Evolving Multi-Resolution Pooling CNN for Monaural Singing Voice Separation

Weitao Yuan, Bofei Dong, Shengbei Wang, Masashi Unoki, *Member, IEEE*,
and Wenwu Wang, *Senior Member, IEEE*

***Abstract*—Monaural singing voice separation (MSVS) is a challenging task and has been extensively studied. Deep neural networks (DNNs) are current state-of-the-art methods for MSVS. However, they are often designed manually, which is time-consuming and error-prone. They are also pre-defined, thus cannot adapt their structures to the training data. To address these issues, we first designed a multi-resolution convolutional neural network (CNN) for MSVS called multi-resolution pooling CNN (MRP-CNN), which uses various-sized pooling operators to extract multi-resolution features. We then introduced Neural Architecture Search (NAS) to extend the MRP-CNN to the evolving MRP-CNN (E-MRP-CNN) to automatically search for effective MRP-CNN structures using genetic algorithms optimized in terms of a single objective taking into account only separation performance and multiple objectives taking into account both separation performance and model complexity. The E-MRP-CNN using the multi-objective algorithm gives a set of Pareto-optimal solutions, each providing a trade-off between separation performance and model complexity. Evaluations on the MIR-1 K, DSD100, and MUSDB18 datasets were used to demonstrate the advantages of the E-MRP-CNN over several recent baselines.**

***Index Terms*—Evolving multi-resolution pooling CNN, genetic algorithm, monaural singing voice separation, neural architecture search.**

I. Introduction

POPULAR music, which plays a central role in entertainment industries, usually consists of two components: singing voice and music accompaniment [1]. Humans can easily hear/distinguish the singing voice from music accompaniment when listening to a popular song. This effortless task, however, is very difficult for machines, which presents both challenges and opportunities to advance audio-signal processing techniques [1], [2]. Monaural singing voice separation (MSVS), an important application of music source separation (MSS), aims to separate the singing voice and background music accompaniment from a single-channel mixture signal. Research on MSVS is useful in many areas such as automatic lyric recognition/alignment, singer identification, and music-information retrieval [2].

Many traditional methods are found to be effective for MSVS [1]. Benefiting from these methods, recent data-driven methods, especially deep neural networks (DNNs) [3], strongly boost the performance of MSVS with the help of large-scale data. The basic building blocks of a DNN for MSVS are mainly a feed-forward network (FFN) [4], recurrent neural network (RNN) [5], convolutional neural network (CNN) [6], and attention mechanism [7]. Among these building blocks, a CNN is proven to be very effective in extracting vocal/music features for MSVS, since efficient representations related to discriminative features of vocal/music can be learned using convolutional filters via sharing weights.

Music relies heavily on its multi-scale repetitions (e.g., from very basic elements such as individual notes, timbre, or pitch, to larger structure chords [8]) to build the logical structure [9]. These repetitions appearing at various musical levels also distinguish the music accompaniment from vocals which are less redundant and mostly harmonic [1]. An important CNN for MSVS is a multi-resolution CNN (MR-CNN) [10]–[13], which can capture multi-resolution features by constructing various-sized receptive fields (RFs) and has been found to be effective in extracting multi-scale music features. MR-CNNs have been widely used in many state-of-the-art (SOTA) MSVS methods.

In accordance with the implementation of multi-resolution RFs, there are two types of MR-CNNs for MSVS/MSS. MR-CNNs of the first type, such as stacked hourglass network (SHN) [10] and U-net [11], are constructed in a cascade manner with a fixed-size or single-resolution RF in each layer. The input signal is repeatedly convolved and downsampled to form multiple consecutive layers. In this case, different resolution features can only be found in different layers; thus, the cascade structure of these MR-CNNs should be deep enough to extract effective multi-resolution features. In contrast, MR-CNNs of

Manuscript received February 14, 2020; revised November 14, 2020 and December 29, 2020; accepted January 7, 2021. Date of publication January 15, 2021; date of current version February 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 61902280, 61373104, and 61771340, in part by the Natural Science Foundation of Tianjin under Grants 19JCYBJC15600 and 18JCYBJC15300, in part by Tianjin Science and Technology Project 20YDTPJC00870, in part by a Grant-in-Aid for Scientific Research (B) (17H01761), I-O DATA foundation, and in part by Fund for the Promotion of Joint International Research (Fostering Joint International Research (B)) (20KK0233). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alexey Ozerov. *(Corresponding author: Shengbei Wang.)*

Weitao Yuan, Bofei Dong, and Shengbei Wang are with the Tianjin Key Laboratory of Autonomous Intelligence Technology and Systems, School of Computer Science and Technology, Tiangong University, Tianjin 300387, China (e-mail: weitaoyuan@hotmail.com; tuku@tuxzz.org; wangshengbei@tjpu.edu.cn).

Masashi Unoki is with the School of Information Science, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan (e-mail: unoki@jaist.ac.jp).

Wenwu Wang is with the Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, U.K. (e-mail: w.wang@surrey.ac.uk).

Digital Object Identifier 10.1109/TASLP.2021.3051331

the second type, such as multi-resolution convolutional autoencoder (MRCAE) [12] and multi-resolution fully CNN (MR-FCNN) [13], directly implement multi-resolution RFs in the same layer using multiple sets of various-sized convolutional operators; accordingly, multi-resolution features can be extracted without deepening the cascade structure.

In spite of these achievements, several issues need to be addressed for current MR-CNNs:

1) Structure Limitations: MR-CNNs of the first type depend on their cascade structure to extract multi-scale music features. However, the optimization algorithm will become less effective in capturing the dependencies across multiple layers [14]. MR-CNNs of the second type are not affected by this issue, while to extract global features, large convolutional filters should be used, which results in low computational efficiency [15]. In addition, since a minor linear shift in time-frequency (T-F) representations (e.g., magnitude spectrogram) could cause significant distortions on vocal and music perception [11], many methods use skip connections to transmit low-level information between different layers [10], [11]. However, such mechanisms have not been implemented for the second type of MR-CNNs.

