PROXIMAL DEEP RECURRENT NEURAL NETWORK FOR MONAURAL SINGING VOICE SEPARATION

Weitao Yuan[⋆] *Shengbei Wang*[⋆*] *Xiangrui Li*[⋆] *Masashi Unoki* [†] *Wenwu Wang*[††]

[⋆] Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems,
Tianjin Polytechnic University, China
[†]Japan Advanced Institute of Science and Technology, Japan
[††]University of Surrey, UK

ABSTRACT

The recent deep learning methods can offer state-of-the-art performance for Monaural Singing Voice Separation (MSVS). In these deep methods, the recurrent neural network (RNN) is widely employed. This work proposes a novel type of Deep RNN (DRNN), namely Proximal DRNN (P-DRNN) for MSVS, which improves the conventional Stacked RNN (S-RNN) by introducing a novel interlayer structure. The interlayer structure is derived from an optimization problem for Monaural Source Separation (MSS). Accordingly, this enables a new hierarchical processing in the proposed P-DRNN with the explicit state transfers between different layers and the skip connections from the inputs, which are efficient for source separation. Finally, the proposed approach is evaluated on the MIR-1K dataset to verify its effectiveness. The numerical results show that the P-DRNN performs better than the conventional S-RNN and several recent MSVS methods.

Index Terms— Proximal Algorithm, Monaural Source Separation, Recurrent Neural Network

1. INTRODUCTION

Monaural Singing Voice Separation (MSVS), as an important examplar of Monaural Source Separation (MSS), aims to separate the singing voice (vocal) from the background music components in a single channel mixture signal. Compared to traditional shallow methods, deep learning methods such as Deep Neural Network (DNN) [1, 2] have recently emerged as powerful alternatives and provided state-of-the-art performance for MSVS with the help of large datasets. There are three basic structures to construct DNN for MSVS: (i) Feed-Forward Network (FFN) (e.g., [3]); (ii) Convolutional Neural Network (CNN) (e.g., [4, 5]); (iii) Recurrent Neural Network (RNN) (e.g., [6]). The advantage of employing deep methods is built on a hypothesis that "a deep, hierarchical model can be exponentially more efficient at representing some functions than a shallow one" [7]. This research concentrates on constructing a more effective deep architecture of RNNs for MSVS.

RNN can learn the temporal dynamics in audio signals, thanks to the recurrent (feedback) connections between the hidden units. However, the recurrent connections in RNN offer deep structures only in time [8], and lack hierarchical processing of the input at different scales [9]. To address this problem, Deep Recurrent Neural Network (DRNN) is proposed [10, 11], such as the Stacked RNN (S-RNN), which stacks multiple recurrent hidden layers on top of each other [12–14]. However, the connection ('stacking') between layers of S-RNN is shallow [11], without intermediate, nonlinear hidden layers (interlayers) between different layers. Here, we introduce an improved S-RNN, namely Proximal-DRNN (P-DRNN) for MSVS, which has a novel interlayer (Proximal Layer) between different layers of RNNs, as illustrated in Fig. 1. The proposed interlayer architecture (Eqs. (3)-(5)) is derived from a proximal algorithm [15, 16] designed to solve a general MSS optimization problem. This design introduces two new structures (illustrated by the blue dotted lines in Fig. 1), which are formulated in Eq. (5): (i) explicit state transfers between different Proximal Layers; (ii) 'skip' connections from the inputs to each Proximal Layer. These two structures are customized for MSS and can deepen RNNs effectively for MSVS. It is found that some previous works [17, 18] have proposed to design the architecture of deep networks via proximal algorithms. However, none of these previous works considered how to effectively deepen RNNs. Although [11] explored different ways to extend a RNN to DRNNs, they did not focus on constructing the interlayer structures between different RNN layers.

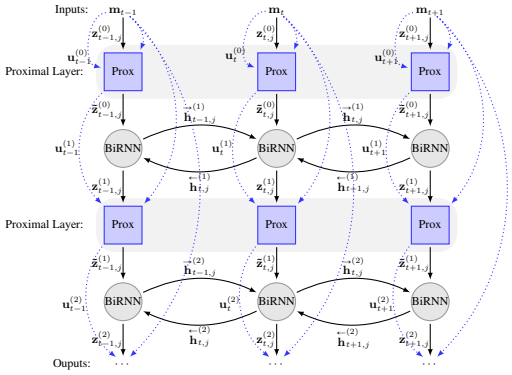

Fig. 1. The proposed P-DRNN made of alternating layers of Bidirectional RNN (BiRNN) Layer and Proximal (Prox) Layer.

2. PROPOSED METHOD

One of the most widely-used strategies for MSVS employs the source-dependent filters, which are applied to the mixture signal in the Short-Time Fourier Transform (STFT) domain [19–21]. The source-dependent filtering is usually based on the Time-Frequency (T-F) mask, which can be learned from the DRNN. In this work, we use the T-F masking framework illustrated in Fig. 2 to evaluate the performance of the conventional S-RNN and the proposed P-DRNN.

