loss overweights the virtual microphones. Because they inherit demixer imperfections, the separator starts fitting demixing artifacts instead of enforcing mixture consistency on all channels.

Table 2 compares using IVA and spatial clustering as the linear demixers for forming virtual microphones. The demixer-only rows measure the demixers by themselves without VM-UNSSOR. IVA itself obtains 13.4 dB SI-SDR, which is higher than 7.4 dB for spatial clustering. For VM-UNSSOR, using an IVA frontend yields 17.1 dB SI-SDR, whereas using a spatial clustering frontend yields 16.9 dB. These results indicate that VM-UNSSOR is compatible with different demixers and that better demixing quality leads to better separation, since higher-SNR virtual microphones can strengthen mixture consistency and provide clearer source dominance cues for learning.

Table 3: Results on SMS-WSJ (2-microphone, 2-speaker setup). "Demixer-only" means using demixer alone. "Input ch.=6" means 2 physical plus 4 virtual microphones.

Systems	Demixer	Input ch.	SI-SDR(dB)↑
Demixer-only baseline [24]	SC (2 mics)	-	6.2
Demixer-only baseline [24]	IVA (2 mics)	-	9.1
UNSSOR	-	2	−2.7
VM-UNSSOR	SC	6	−0.8
VM-UNSSOR	IVA	6	**10.7**

Table 3 reports the determined two-microphone, two-speaker setting. UNSSOR fails to train and obtains −2.7 dB SI-SDR. The demixer-only rows show IVA at 9.1 dB and spatial clustering at 6.2 dB. With VM-UNSSOR, IVA-based virtual microphones achieve 10.7 dB SI-SDR using the same separator and pipeline. Replacing IVA with spatial clustering leads to failure (−0.8 dB). A possible explanation is that the spatial clustering demixer yields lower-quality virtual microphones, providing insufficient high-SNR cues to stabilize learning in this setup with a limited number of microphones.

6. CONCLUSION

We have proposed VM-UNSSOR, an unsupervised speech separation algorithm that augments the physical array with higher-SNR virtual microphones formed by linear spatial demixers. As linear projections of the observed mixtures, the virtual microphone signals satisfy mixture consistency and increase source dominance. By combining physical and virtual microphones and enforcing mixture consistency, VM-UNSSOR enlarges the number of constraints and remains effective in determined conditions. On SMS-WSJ, VM-UNSSOR clearly outperforms UNSSOR. In the 6-microphone, 2-speaker setup, it reaches 17.1 dB SI-SDR and 18.0 dB SDR. In the determined 2-microphone 2-speaker setup, UNSSOR fails to train (−2.7 dB SI-SDR), while VM-UNSSOR attains 10.7 dB. VM-UNSSOR requires no labeled sources and no additional hardware, making it attractive for rapid in-domain adaptation.

7. REFERENCES

[1] E. C. Cherry, "Some Experiments on The Recognition of Speech, with One and with Two Ears," *Journal of the acoustical society of America*, vol. 25, pp. 975–979, 1953.

[2] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, D. Wang and G. J. Brown, Eds. Hoboken, NJ: Wiley-IEEE Press, sep 2006.

[3] J. H. McDermott, "The Cocktail Party Problem," *Current Biology*, vol. 19, no. 22, pp. 1024–1027, 2009.

[4] 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, 2018.

[5] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep Clustering: Discriminative Embeddings for Segmentation and Separation," in *Proc. ICASSP*, 2016, pp. 31–35.

[6] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker 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, 2017.

[7] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai *et al.*, "Speakerbeam: Speaker Aware Neural Network for Target Speaker Extraction in Speech Mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[8] Z.-Q. Wang, K. Tan, and D. Wang, "Deep Learning Based Phase Reconstruction for Speaker Separation: A Trigonometric Perspective," in *Proc. ICASSP*, 2019, pp. 71–75.

[9] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 27, no. 8, pp. 1256–1266, 2019.

[10] Y. Liu and D. Wang, "Divide and Conquer: A Deep CASA Approach to Talker-independent Monaural Speaker Separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 27, no. 12, pp. 2092–2102, 2019.

[11] Z.-Q. Wang and D. Wang, "Combining Spectral and Spatial Features for Deep Learning Based Blind Speaker Separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 27, no. 2, pp. 457–468, 2018.

[12] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-Independent Speech Separation with Deep Attractor Network," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 26, no. 4, pp. 787–796, 2018.

[13] W. Zhang, K. Saijo, Z.-Q. Wang, S. Watanabe *et al.*, "Toward Universal Speech Enhancement For Diverse Input Conditions," in *Proc. ASRU*, 2023, pp. 1–6.

[14] Z.-Q. Wang and S. Watanabe, "UNSSOR: Unsupervised Neural Speech Separation by Leveraging Over-determined Training Mixtures," in *Proc. NeurIPS*, vol. 36, 2023, pp. 34 021–34 042.