2) Manual Design: Current MR-CNN (or DNN) based MSVS methods are designed manually. This manual design procedure usually has the following shortcomings:

1) Manual design is often achieved empirically via trial and error: An MR-CNN learns hierarchical feature extractors from the data in an "end-to-end" fashion. In this case, slight modifications to the neural structure may significantly affect separation performance. To find suitable structures for MSVS, repetitive modifications and training&testing are required, which is time-consuming, error-prone, and ineffective.
2) Domain knowledge may not be sufficient for neural structure design: For MSVS, domain knowledge may suggest to use vertical and horizontal filters to extract timbral features [16], [17]. However, when dealing with an actual network, how to combine and deploy these filters cannot be fully answered by domain knowledge.
3) Pre-designed structures cannot adapt themselves to the training data: The data-driven optimization process enables MR-CNNs to learn parameters of the convolutional filters. However, the sizes of the pre-defined convolutional operator, hyper-parameters, and structure of MR-CNNs cannot adapt to the dataset during the training process. As a result, the information learned from real data cannot be used to improve the pre-designed structures.

To address these issues, we first designed a flexible and effective MR-CNN for MSVS called the multi-resolution pooling CNN (MRP-CNN). We then introduced neural architecture search (NAS) to extend the MRP-CNN to the evolving framework for MSVS, i.e., evolving MPR-CNN (E-MRP-CNN). The contributions of our study are described below.

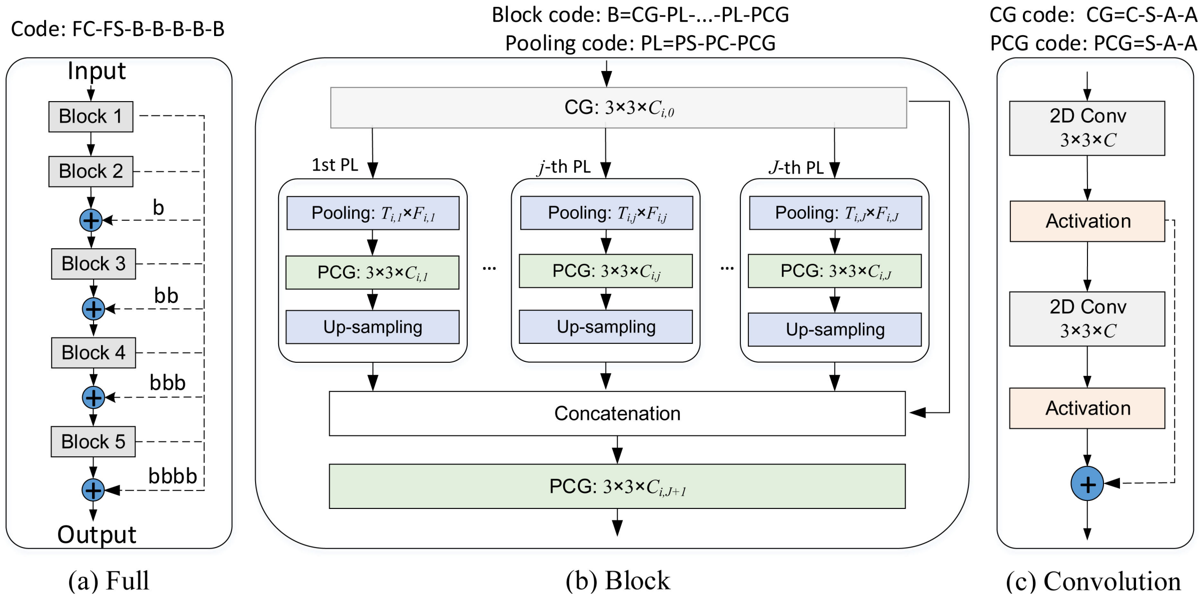
1) MRP-CNN: The MRP-CNN uses sets of average pooling operators of various sizes at the same layer to obtain multi-resolution features. All these pooling operators are embedded in stacked convolutional networks made of small and fixed-sized convolutional kernels. Compared with cascade MR-CNNs (the first type), e.g. U-net and SHN, the MRP-CNN does not need to optimize the deep cascade structure. Compared with the second type of MR-CNNs, large pooling operators instead of large convolutional filters are used to extract global features, which reduces the number of trainable parameters and leads to much better computational efficiency. Moreover, the MRP-CNN has a flexible design and enables skip connections (or similar connections) to be implemented between different layers.

2) Automatic Neural Architecture Search: We introduce NAS to the MRP-CNN to extend it to evolving MRP-CNN, i.e., E-MRP-CNN, for MSVS. The E-MRP-CNN can evolve its structure using two genetic algorithms: single-objective and multi-objective. The E-MRP-CNN using the single-objective algorithm evolves its structure with the only objective of optimizing separation performance (hereafter, "single-objective E-MRP-CNN"). However, it may select a very complex model to optimize separation performance. The E-MRP-CNN using the multi-objective algorithm is used to address the balance between separation performance and model complexity (hereafter, "multi-objective E-MRP-CNN"). It provides a set of Pareto-optimal MRP-CNN structures [18] for MSVS, each providing a good balance between separation performance and model complexity. The E-MRP-CNN can enhance current MR-CNNs, making DNN-based MSVS methods less dependent on domain knowledge.