*Thanks to Natural Science Foundation of Tianjin (No. 17JCQNJC00100), the Science&Technology Development Fund of Tianjin Education Commission for Higher Education (No. 2017KJ089 and No. 2018KJ218), National Natural Science Foundation of China (No. 6137104), and the Program for Innovative Research Team in University of Tianjin (No. TD13-5032).

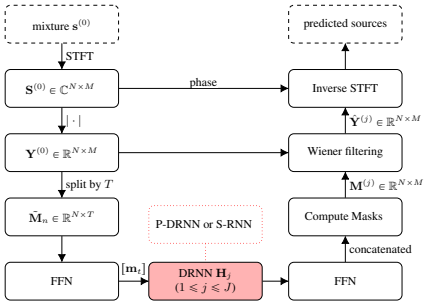

Fig. 2. The T-F masking framework, where DRNN $\mathbf{H}_j$ can be implemented with P-DRNN or S-RNN.

That is, we employ these two models to implement the DRNN (denoted as $\mathbf{H}_j$) in Fig. 2. By testing different depths, we can compare their performance under the same T-F masking framework.

2.1. The overall T-F masking framework

Let $\mathbf{s}^{(0)}$ be the time-domain mixture signal made of J target sources $\mathbf{s}^{(j)}$ $(1 \leqslant j \leqslant J)$,

$$\mathbf{s}^{(0)} = \mathbf{s}^{(1)} + \mathbf{s}^{(2)} + ... + \mathbf{s}^{(J)}.$$

At first, we compute $\mathbf{S}^{(k)} \in \mathbb{C}^{N\times M}$, which is the complex-valued STFT representation of $\mathbf{s}^{(k)}$ $(0 \leqslant k \leqslant J)$, comprising of M time frames and N frequency bins (for testing, we have only $\mathbf{S}^{(0)}$). Then we compute

$$\mathbf{Y}^{(k)}(\stackrel{\text{def}}{=}[\mathbf{y}_t^{(k)}]_{1\leqslant t\leqslant M}) = |\mathbf{S}^{(k)}| \in \mathbb{R}^{N\times M}, (0 \leqslant k \leqslant J),$$

with $|\cdot|$ denoting the element-wise magnitude spectrogram of the matrix (for testing, we use only $\mathbf{Y}^{(0)}$). Suppose T is an integer indicating the amount of time frames used for each processing. Without loss of generality, assume that $B = M/T$ is an integer (if not, $\mathbf{Y}^{(0)}$ can be zero-padded). Then $\mathbf{Y}^{(0)}$ is splitted into B subsequences,

$$\begin{aligned}\mathbf{Y}^{(0)} &= [\tilde{\mathbf{M}}_1, ..., \tilde{\mathbf{M}}_B],\\ \tilde{\mathbf{M}}_n &= [\mathbf{y}^{(0)}_{(n-1)*T+1}, \cdots, \mathbf{y}^{(0)}_{n*T}] \in \mathbb{R}^{N\times T}, \quad 1 \leqslant n \leqslant B.\end{aligned}$$

Then each $\tilde{\mathbf{M}}_n$ $(1 \leqslant n \leqslant B)$ is fed into a FFN layer with shared weights through time frames,

$$\mathbf{M}_n = [\mathbf{m}_{n,t}]_{1\leqslant t\leqslant T} = \text{ReLU}(\mathbf{W}^{(0)}\tilde{\mathbf{M}}_n + \mathbf{b}^{(0)}) \in \mathbb{R}^{N\times T}, \quad (1)$$

where ReLU is the element-wise rectified linear unit function. The DRNN $\mathbf{H}_j$ $(1 \leqslant j \leqslant J)$ which has L layers takes the input $\mathbf{M}_n$ to create T-F masks. Firstly, each $\mathbf{H}_j$ takes each $\mathbf{M}_n$ and outputs a prediction $\breve{\mathbf{Y}}_n^{(j)}$ $(1 \leqslant n \leqslant B)$,

$$\breve{\mathbf{Y}}_n^{(j)} = \mathbf{H}_j(\mathbf{M}_n) \in \mathbb{R}^{N\times T}, 1 \leqslant j \leqslant J, 1 \leqslant n \leqslant B.$$

Secondly, $\breve{\mathbf{Y}}_n^{(j)}$s are fed into FFN layers to output $\tilde{\mathbf{Y}}_n^{(j)}$,

$$\tilde{\mathbf{Y}}_n^{(j)} = \text{ReLU}(\mathbf{W}_j^{(L+1)}\breve{\mathbf{Y}}_n^{(j)} + \mathbf{b}_j^{(L+1)}) \in \mathbb{R}^{N\times T}.$$