[15] K. Saijo and T. Ogawa, "Self-Remixing: Unsupervised Speech Separation via Separation and Remixing," in *Proc. ICASSP*, 2023, pp. 1–5.

[16] Y. Wang, J. Du, L.-R. Dai, and C.-H. Lee, "A gender mixture detection approach to unsupervised single-channel speech separation based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 7, pp. 1535–1546, 2017.

[17] A. Sivaraman, S. Wisdom, H. Erdogan, and J. R. Hershey, "Adapting Speech Separation to Real-World Meetings using Mixture Invariant Training," in *Proc. ICASSP*, 2022, pp. 686–690.

[18] K. Saijo and T. Ogawa, "Unsupervised Training of Sequential Neural Beamformer using Coarsely-Separated and Non-Separated Signals," in *Proc. Interspeech*, 2022, pp. 251–255.

[19] E. Tzinis, S. Venkataramani, and P. Smaragdis, "Unsupervised Deep Clustering for Source Separation: Direct Learning from Mixtures Using Spatial Information," in *Proc. ICASSP*, 2019, pp. 81–85.

[20] L. Drude, D. Hasenklever, and R. Haeb-Umbach, "Unsupervised Training of a Deep Clustering Model for Multichannel Blind Source Separation," in *Proc. ICASSP*, 2019, pp. 695–699.

[21] H. Sawada, N. Ono, H. Kameoka, D. Kitamura *et al.*, "A Review of Blind Source Separation Methods: Two Converging Routes to ILRMA Originating from ICA and NMF," *APSIPA Trans. Signal, Information. Process*, vol. 8, p. e12, 2019.

[22] T. Kim, T. Eltoft, and T.-W. Lee, "Independent Vector Analysis: An Extension of ICA to Multivariate Components," in *Proc. ICA*, 2006, pp. 165–172.

[23] A. Hiroe, "Solution of permutation problem in frequency domain ica, using multivariate probability density functions," in *Proc. ICA*, 2006, pp. 601–608.

[24] C. Boeddeker, F. Rautenberg, and R. Haeb-Umbach, "A Comparison and Combination of Unsupervised Blind Source Separation Techniques," *Speech Communication*, pp. 1–5, 2021.

[25] S. Rickard, "The duet blind source separation algorithm," in *Blind speech separation*, 2007, pp. 217–241.

[26] D. H. T. Vu and R. Haeb-Umbach, "Blind Speech Separation Employing Directional Statistics in an Expectation Maximization Framework," in *Proc. ICASSP*, 2010, pp. 241–244.

[27] H. Sawada, S. Araki, and S. Makino, "Underdetermined Convolutive Blind Source Separation via Frequency Bin-Wise Clustering and Permutation Alignment," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 19, no. 3, pp. 516–527, 2010.

[28] N. Ito, S. Araki, and T. Nakatani, "Complex Angular Central Gaussian Mixture Model for Directional Statistics in Mask-based Microphone Array Signal Processing," in *Proc. EUSIPCO*, 2016, pp. 1153–1157.

[29] L. Drude, J. Heitkaemper, C. Boeddeker, and R. Haeb-Umbach, "Smswsj: Database, performance measures, and baseline recipe for multi-channel source separation and recognition," in *arXiv:1910.13934*, 2019.

[30] 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, 2017.

[31] Z.-Q. Wang, "SuperM2M: Supervised and Mixture-to-mixture Co-learning for Speech Enhancement and Noise-robust ASR," *Neural Networks*, vol. 188, p. 107408, 2025.

[32] ——, "Usdnet: Unsupervised speech dereverberation via neural forward filtering," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 3882–3895, 2024.

[33] Z.-Q. Wang, A. Kumar, and S. Watanabe, "Cross-Talk Reduction," in *Proc. IJCAI*, 8 2024, pp. 5171–5180.

[34] Z. Xu, X. Fan, Z.-Q. Wang, X. Jiang, and R. Roy Choudhury, "ArrayDPS: Unsupervised blind speech separation with a diffusion prior," in *Proc. ICML*, vol. 267, 2025, pp. 69 160–69 188.

[35] E. Vincent, R. Gribonval, and C. Févotte, "Performance Measurement in Blind Audio Source Separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.

[36] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR - Half-Baked or Well Done?" in *Proc. ICASSP*, 2019, pp. 626–630.

[37] 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. ICASSP*, vol. 2, 2001, pp. 749–752.

[38] 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, 2011.

[39] J. Jensen and C. H. Taal, "An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 11, pp. 2009–2022, 2016.

[40] R. Scheibler and K. Saijo, "https://github.com/fakufaku/torchiva," 2022.

[41] C. Boeddeker, "https://github.com/fgnt/pb_bss/blob/master/examples /mixture_model_example.ipynb," 2019.