II. Related Work

Deep networks for MSVS/MSS mainly use RNNs [19], [20] and CNNs [6], [10], [11], [21], [22]. An RNN can effectively model dependencies of temporal patterns and music structure [19], [20]. A CNN, which is effective for feature extraction in the T-F domain, is usually constructed as a convolutional encoder-decoder with skip connections, such as in U-net [11], Wave-U-net [21], Exp-Wave-U-Net [22], and SHN [10]. A CNN can be combined with other structures to improve MSVS/MSS performance. For example, a CNN and RNN were combined to improve MSS performance [23], and skip attention (SA) inspired from Transformer [24] was introduced into the CNN encoder-decoder structure [7]. Generative adversarial networks (GANs) were used for (semi-supervised) MSVS [25], [26], and the Chimera network based on deep clustering was designed for singing voice separation [27]. Mapping functions of neural networks were examined on the basis of the denoising autoencoder (DAE) model [28]. Nevertheless, all these networks are designed manually.

Over the past few years, NAS has achieved impressive progress and begun to outperform human-designed deep models in many research areas [29], [30]. As a classic search strategy of NAS, the NeuroEvolution of Augmenting Topologies (NEAT) [31] adopts a genetic algorithm to evolve neural networks and their weights. Recently, evolved Transformer [29] developed on the basis of NAS has been applied to sequence-to-sequence tasks, and reinforcement learning (RL)-based NAS has been introduced to GANs [32].



(a) Full (b) Block (c) Convolution

Fig. 1. Structure of MRP-CNN.

To the best of our knowledge, NAS has not been explored for MSVS/MSS. We used NAS for automatic architecture design and developed an evolving framework, i.e., E-MRP-CNN, for MSVS. Since the neural structure for MSVS usually has millions of weights, we use genetic algorithms to optimize the neural structure and the gradient-based method to optimize the weights [30], which is different from NEAT [31]. Compared with RL-based NAS (e.g., [32]) which relies on an RNN controller to generate candidate structures [33], the evolution guided NAS only needs to apply genetic operations (mutation and crossover) to create new structures, which is simpler and more efficient for MSVS. Following previous studies [34] [35], we apply a classic multi-objective evolutionary algorithm, i.e., non-dominated sorting genetic algorithm II (NSGA-II) to implement the multi-objective E-MRP-CNN when searching for effective MRP-CNN structures for MSVS.

III. MRP-CNN

A. Proposed Framework

The MRP-CNN is composed of five stacked blocks, as shown in Fig. 1(a). The number of blocks is chosen empirically. Each block (indexed by i, $1 \leq i \leq 5$) works as a basic unit to extract multi-resolution features, and five blocks form a stacked structure. Skip connections (dotted lines in Fig. 1(a)) can be optionally used between different blocks to improve separation performance.

As shown in Fig. 1(b), each block consists of a convolution group (CG) layer, multiple pooling layers (PLs, indexed by j, $1 \leq j \leq J$), concatenation, and a post-convolution group (PCG) layer. The j-th PL in the i-th block is composed of three components: an average pooling operator of size $T_{i,j} \times F_{i,j}$, PCG layer, and upsampling operation. Each PL ($1 \leq j \leq J$) is responsible for extracting one specific resolution feature, and the block, which has multiple PLs, can extract multi-resolution features. The CG and PCG layers in each block have the same structure. As shown in Fig. 1(c), both the CG and PCG layers are made of two consecutive convolution layers with the same size of $3 \times 3 \times C$ and a possible skip connection, where 3×3 represents the kernel size of the 2D convolutional operator and C is the channel number.

Using the hyper-parameters (e.g., $T_{i,j}$, $F_{i,j}$, C, etc.) and flexible components (e.g., skip connection) of the basic MRP-CNN, many different MRP-CNN structures can be induced. For example, in each block, the number of PLs, i.e., J, can be adjusted by the data-driven evolution process of the E-MRP-CNN. In particular, when the size of the average pooling operator of one PL is changed to $T_{i,j} = F_{i,j} = 1$ during the evolution process, this PL will not be used in the current block. In addition, the CG/PCG layer can have different channel numbers (different C) and when $C = 0$, the CG/PCG layer is turned into direct connection, skip connections can be used optionally between different blocks, and nonlinear activation functions can be different (e.g., rectified linear unit (ReLU) or sigmoid). Hence, the MRP-CNN is flexible for MSVS.

B. Encoding Method

To evolve the MRP-CNN structures with genetic algorithms, we first encode the MRP-CNN structure. The encoding process is to assign each specific MRP-CNN structure a unique binary code, i.e., the gene. The binary code makes it convenient for a genetic algorithm to operate since all candidate MRP-CNN structures can be produced by flipping the bit-value of the gene. Genetic algorithms begin with a set of genes (MRP-CNN structures) called a population. In the evolving process, high fitness genes are selected to produce their offspring (new MRP-CNN structures). All possible MRP-CNN structures form a searching space, enabling NAS to find suitable structures for MSVS.

TABLE I
ENCODING METHOD OF MRP-CNN

FC				2bit: 00 (32), 01 (64), 11 (128), 10 (256)
FS				10bit: b-bb-bbb-bbbb (b $\in \{0, 1\}$)
B	CG	C		2bit: 00 (None), 01 (32), 11 (64), 10 (128)
		S		1bit: 0 (No), 1 (Yes)
		A		1bit: 0 (ReLU), 1 (Sigmoid)
		A		1bit: 0 (ReLU), 1 (Sigmoid)
	PL	PS		(2bit)×(2bit): 00 (1), 01 (4), 11 (16), 10 (64)
		PC		2bit: 00 (16), 01 (32), 11 (64), 10 (128)
		PCG	S	1bit: 0 (No), 1 (Yes)
			A	1bit: 0 (ReLU), 1 (Sigmoid)
			A	1bit: 0 (ReLU), 1 (Sigmoid)
	PL	....		
	....	....		
	PCG	S		1bit: 0 (No), 1 (Yes)
		A		1bit: 0 (ReLU), 1 (Sigmoid)
		A		1bit: 0 (ReLU), 1 (Sigmoid)
....	....			
B	....			