By concatenating $\tilde{\mathbf{Y}}_n^{(j)}$ $(1 \leqslant n \leqslant B)$ together, we obtain $\tilde{\mathbf{Y}}^{(j)} = [\tilde{\mathbf{Y}}_1^{(j)}, ..., \tilde{\mathbf{Y}}_B^{(j)}] \in \mathbb{R}^{N\times M}$. At last, the estimation $\hat{\mathbf{Y}}^{(j)}$ for the target $\mathbf{Y}^{(j)}$ is achieved by the soft-ratio mask or 1-Wiener filter [21], that is, multiplying a soft mask $\mathbf{M}^{(j)} \in \mathbb{R}_+^{N\times M}$ (the j-th source-dependent filter/mask) element-wise with the mixture magnitude spectrogram $\mathbf{Y}^{(0)}$,

$$\hat{\mathbf{Y}}^{(j)}(\stackrel{\text{def}}{=}[\hat{\mathbf{y}}_t^{(j)}]_{1\leqslant t\leqslant M}) = \mathbf{Y}^{(0)} \odot \mathbf{M}^{(j)}, \mathbf{M}^{(j)} = \frac{|\tilde{\mathbf{Y}}^{(j)}|}{\sum_{j=1}^{J}|\tilde{\mathbf{Y}}^{(j)}| + \epsilon}, \quad (2)$$

where $\odot$ is element-wise product, the fraction bar denotes the element-wise matrix division and $\epsilon > 0$ is a small floating point number preventing the zero denominator of $\mathbf{M}^{(j)}$. The final outputs of the masking framework are the time-domain predicted sources obtained by the STFT synthesis operation of $\hat{\mathbf{Y}}^{(j)}$ along with the original mixture phase spectrum of $\mathbf{S}^{(0)}$. For the MSVS task, we need to separate two sources, that is, $J = 2$. For training, the L_2 loss function is employed [9], $L_2 = \|\hat{\mathbf{y}}_t^{(1)} - \mathbf{y}_t^{(1)}\|_2^2 + \|\hat{\mathbf{y}}_t^{(2)} - \mathbf{y}_t^{(2)}\|_2^2$. The key part in Fig. 2 is the DRNN $\mathbf{H}_j$. In the following, we will describe the implementation of $\mathbf{H}_j$ based on P-DRNN and S-RNN, respectively.

2.2. P-DRNN for $\mathbf{H}_j$

Since the inputs $\mathbf{M}_n = [\mathbf{m}_{n,t}] = [\mathbf{m}_t]_{1\leqslant t\leqslant T}$[1] for $\mathbf{H}_j$ are T-F representations, in order to compute effective masks, $\mathbf{H}_j$ based on P-DRNN is designed to

(i) learn the temporal inter-dependencies of different $\mathbf{m}_t$s;

(ii) have the ability to convey information of the mixture $\mathbf{m}_t$ between different layers via interlayers for better MSS effect.

Base on these considerations, the proposed P-DRNN is constructed with two basic sub-layers: Bidirectional RNN (BiRNN) Layer for purpose (i) and Proximal Layer for purpose (ii). These two sub-layers are placed alternately in P-DRNN. As shown in Fig. 1, the i-th layer of P-DRNN [2] with outputs $\mathbf{z}_{t,j}^{(i)}$ is defined as,

(1) Proximal Layer

$$\mathbf{z}_{t,j}^{(i-1/2)} = \text{ReLU}(\mathbf{O}_j^{(i)}(\mathbf{z}_{t,j}^{(i-1)} - \tau\mathbf{u}_t^{(i-1)}) + \mathbf{d}_j^{(i)}) \quad (3)$$

$$\tilde{\mathbf{z}}_{t,j}^{(i-1)} = \mathbf{z}_{t,j}^{(i-1)} + \rho_i(\mathbf{z}_{t,j}^{(i-1/2)} - \mathbf{z}_{t,j}^{(i-1)}) \quad (4)$$

$$\mathbf{u}_t^{(i)} = \mathbf{u}_t^{(i-1)} + \frac{\rho_i\sigma}{N}(\sum_{j=1}^{J}(2\mathbf{z}_{t,j}^{(i-1/2)} - \mathbf{z}_{t,j}^{(i-1)}) - \mathbf{m}_t) \quad (5)$$

(2) BiRNN Layer:

$$\overleftarrow{\mathbf{h}}_{t,j}^{(i)} = \mathcal{H}\left(\overleftarrow{\mathbf{W}}_j^{(i)}\tilde{\mathbf{z}}_{t,j}^{(i-1)} + \overleftarrow{\mathbf{V}}_j^{(i)}\overleftarrow{\mathbf{h}}_{t+1,j}^{(i)} + \overleftarrow{\mathbf{b}}_j^{(i)}\right), \quad (6)$$

$$\overrightarrow{\mathbf{h}}_{t,j}^{(i)} = \mathcal{H}\left(\overrightarrow{\mathbf{W}}_j^{(i)}\tilde{\mathbf{z}}_{t,j}^{(i-1)} + \overrightarrow{\mathbf{V}}_j^{(i)}\overrightarrow{\mathbf{h}}_{t-1,j}^{(i)} + \overrightarrow{\mathbf{b}}_j^{(i)}\right), \quad (7)$$

$$\mathbf{z}_{t,j}^{(i)} = \text{ReLU}(\mathbf{U}_j^{(i)}[\overleftarrow{\mathbf{h}}_{t,j}^{(i)}; \overrightarrow{\mathbf{h}}_{t,j}^{(i)}] + \mathbf{c}_j^{(i)}). \quad (8)$$

On one hand, BiRNN [22] Layer contains two hidden sublayers, one for the left-to-right propagation and the other for the right-to-left propagation. $\mathcal{H}$ is the hidden layer activation function and we adopt ReLU in this work. The input $\tilde{\mathbf{z}}_{t,j}^{(i-1)}$ of BiRNN is from the previous Proximal Layer. The output $\mathbf{z}_{t,j}^{(i)}$ is fed to the next Proximal Layer.

On the other hand, Proximal Layer is a novel structure with input $\mathbf{z}_{t,j}^{(i-1)}$ and output $\tilde{\mathbf{z}}_{t,j}^{(i-1)}$, which connects two sequential BiRNN layers when $i > 1$ or the input $\mathbf{m}_t$ and the first BiRNN layer when $i = 1$. Specifically, Eqs. (3)-(5) are based on solving a MSS optimization model with variable $\mathbf{x}_{t,j}$, which corresponds to the j-th estimated source from the mixture $\mathbf{m}_t$,

$$\begin{aligned}\underset{\mathbf{x}_{t,j}}{\text{minimize}} \quad & \phi_1(\mathbf{x}_{t,1}) + \phi_2(\mathbf{x}_{t,2}) + ... + \phi_J(\mathbf{x}_{t,J})\\ \text{subject to} \quad & \sum_{j=1}^{J}\mathbf{x}_{t,j} = \mathbf{m}_t.\end{aligned} \quad (9)$$

[1] Without loss of generality, we omit index n for simplicity.

[2] In the formulation of P-DRNN, $\mathbf{z}_{t,j}^{(0)} = \mathbf{u}_t^{(0)} = \mathbf{m}_t$, $1\leqslant i\leqslant L$ represents the i-th layer, $1\leqslant j\leqslant J$ represents the j-th source, $1\leqslant t\leqslant T$ is the time index, $\overleftarrow{\mathbf{W}}_j^{(i)}$, $\overrightarrow{\mathbf{W}}_j^{(i)}$, $\overleftarrow{\mathbf{V}}_j^{(i)}$, $\overrightarrow{\mathbf{V}}_j^{(i)}$, $\mathbf{U}_j^{(i)}$ and $\mathbf{O}_j^{(i)}$ are matrices of trainable parameters that represent the connection weights of the network and $\overleftarrow{\mathbf{b}}_j^{(i)}$, $\overrightarrow{\mathbf{b}}_j^{(i)}$, $\mathbf{d}_j^{(i)}$ and $\mathbf{c}_j^{(i)}$ are vectors of trainable bias parameters, and ρ_i and σ are trainable scalars.

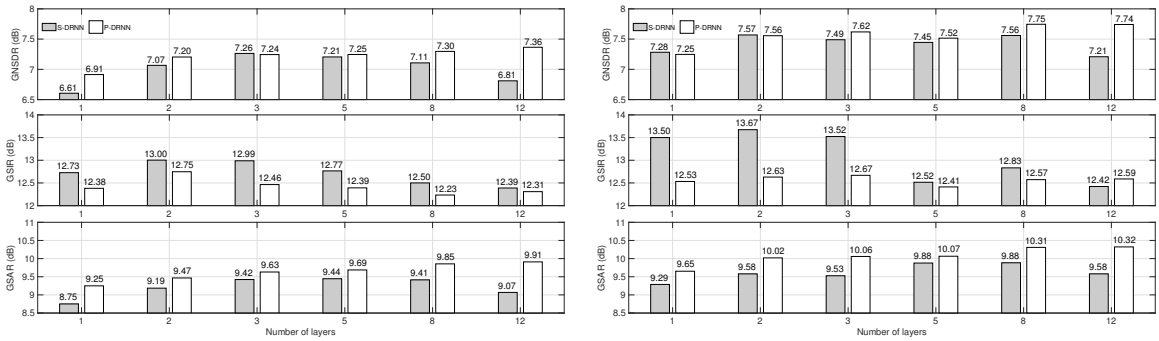

Fig. 3. The separation performances of S-RNN and P-DRNN: the left figure is for $T = 4$ and the right figure is for $T = 10$.

The goal of Eq. (9) is to decompose each frequency feature vector $\mathbf{m}_t$ into $\mathbf{x}_{t,j}$. Since there are infinite candidate solutions to satisfy the additive constraint[3], the objective function employs the signal prior $\phi_j(\cdot)$ to penalize $\mathbf{x}_{t,j}$ for the j-th source. If the minimizers of Eq. (9) exist and each prior ϕ_j is convex, Eq. (9) can be solved by proximal algorithms [15, 16]. Here we adopt the popular primal-dual method [23–25]. All details are in the appendix. The derived iterative algorithm for Eq. (9) is,

$$\mathbf{x}_{t,j}^{k-1/2} \leftarrow \text{Prox}_{\tau\phi_j}(\mathbf{x}_{t,j}^{k-1} - \tau\mathbf{u}_t^{k-1}), \quad (10)$$