TABLE II
CODE (GENE) OF EXAMPLE MRP-CNN STRUCTURE

FC	11 (128)						
FS	0-00-000-0000						
Blocks →			Block 1	Block 2	Block 3	Block 4	Block 5
CG	C		11 (64)	11 (64)	11 (64)	11 (64)	11 (64)
	S		1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)
	A		0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)
	A		0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)
PL	PS		0011 (1×16)	0011 (1×16)	0011 (1×16)	0011 (1×16)	0011 (1×16)
	PC		11 (64)	11 (64)	11 (64)	11 (64)	11 (64)
	PCG	S	1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)
		A	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)
		A	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)
PL	PS		0000 (1×1)	0000 (1×1)	0000 (1×1)	0000 (1×1)	0000 (1×1)
	PC		11 (64)	11 (64)	11 (64)	11 (64)	11 (64)
	PCG	S	1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)
		A	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)
		A	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)
PCG	S		1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)	1 (Yes)
	A		0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)
	A		0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)	0 (ReLU)

To encode the MRP-CNN, we first divide the MRP-CNN structure in Fig. 1 into the following four levels from low to high:

Convolution-level $\subset$ *Pooling-level* $\subset$ *Block-level* $\subset$ *Full-level*,

where *Convolution-level* represents convolutional layers (the CG and PCG layers belong to this level), the *Pooling-level*, *Block-level*, and *Full-level* correspond to a PL, block, and the whole MRP-CNN structure, respectively. The whole MRP-CNN structure is encoded in Table I, where all four levels are included.

1) Full-Level: The *Full-level*, i.e., the whole MRP-CNN structure, is encoded by $\mathrm{FC} - \mathrm{FS} - \mathrm{B} - \mathrm{B} - \mathrm{B} - \mathrm{B} - \mathrm{B}$, where FC encodes the channel number of the last PCG layer in all blocks, i.e., $C_{i,J+1}$ (see Fig. 1(b)), FS encodes possible skip connections between different blocks, B stands for block, and "$-$" represents concatenation of codes.

The FC can be 32/64/128/256, as shown in Table I, where we use 2 bits to represent four options: 00 (32), 01 (64), 11 (128), 10 (256), respectively. The same FC (one of the four options) is used for all blocks in one MRP-CNN structure since the output channels of different blocks should be the same to enable skip connections.

The FS is encoded in form of "b-bb-bbb-bbbb" using 10 bits (see the second row in Table I). The first bit 'b' stands for the skip connection from the first block to the second block, the second 'bb' stands for skip connections from the first and second blocks to the third block, and so on. The value of b determines if skip connection exists (b = 1) or not (b = 0).

2) Block-Level: This level is important to extract multi-resolution features. Each block is encoded as

$$\mathrm{B} = \mathrm{CG} - \underbrace{\mathrm{PL} - \cdots - \mathrm{PL}}_{J} - \mathrm{PCG}.$$

3) Convolution-Level: The CG and PCG layers which have the same structure (see Fig. 1(c)) are encoded differently. The CG is encoded as

$$\mathrm{CG} = \mathrm{C} - \mathrm{S} - \mathrm{A} - \mathrm{A},$$

where C encodes the channel number of convolutional layers in the CG, i.e., $C_{i,0}$ in Fig. 1(b), S stands for skip connection (S $\in \{0, 1\}$), and two consecutive bits $\mathrm{A} - \mathrm{A}$ imply the activation functions for the two-layer convolution operators, where $\mathrm{A} = 0$ represents ReLU and $\mathrm{A} = 1$ represents Sigmoid. The C can be $0/32/64/128$. When $\mathrm{C} = 0$, the CG turns into a direct connection, i.e., there is no convolution, activation, or skip connection. In this case, the $\mathrm{S} - \mathrm{A} - \mathrm{A}$ will be ignored.

The code of the PCG layer is similar to that of the CG but without the channel-number information, i.e.,

$$\mathrm{PCG} = \mathrm{S} - \mathrm{A} - \mathrm{A}.$$

In accordance with Fig. 1(b), the PCG layer is used in both block and PL. Thus the channel number of the PCG layer in block and PL is determined by FC in *Full-level* and PC in *Pooling-level* (see the following), respectively.

4) Pooling-Level: Each PL is encoded using

$$\mathrm{PL} = \mathrm{PS} - \mathrm{PC} - \mathrm{PCG},$$

where PS is the size of the pooling operator in a PL, PC is the channel number of PCG (i.e., $C_{i,j}$ of the j-th PL in the i-th block in Fig. 1(b)). For the j-th PL in the i-th block, PS is defined as $[T_{i,j}, F_{i,j}]$, where $T_{i,j}$ is the downsampling size on the time axis and $F_{i,j}$ on the frequency axis. When $T_{i,j} = F_{i,j} = 1$, the j-th PL will not appear in the i-th block, and the code $\mathrm{PC} - \mathrm{PCG}$ will be ignored. We use 2 bits to encode $T_{i,j}$ and $F_{i,j}$ of PS. As shown in Table I, four possible values are represented by 00 (1), 01 (4), 11 (16), and 10 (64). The PC is also encoded by 2 bits: 00 (16), 01 (32), 11 (64), and 10 (128). The upsampling operator in a PL is not encoded since it has no freedom but to upsample the extracted features back to the same size as the input of the current PL.

A simple example of an MRP-CNN structure is shown in Table II, where all five blocks have two PLs. The PS of the second PL is 0000 ($T_{i,2} = F_{i,2} = 1$), i.e., the PC and PCG layers are ignored (shown in gray). This MRP-CNN structure (or other MRP-CNN structures) can be used as a seed in the E-MRP-CNN.