$$\mathbf{x}_{t,j}^{k} \leftarrow \mathbf{x}_{t,j}^{k-1} + \rho_k(\mathbf{x}_{t,j}^{k-1/2} - \mathbf{x}_{t,j}^{k-1}), \quad (11)$$

$$\mathbf{u}_t^{k} \leftarrow \mathbf{u}_t^{k-1} + \frac{\rho_k\sigma}{N}\sum_{j=1}^{J}\left(2\mathbf{x}_{t,j}^{k-1/2} - \mathbf{x}_{t,j}^{k-1}\right) - \frac{\rho_k\sigma}{N}\mathbf{m}_t, \quad (12)$$

where k represents the k-th iteration step, $\text{Prox}_{\tau\phi_j}$ is the proximal operators of ϕ_j (see Eq. (1.1) in [15]), and ρ_k, τ, and σ are positive [23–25]. If we directly apply the iterative algorithm of Eqs. (10)-(12) to solve Eq. (9), the proximal operator $\text{Prox}_{\tau\phi_j}$ needs to be given analytically. However, due to the diversity of different sources, they cannot be given and it is better to directly learn $\text{Prox}_{\tau\phi_j}$ from the dataset. Thus instead of using Eqs. (10)-(12) directly, we employ it to inspire a novel deep structure, where the $\text{Prox}_{\tau\phi_j}$ operator in Eq. (10) is approximated with one layer of FFN. The final architecture of the Proximal Layer in Eqs. (3)-(5) is constructed by imitating the data flow of one-loop iteration of the proximal algorithm in Eqs. (10)-(12).

It is worthwhile to note that although the proposed interlayer architecture in Eqs. (3)-(5) is inspired from a proximal algorithm, it forms a typical RNN structure between the corresponding units of different proximal layers. As shown in Fig. 1, for the i-th Proximal Layer, the primal variable $\mathbf{z}_{t,j}^{(i-1)}$ and $\tilde{\mathbf{z}}_{t,j}^{(i-1)}$ are the input and output respectively. The dual variable $\mathbf{u}_t^{(i)}$ is the 'state' of the i-th Proximal Layer, because it keeps track of the error to the additive constraint before the i-th layer so that the next layer output does not go far away from the constraint. Different from the recurrent states $\overleftarrow{\mathbf{h}}_{t,j}^{(i)}$ and $\overrightarrow{\mathbf{h}}_{t,j}^{(i)}$ of the BiRNN Layer that collect information from the same layer in time, the state $\mathbf{u}_t^{(i)}$ of the Proximal Layer gathers information from the state $\mathbf{u}_t^{(i-1)}$ of previous layers (see Eq. (5)). Besides, our design also introduces 'skip' connection from the inputs $\mathbf{m}_t$ to each Proximal Layer (see Eq. (5)). Both the explicit state transfers between different Proximal Layers and 'skip' connections from the inputs are useful for improving the separation performance of P-DRNN.

[3]The constraint of the optimization problem in Eq. (9) is additive which is only an approximation, since the sum of each source magnitude does not coincide with that of mixture owing to the phase, except that all sources are uncorrelated [32]. However, because we do not directly use this formulation to solve MSVS task, but only use it to inspire the interlayer architecture of P-DRNN, this approximation is acceptable.

2.3. S-RNN for $\mathbf{H}_j$

As a baseline to be compared with the proposed P-DRNN, we also implement the S-RNN[4] for $\mathbf{H}_j$ with outputs $\mathbf{z}_{t,j}^{(L)}$, defined as,

$$\overleftarrow{\mathbf{h}}_{t,j}^{(i)} = \mathcal{H}\left(\overleftarrow{\mathbf{P}}_j^{(i)}\,\mathbf{h}_{t,j}^{(i-1)} + \overleftarrow{\mathbf{Q}}_j^{(i)}\,\overleftarrow{\mathbf{h}}_{t+1,j}^{(i)} + \overleftarrow{\mathbf{d}}_j^{(i)}\right), \quad (13)$$

$$\overrightarrow{\mathbf{h}}_{t,j}^{(i)} = \mathcal{H}\left(\overrightarrow{\mathbf{P}}_j^{(i)}\,\mathbf{h}_{t,j}^{(i-1)} + \overrightarrow{\mathbf{Q}}_j^{(i)}\,\overrightarrow{\mathbf{h}}_{t-1,j}^{(i)} + \overrightarrow{\mathbf{d}}_j^{(i)}\right), \quad (14)$$

$$\mathbf{h}_{t,j}^{(i)} = [\overleftarrow{\mathbf{h}}_{t,j}^{(i)}, \overrightarrow{\mathbf{h}}_{t,j}^{(i)}], \quad (15)$$

$$\mathbf{z}_{t,j}^{(L)} = \text{ReLU}(\mathbf{W}_j^{(L)}\mathbf{h}_{t,j}^{(L)} + \mathbf{d}^{(L)}) \quad (16)$$