IV. E-MRP-CNN

The E-MRP-CNN uses single-objective and multi-objective algorithms to search for effective MRP-CNN structures for MSVS. Both single/multi-objective E-MRP-CNNs start with an initial population, which is made of a seed gene (a specific MRP-CNN structure) and other genes (structures) randomly mutated from this gene. After initialization, the single-objective and multi-objective E-MRP-CNNs iteratively generate new offspring genes by applying genetic operations (crossover and mutation) to randomly selected gene(s) from the current population. The new offspring genes are decoded to MRP-CNN structures then trained and tested to compute the fitness. The low-fitness genes will be removed. The computations of fitness in the single-objective and multi-objective E-MRP-CNN are different: the single-objective E-MRP-CNN takes into account only separation performance while the multi-objective E-MRP-CNN takes into account both separation performance and model complexity.

A. Single-Objective E-MRP-CNN

The separation performance of MSVS is often evaluated using three metrics: source-to-distortion ratio (SDR), source-to-interferences ratio (SIR), and sources-to-artifacts ratio (SAR), as used in the Blind Source Separation Evaluation (BSS Eval) toolbox [36]. As a proof of concept, we choose SDR as the fitness function to guide the evolution process of the single-objective E-MRP-CNN because it is a global measure that taking into account three goals: (i) rejection of interferences, (ii) absence of forbidden distortions and "burbling" artifacts, and (iii) rejection of sensor noise, as equally important [36]. In particular, since each gene is only partially trained in the evolution process (to accelerate computation), the global measure SDR would be more suitable than SIR and SAR.

The single-objective E-MRP-CNN is presented in Algorithm 1, where rows 1-4 show the initialization process and rows 5-12 show the evolution process.

Algorithm 1: Single-Objective E-MRP-CNN.

```
1:  Generate the initial population of size $n$ from the seed
    gene
2:  Data preparation: training set $\mathscr{D} \rightarrow \{\mathscr{D}_{tr}, \mathscr{D}_{te}, \mathscr{D}_v\}$
3:  Compute SDR fitness of the initial population by
    partially training each gene on $\mathscr{D}_{tr}$ and testing on $\mathscr{D}_{te}$
4:  Remove low-fitness genes according to the population
    limit $Z$
5:  for $i = 1$ to $N$ (maximum generation) do
6:    Generate $o_c$ new genes by crossover with prob. $p_1$
7:    Generate $o_m$ new genes by mutation with prob. $p_2$
8:    Compute SDR fitness of new offspring using $\mathscr{D}_{tr}$
      and $\mathscr{D}_{te}$
9:    Sort all genes (current+new) by SDR fitness
10:   Remove low-fitness genes by population limit $Z$
11:   break, if stopping criterion is satisfied
12: end for
```

1) Initialization Process:

- In the first step (row 1), we generate the initial population of size n, including one seed gene and the other $n-1$ genes randomly mutated from this seed. To do this, the n_b bits of the seed gene are flipped to generate a new gene, where n_b is a random number and $1 \leq n_b \leq u$ (u is the maximum flipping number). We repeat this process until $n-1$ different genes are obtained.
- In the second step (row 2), we divide the training dataset $\mathscr{D}$ into three subsets $\mathscr{D} \rightarrow \{\mathscr{D}_{tr}, \mathscr{D}_{te}, \mathscr{D}_v\}$, where the training subset $\mathscr{D}_{tr}$ is used for training, the testing subset $\mathscr{D}_{te}$ is used for computing the fitness, and the validation subset $\mathscr{D}_v$ is used to decide when to stop the evolution.
- In the third step (rows 3-4), we compute the fitness of each gene in the initial population. Specifically, the MRP-CNN structure decoded from each gene is trained with $\mathscr{D}_{tr}$ for only a few iterations (i.e., partial training). These partially trained structures are tested on $\mathscr{D}_{te}$, and we compute the average SDR performance over all clips of $\mathscr{D}_{te}$ as the fitness of each gene. The low-fitness genes are removed according to the population limit Z.

2) Evolution Process:

- In each iteration of evolution, we use crossover (row 6) and mutation (row 7) to generate new offspring. The crossover recombines the information of two randomly selected genes, where one gene is used as the baseline and each bit within this gene has a probability (prob.) p_1 to be exchanged with the corresponding bit of the other gene. The mutation produces a new offspring by randomly flipping each bit of one gene with prob. p_2. We repeat the crossover operation to create o_c new offspring then apply mutation to both the current generation (Z genes) and o_c new offspring generated by crossover to create $o_m = Z + o_c$ new offspring.
- The SDR fitnesses of all new offspring ($o_c + o_m$ genes) are computed (row 8). All populations including the new offspring ($o_c + o_m$ genes) and the current populations (Z genes) are sorted by fitness (row 9), and the low-fitness genes are removed in accordance with Z (row 10).
- We check if the stopping criterion is satisfied with the validation subset $\mathscr{D}_v$ (row 11). Specifically, we test the best-fitness gene of the current generation on $\mathscr{D}_v$ to compute its SDR. This SDR is then compared with those of the best-fitness gene of several recent generations (S generations). If there is no improvement in this value for S generations, the evolution iteration will be stopped and the earliest generation with no SDR improvement will be the output.

The single-objective E-MRP-CNN evolves its structure to improve separation performance. However, it may select a very complex model to optimize this performance. In real applications (e.g., the embedded FPGA platform) [37], the computing resources and on-chip memory are usually limited, in this case, both model complexity and separation performance should be considered.

Algorithm 2: Multi-Objective E-MRP-CNN.

```
1:  Generate the initial population of size $n$ from the seed gene
2:  Data preparation: training set $\mathscr{D} \rightarrow \{\mathscr{D}_{tr}, \mathscr{D}_{te}, \mathscr{D}_{v}\}$
3:  Compute SDR fitness and model complexity and then perform fast non-dominated sorting and crowded-comparison
4:  Remove low-fitness genes according to the population limit $Z$
5:  for $i = 1$ to $N$ (maximum generation) do
6:    Generate $o_c$ new genes by crossover with prob. $p_1$
7:    Generate $o_m$ new genes by mutation with prob. $p_2$
8:    Compute SDR and Params for all new offspring
9:    Sort all genes (current+new) using fast non-dominated sorting and crowded-comparison
10:   Remove low-fitness genes by population limit $Z$
11: end for
```

B. Multi-Objective E-MRP-CNN

The multi-objective E-MRP-CNN is designed to balance separation performance and model complexity. It approximates a set of Pareto-optimal solutions [18], i.e., Pareto-optimal MRP-CNN structures, for MSVS. Each solution (structure) is Pareto-optimal, that is, no objective can be improved without degrading the other objective, e.g., the separation performance cannot be improved without increasing the model complexity.

There are generally two properties to design evolutionary multi-objective optimization algorithms: convergence and diversity [38]. Convergence measures the distances of solutions toward the Pareto front (i.e., Pareto-optimal front), which should be as small as possible [38]. Diversity is the spread of solutions along the Pareto front and should be as uniform as possible [38]. For MSVS, convergence encourages each evolved structure to offer the best separation performance as possible under a certain complexity, and diversity encourages the evolved structures to be varied enough to handle different complexity levels.

Following previous studies [34] [35], we approximate the Pareto-optimal solutions on the basis of NSGA-II [18], where the fast non-dominated sorting is used to promote convergence and the crowded-comparison operator is used to address diversity [18]. We use the gray code to encode the MRP-CNN structures (see Table I), as it significantly reduces the searching space and improves the searching efficiency for MSVS.

The multi-objective E-MRP-CNN is presented in Algorithm 2, where rows 1-4 show the initialization process and rows 5-11 show the evolution process. The first two steps in initialization process (rows 1-2) are the same as those in the single-objective E-MRP-CNN (note that the subset $\mathscr{D}_v$ is not used here). In the third step, we compute the fitness of each gene in the initial population. Instead of considering SDR as the only fitness, we compute both SDR and model complexity (measured by the amount of parameters (Params)) for each gene then use fast non-dominated sorting of NSGA-II [18] to calculate the non-dominated levels of all genes. By sorting all these levels with a crowded-comparison operator, low-fitness genes are removed in accordance with Z (row 4).

In each iteration of evolution, we use crossover (row 6) and mutation (row 7) to generate o_c and o_m ($o_m = o_c + Z$) new offspring, respectively. The SDRs and model complexities of all $o_c + o_m$ new offspring are computed. Both the current populations (Z genes) and new offspring ($o_c + o_m$ genes) are sorted by fast non-dominated sorting and the crowded-comparison operator of NSGA-II. We remove low-fitness genes in accordance with Z. The multi-objective E-MRP-CNN stops when the maximum iteration number is reached.

V. Experiment Setting

A. Datasets

The E-MRP-CNN was evaluated on three datasets: MIR-1K [39], DSD100 [40], and MUSDB18 [41]. The original sampling rate for MIR-1 K is 16 kHz and 44.1 kHz for DSD100 and MUSDB18. The MIR-1 K dataset contains a thousand song clips extracted from 110 karaoke songs. For fair comparison, we followed the evaluation conditions in previous studies [7], [10], [42], [43]: 175 clips performed by one male singer 'abjones' and one female singer 'amy' were used for training, the other 825 clips performed by 17 singers were used for testing. On DSD100, songs of the "Dev" subset were used for training, and we followed previous studies [7], [10] to convert all sources to monophonic then added three sources except for vocals together to form the music accompaniment. The MUSDB18 dataset is distributed as stereo mixtures with multiple sound objects sharing a track. In the MSVS task, all musical sources except for vocals were taken as the music accompaniment. To compare the E-MRP-CNN with Open-Unmix [44], we estimated vocal and accompaniment respectively from the left-channel and right-channel of the mixture signal in MUSDB18.

B. T-F Masking Framework

The E-MRP-CNN was evaluated using the T-F masking framework in Fig. 2, where the separation module.[1] can be the structures of the E-MRP-CNN or other compared structures. The output of the separation module is fed to the convolution layer, which is made of two 2D convolution filters (kernel size of 1x1 and no activation function) to estimate the T-F masks for vocal and accompaniment [10], [19], [20], [45]. This framework was used in both the evolution (Evo) process and final evaluation (Eva). For each situation, we have two scenarios: training (Tra) and testing (Tes). For Evo, we trained the evolved structures using $\mathscr{D}_{tr}$ (Evo&Tra) and tested the structures on $\mathscr{D}_{te}$ (Evo&Tes). For Eva, the final evolved structures were trained using the full training set (Eva&Tra) and tested on the full testing set (Eva&Tes).

It should be noted that the T-F masking framework was used differently on the three datasets.

[1] Although it is advantageous to use an independent separation module for each source, i.e., two separation modules for two sources, it is computationally expensive according to a previous study [10] Hence, following that study, we used only one separation module.

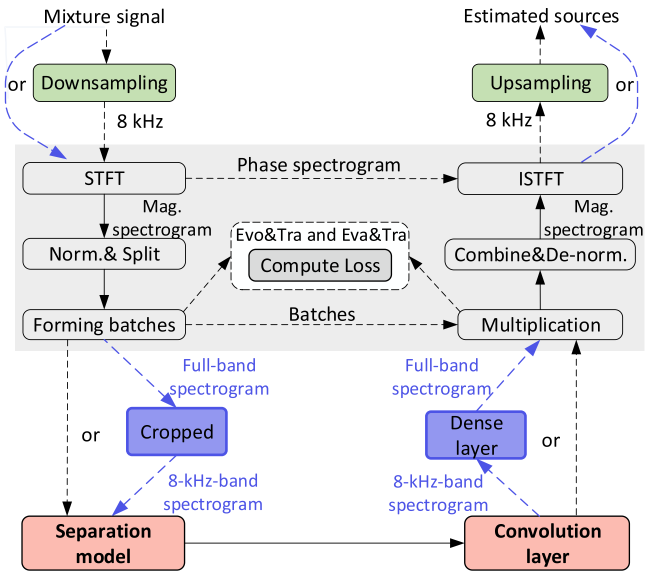


Fig. 2. T-F masking framework, where blue lines signify Eva&Tra and Eva&Tes process for MUSDB18.