3. EVALUATION

The T-F masking framework was evaluated on the MIR-1K dataset [26]. The goal was to separate singing voice from music recordings. The magnitude spectrum was obtained by applying 1024-point STFT with a hop size of 512, which was the same as in [9]. We also used the same training and testing set with [9] for fair comparison: in the MIR-1K dataset, 175 clips performed by one male "abjones" and one female singer "amy" were used as the training set and the other 825 clips performed by 17 singers were used for testing. The mixture signals were produced by mixing the vocal and background music components at signal-to-noise ratio (SNR) of 0 dB. The separation performance was measured by BSS-EVAL toolkit [27] with respect to three criteria, i.e., source-to-distortion ratio (SDR), source-to-interferences ratio (SIR), and sources-to-artifacts ratio (SAR). Normalized SDR (NSDR) was calculated to show the improvement of SDR compared to the original mixture. The final Global SIR (GSIR), Global SAR (GSAR), and Global NSDR (GNSDR) results were computed by taking the average of all test clips and weighted by their lengths.

Figure 3 presents the vocal separating performance of both P-DRNN and S-RNN for various depths L with respect to different T. Based on the 'deep' hypothesis, we would expect that a deeper S-RNN should be more effective for MSVS. However, as shown in Fig. 3, when the number of layers is increased to more than 3, the S-RNN experienced a rapid performance decrease. For example, it can be seen that when $T = 4$ (the left panel in Fig. 3), the GNSDR of S-RNN dropped from 7.26 dB (3-layer) to 6.81 dB (12-layer). The similar phenomena could be observed for $T = 10$. However for P-DRNN, we can see that its performances of GNSDR and GSAR for

[4]In the formulation of S-RNN, $\mathbf{h}_{t,j}^{(0)}=\mathbf{m}_t$, $1\leqslant i\leqslant L$, $1\leqslant j\leqslant J$, $1\leqslant t\leqslant T$, $\overleftarrow{\mathbf{P}}_j^{(i)}$, $\overrightarrow{\mathbf{P}}_j^{(i)}$, $\overleftarrow{\mathbf{Q}}_j^{(i)}$, $\overrightarrow{\mathbf{Q}}_j^{(i)}$, $\mathbf{W}_j^{(L)}$, $\overleftarrow{\mathbf{d}}_j^{(i)}$, $\overrightarrow{\mathbf{d}}_j^{(i)}$, $\mathbf{d}^{(L)}$ are trainable parameters.

Table 1. Comparisons of the separation results (in dB) between the proposed method (12-layer) and previous approaches.

Unsupervised			
Model	GNSDR (dB)	GSIR (dB)	GSAR(dB)
RPCA [28]	3.15	4.43	11.09
RPCAh [29]	3.25	4.52	11.10
RPCAh + FASST [29]	3.84	6.22	9.19
Supervised			
Model	GNSDR (dB)	GSIR (dB)	GSAR (dB)
MLRR [30]	3.85	5.63	10.70
RNMF [31]	4.97	7.66	10.03
DRNN-2 [9] (L_2)	7.27	11.98	9.99
P-DRNN ($L_2, T = 4$)	7.36	12.31	9.91
P-DRNN ($L_2, T = 10$)	7.74	12.59	10.32

both Ts were stably improved with deeper layers. The P-DRNN approximately reached the highest performance for both GNSDR and GSAR with the deepest layer (12-layer) for both Ts. Since the T-F masking framework for S-RNN and P-DRNN are the same, this observation can be attributed to the Proximal Layer, which leads to an improved P-DRNN over S-RNN for deep layers. The best GNSDR performance of S-RNN was 0.1dB (3-layer) lower than P-DRNN (12-layer) when $T = 4$ and 0.18dB lower (2-layer) than P-DRNN (8-layer) when $T = 10$ and it seems that increasing T also benefits the performance of P-DRNN. For GSIR, the performance of P-DRNN was varying, the reason will be investigated in the future.

Finally, we compared our results with other previous works. Table 1 shows the results with unsupervised and supervised settings. For the loss function L_2, our model of $T = 10$ obtained 0.47 dB GNSDR gain, 0.61 dB GSIR gain, and 0.33 dB GSAR gain, compared to the best results ('DRNN-2') in [9].

4. CONCLUSION

We have introduced a new method to deepen RNNs, i.e., Proximal DRNN, to improve separation performance in MSVS. Our design was derived from the primal-dual method, which offered a proximal interlayer structure that induced more effective information transfer between different layers. In numerical tests, the P-DRNN outperformed many previous approaches on the MSVS problem. Moreover, the proposed method can be potentially extended to the scenario of designing deep models for other MSS problems.