- For MIR-1 K and DSD100, the input time-domain mixture signal was downsampled to 8 kHz to speed up computation [10] (see the black dashed line). The spectrogram of the 8-kHz mixture signal was computed via short-time Fourier transform (STFT) using a window size of 1024 (0.128 s) and hop size of 256 (0.032 s). The magnitude spectrogram of the mixture was normalized by dividing by its maximum value then split into blocks of size 512×64 (frequency $\times$ frames) to form batches. The batches of the mixture were fed to the separation module, and its output was fed to the convolution layer to predict the masks (in batches). The predicted masks were used in (*i*) the training process (Evo&Tra and Eva&Tra) to compute the loss function and (*ii*) testing process (Evo&Tes and Eva&Tes) to output the time-domain-estimated sources.
- For MUSDB18, we used the same procedures as above for Evo&Tra and Evo&Tes. However in Eva&Tra and Eva&Tes, to compare with Open-Unmix, we directly calculated the spectrogram with STFT using the original 44.1-kHz sampled mixture signal (see the blue dashed curve). In this case, we set the window size of STFT to 5644 (0.128 s) and hop size to 1411 (0.032 s). The obtained magnitude spectrogram was cropped to the 8-kHz-band spectrogram[2] then split into blocks of size 2822×64 (frequency $\times$ frames) to form batches. The output 8-kHz-band masks were expanded to full-band of 44.1 kHz using a dense layer. This procedure is indicated with blue dashed lines in Fig. 2.

In the training process (Evo&Tra and Eva&Tra), the loss function $L_{1,1}$ norm [10], [46] was adopted for fair comparison. Formally, given the mixture $\mathbf{X}$, the i-th ground truth source $\mathbf{Y}_i$, and predicted mask $\mathbf{M}_i$ for the i-th source ($i = 1\ldots s$, $s = 2$ in MSVS), the loss function is defined as $\mathcal{J} = \sum_{i=1}^{s} \|\mathbf{Y}_i - \mathbf{X} \odot \mathbf{M}_i\|_{1,1}$, where $\odot$ denotes the element-wise multiplication of matrices. Note that when computing the loss function, the magnitude spectrograms of the ground-truth vocal and accompaniment were also normalized by dividing by the maximum value of their mixture's magnitude spectrogram.

TABLE III
HYPER-PARAMETERS OF E-MRP-CNN

Scheme	n	u	N	Z	o_c	o_m	p_1	p_2	$\mathscr{D}_{tr}$	$\mathscr{D}_{te}$	$\mathscr{D}_v$	S
Single	22	20	100	15	10	25	0.5	0.02	100/30/60	55/15/30	20/5/10	8
Multi.	37	20	100	25	10	35	0.5	0.02	100/30/60	55/15/30	–	–

In the testing process (Evo&Tes and Eva&Tes), the predicted masks for vocal and accompaniment were truncated to the range of $[0.0, 1.0]$ and multiplied with the normalized spectrogram of the mixture [10]. After de-normalization and batch combination, the time-domain sources were obtained via inverse STFT (ISTFT) followed by upsampling. For Eva&Tra, two data augmentation operations, gain and sliding, were applied to the original time-domain ground-truth sources to create new mixtures. The gain operation multiplied the original source by a random factor a ($0.5 \leq a \leq 1.5$), and the sliding operation added a random delay d (0 s $\leq d \leq 0.5$ s) to the beginning of the original source. The newly obtained ground-truth sources were mixed to form new mixtures. The ratio of the augmented data to the original data is 1:4.