Appendix: a proximal algorithm for MSS

In the following, we omit the index t in all the variables for simplicity. First, we rewrite Eq. (9) as an unconstrained minimization problem. We denote

$$\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_J] \in \mathbb{R}^{N\times J}, \tag{17}$$

$$f(\mathbf{X}) = \sum_{j=1}^{J} \phi_j(\mathbf{x}_j), \tag{18}$$

$$g(\mathbf{X}) = \mathbb{I}_C(\mathbf{X}), \tag{19}$$

where $\mathbb{I}_C$ is the indicator function of set C (see Eq. (1.39) in [16]),

$$C = \left\{ \mathbf{X} \in \mathbb{R}^{M\times J} \middle| \sum_{j=1}^{J} \mathbf{x}_j = \mathbf{m}_t \right\}. \tag{20}$$

Thus Eq. (9) becomes

$$\underset{\mathbf{X}}{\text{minimize}} \quad f(\mathbf{X}) + g(\mathbf{X}). \tag{21}$$

If f and g are closed convex functions with nonempty domains, and the solution of this minimization problem is not empty, the problem in Eq. (21) can be solved by the primal-dual proximal method. Given an auxiliary variable,

$$\mathbf{U} = [U_1, ..., U_J] \in \mathbb{R}^{N\times J}, \tag{22}$$

the primal-dual method gives the following iteration,

$$\mathbf{X}^{k-1/2} \leftarrow \text{Prox}_{\tau f}(\mathbf{X}^{k-1} - \tau\mathbf{U}^{k-1}), \tag{23}$$

$$\mathbf{U}^{k-1/2} \leftarrow \text{Prox}_{\sigma g*}(\mathbf{U}^{k-1} + \sigma(2\mathbf{X}^{k-1/2} - \mathbf{X}^{k-1})), \tag{24}$$

$$\mathbf{X}^{k} \leftarrow \mathbf{X}^{k-1} + \rho_k(\mathbf{X}^{k-1/2} - \mathbf{X}^{k-1}), \tag{25}$$

$$\mathbf{U}^{k} \leftarrow \mathbf{U}^{k-1} + \rho_k(\mathbf{U}^{k-1/2} - \mathbf{U}^{k-1}), \tag{26}$$

where k represents the k-th iteration step, g^* is the conjugate of g, and $\text{Prox}_{\tau f}$ and $\text{Prox}_{\sigma g*}$ are the proximal operators of f and g^* (see Eq. (1.1) in [15]). The parameters ρ_k, τ, and σ are positive [23–25]. Since Eq. (18) suggests that f is separable, according to Proposition 24.11 in [16], $\text{Prox}_{\tau f}$ in Eq. (23) can be broken into N smaller operations that can be carried out independently in parallel,

$$\text{Prox}_{\tau f}(\mathbf{Y}) = \left(\text{Prox}_{\tau\phi_j}(\mathbf{y}_j)\right)_{1\leqslant j\leqslant J}, \forall \mathbf{Y} = [\mathbf{y}_j] \in \mathbb{R}^{N\times J}. \tag{27}$$

The $\text{Prox}_{\sigma g*}$ can be evaluated analytically. In fact, the proximal operator of an indicator function is a projection operator [15, 16],

$$\text{Prox}_{\sigma g}(\mathbf{Y}) = \text{Proj}_C(\mathbf{Y}) \tag{28}$$

$$= \left(\mathbf{y}_j - \bar{\mathbf{Y}} + (1/J)\mathbf{m}_t\right)_{1\leqslant j\leqslant J}, \tag{29}$$

where $\bar{\mathbf{Y}} = 1/J \sum_{j=1}^{J} \mathbf{y}_j$. Suppose $\mathbf{S}$ is a temporary variable,

$$\mathbf{S} = \mathbf{U}^{k-1} + \sigma(2\mathbf{X}^{k-1/2} - \mathbf{X}^{k-1}), \tag{30}$$

according to the following Moreau identity [16]

$$t\text{Prox}_{t^{-1}g*}(\mathbf{Y}/t) = \mathbf{Y} - \text{Prox}_{tg}(\mathbf{Y}), t > 0, \tag{31}$$

Eq. (24) can be simplified as follows,

$$\begin{aligned}
\mathbf{U}^{k-1/2} \leftarrow & \quad \text{Prox}_{\sigma g*}(\mathbf{S}) && \text{(using Eq. (30))}\\
= & \quad \mathbf{S} - \sigma\text{Prox}_{\sigma^{-1}g}(\sigma^{-1}\mathbf{S}) && \text{(using Eq. (31))}\\
= & \quad \mathbf{S} - \sigma\text{Proj}_C(\sigma^{-1}\mathbf{S}) && \text{(using Eq. (28))}\\
= & \quad \left(\bar{\mathbf{S}} - (1/J)\sigma\mathbf{m}_t\right)_{1\leqslant j\leqslant J} && \text{(using Eq. (29))}
\end{aligned}$$

which implies that all elements of $\mathbf{U}^{k-1/2}$ are equal. From the definition of $\mathbf{S}$ in Eq. (30), we have, for every $1 \leqslant j \leqslant J$,

$$U_j^{k-1/2} \leftarrow \bar{\mathbf{U}}^{k-1} + \sigma(2\bar{\mathbf{X}}^{k-1/2} - \bar{\mathbf{X}}^{k-1}) - (1/J)\sigma\mathbf{m}_t. \tag{32}$$

Furthermore, considering both Eqs. (26) and (32), we can conclude that at any iteration step k (or $k-1/2$), all the elements of $\mathbf{U}^k$ (or $\mathbf{U}^{k-1/2}$) are equal,

$$U_j^k = \mathbf{u}^k, \quad U_j^{k-1/2} = \mathbf{u}^{k-1/2}, \qquad (1 \leqslant j \leqslant J). \tag{33}$$

where the elements of $\mathbf{U}^k$ (or $\mathbf{U}^{k-1/2}$) are assumed to be $\mathbf{u}^k$ (or $\mathbf{u}^{k-1/2}$). Based on Eq. (33), Eq. (32) can be simplified as

$$\mathbf{u}^{k-1/2} \leftarrow \mathbf{u}^{k-1} + \sigma(2\bar{\mathbf{X}}^{k-1/2} - \bar{\mathbf{X}}^{k-1}) - (1/J)\sigma\mathbf{m}_t. \tag{34}$$

Based on Eqs. (27) and (34), the iteration of Eqs. (23)-(26) becomes (the index t is omitted for simplicity.)

$$\begin{aligned}
\mathbf{x}_j^{k-1/2} &\leftarrow \text{Prox}_{\tau\phi_j}(\mathbf{x}_j^{k-1} - \tau\mathbf{u}^{k-1}), && (1 \leqslant j \leqslant J)\\
\mathbf{x}_j^{k} &\leftarrow \mathbf{x}_j^{k-1} + \rho_k(\mathbf{x}_j^{k-1/2} - \mathbf{x}_j^{k-1}), && (1 \leqslant j \leqslant J)\\
\mathbf{u}^{k} &\leftarrow \mathbf{u}^{k-1} + \rho_k\left(\sigma(2\bar{\mathbf{X}}^{k-1/2} - \bar{\mathbf{X}}^{k-1}) - (1/N)\sigma\mathbf{m}_t\right).
\end{aligned}$$

5. REFERENCES

[1] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 52, no. 7553, pp. 436–444, 2015.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[3] A. J. R. Simpson, G. Roma, and M. D. Plumbley, "Deep karaoke: extracting vocals from musical mixtures using a convolutional deep neural network," in *Proceedings of the 12th International Conference on Latent Variable Analysis and Signal Separation*, 2015, pp. 429–436.

[4] P. Chandna, M. Miron, J. Janer, and E. Gómez, "Monoaural audio source separation using deep convolutional neural networks," in *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2017, pp. 258–266.

[5] E. M. Grais and M. D. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," In *Proceedings of the 5th IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017.

[6] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," In *Proceedings of the 15th International Society for Music Information Retrieval Conference* , 2014, pp. 477–482.

[7] Y. Bengio, "Learning deep architectures for AI, " *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp.1–127, 2009.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[9] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, 2015.

[10] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 190–198.

[11] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *Proceedings of the International Conference on Learning Representations*, 2014.

[12] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Neural Computation*, vol. 4 , no. 2 , pp. 234–242, 1992.

[13] S. E. Hihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," In *NIPS*, 1996, pp. 493–499.

[14] A. Graves, "Generating sequences with recurrent neural networks," *arXiv:1308.0850 [cs.NE],* 2013.

[15] N. Parikh and S. Boyd,"Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp.127–239, 2014.

[16] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces, 2nd ed.* Springer, 2017.

[17] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[18] S. Wang, S. Fidler and R. Urtasun, "Proximal deep structured models," in *NIPS*, 2016, pp. 865–873.

[19] S. I. Mimilakis, K. Drossos, G. Schuller, and T. Virtanen, "A recurrent encoder-decoder approach with skip-filtering connections for monaural singing voice separation," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017, pp. 1–6.

[20] S. I. Mimilakis, K. Drossos, J. F. Santos, G. Schuller, T. Virtanen, and Y. Bengio, "Monaural singing voice separation with skip-filtering connections and recurrent inference of time-frequency mask," *ArXiv: 1711.01437*, 2017.

[21] A. Liutkus and R. Badeau, "Generalized Wiener filtering with fractional power spectrograms," in *Proceedings of the 40th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 266–270.

[22] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp.2673–2681, 1997.

[23] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, "An algorithm for minimizing the Mumford-Shah functional," In *Proceedings of the IEEE 12th International Conference on Computer Vision*, 2009, pp. 1133–1140.

[24] E. Esser, X. Zhang, and T. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM J. Imaging Sciences*, vol. 3, no. 4, pp. 1015–1046, 2010.

[25] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp.120–145, 2011.

[26] C.-L. Hsu and J.-S. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2010.

[27] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[28] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. H. Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 57–60.

[29] Y.-H. Yang, "On sparse and low-rank matrix decomposition for singing voice separation," in *Proceedings of the ACM International Conference on Multimedia*, 2012, pp. 757–760.

[30] Y.-H. Yang, "Low-rank representation of both singing voice and music accompaniment via learned dictionaries," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013, pp. 427–432.

[31] P. Sprechmann, A. Bronstein, and G. Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling," in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, 2012, pp. 67–72.

[32] F. Mayer, D. Williamson, P. Mowlaee, D. Wang, "Impact of Phase Estimation on Single-Channel Speech Separation Based on Time-Frequency Masking," *Journal of Acoustical Society of America*, vol. 141, no. 6, pp. 4668–4679, 2017.