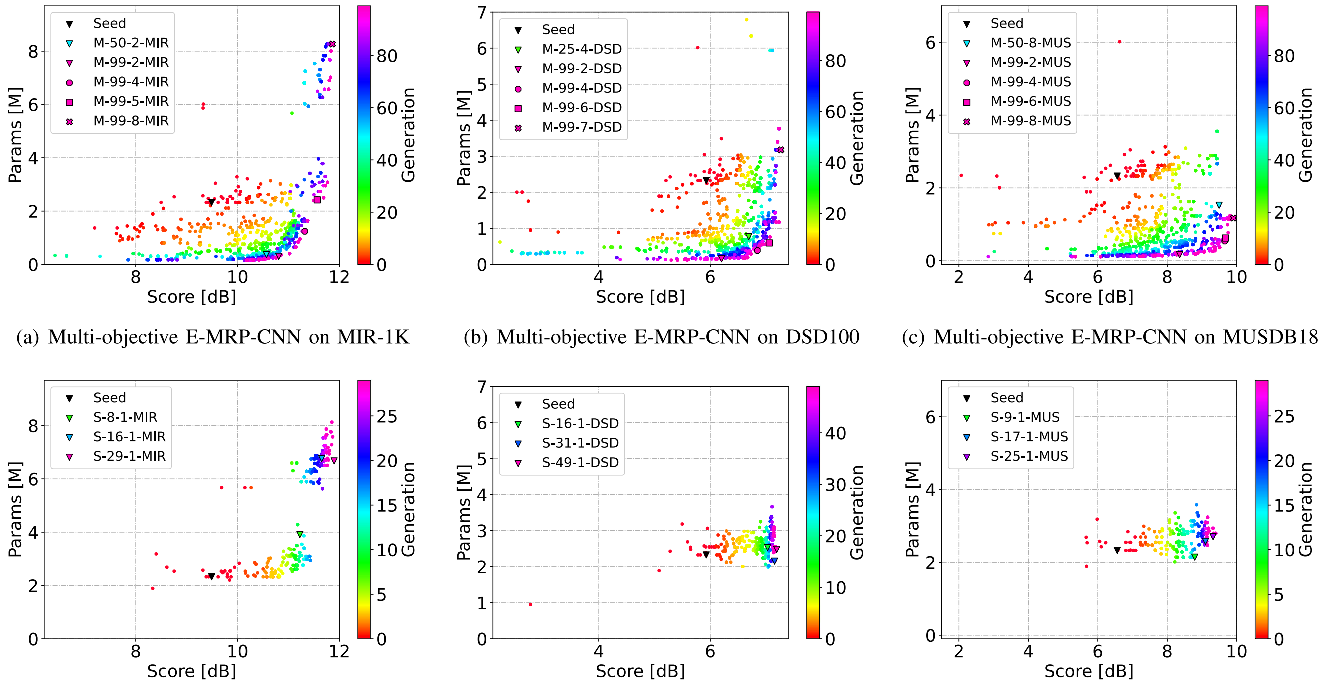
C. Hyper-Parameters of E-MRP-CNN

Table III lists the hyper-parameters of the E-MRP-CNN. Since the multi-objective E-MRP-CNN requires more diversity, its Z and mutation number o_m were higher than those of the single-objective E-MRP-CNN. The parameter n in both single-objective and multi-objective E-MRP-CNNs were higher than Z, which enabled us to remove poor genes and improve the quality of the initial population. The parameter u is the maximum flipping number in mutation, i.e., u controls the flipping percentage. As shown in Table II, each MRP-CNN structure can be encoded with 142 bits. We keep u relatively small to this value as we want to preserve most properties of the parent gene when exploring new genes.

We set $p_1 = 0.5$ in crossover and $p_2 = 0.02$ in mutation. Intuitively, crossover can be considered as combining genes of parents, and mutation can be considered as exploring new genes. The parameter p_1 was set much larger than p_2, because the parents in crossover are selected from the previous generation, so it is relatively safe to combine their genes at a high probability. In contrast, p_2 was much smaller, that is, we only explore new genes around the parent. Such a setting ensures that the newly explored genes inherit most good properties from the parent to maintain its performance.

The parameters $\mathscr{D}_{tr}$, $\mathscr{D}_{te}$, and $\mathscr{D}_v$ were set to $100/55/20$, $30/15/5$, $60/30/10$ for MIR-1 K, DSD100, and MUSDB18, respectively. In the single-objective E-MRP-CNN, we use a criterion, i.e., no improvement on SDR for S generations, to stop the evolution. Since small S may stop the evolution too early, we set S relatively high, i.e., $S = 8$, to ensure the completeness of

[2] The standard Open-Unmix crops the input spectrogram to the 16-kHz-band spectrogram, while to speed up the computation, the E-MRP-CNN crops the input spectrogram to the 8-kHz-band spectrogram.



(a) Multi-objective E-MRP-CNN on MIR-1K (b) Multi-objective E-MRP-CNN on DSD100 (c) Multi-objective E-MRP-CNN on MUSDB18

(d) Single-objective E-MRP-CNN on MIR-1K (e) Single-objective E-MRP-CNN on DSD100 (f) Single-objective E-MRP-CNN on MUSDB18

Fig. 3. Evolution process of single-objective and multi-objective E-MRP-CNNs on MIR-1 K, DSD100, and MUSDB18.

single-objective evolution. In the multi-objective E-MRP-CNN, $\mathscr{D}_v$ and S were not used.

D. Training and Testing Parameters

The Adam optimizer [47] was used to train the T-F masking framework. In Evo&Tra, we aimed to compute the fitness of each gene, so the T-F masking framework was only partially trained with 1500 iterations for MIR-1 K and 3100 iterations for DSD100 and MUSDB18 using batch size of 2. In Eva&Tra, the framework was fully trained with 63 000 iterations for MIR-1 K and 630 000 iterations for DSD100 and MUSDB18 using batch size of 3. In Evo&Tes and Eva&Tes, we used batch size of 1.

In both Evo&Tra and Eva&Tra, two tricks were used: (i) cosine decay learning rate and warm restart [48] and (ii) learning rate warmup [49]. For (i), we set T_0=100 and T_m=2 in Evo&Tra, and T_0=1000 (10 000) for MIR-1 K (DSD100/MUSDB18) and T_m=2 in Eva&Tra, where T_0 is the length of the first decay period [48] and T_m is the multiplication factor for decay-period length at every new warm restart [48]. The maximum learning rate for Evo&Tra and Eva&Tra was 3×10^{-4}, and the minimum learning rates for Evo&Tra and Eva&Tra were 1×10^{-4} and 1×10^{-5}, respectively (more details can be found in [48]). For (ii), we scaled the learning rate in the first 100 (1000) iterations for Evo&Tra (Eva&Tra) with a factor 0.3 to avoid the maximum learning rate being too large for some genes.

VI. Evolution Analysis of E-MPR-CNN

We used the MRP-CNN structure in Table II as the seed gene of the initial population for both single-objective and multi-objective E-MRP-CNNs. The evolved genes (structures) of the E-MRP-CNN are represented in the form of “S/M-G-Index-Dataset,” where S and M denote the single-objective and multi-objective E-MRP-CNNs, respectively, G represents the generation number, Index is the gene index in the G-th generation, and Dataset can be MIR (MIR-1 K), DSD (DSD100), or MUS (MUSDB18). For the single-objective E-MRP-CNN, the Index is the SDR ranking of a gene in the current generation. For the multi-objective E-MRP-CNN, the Index is the gene index in the current generation. For example, “S-25-2-MIR” represents the structure with the second highest SDR performance in the 25th generation of the single-objective E-MRP-CNN on MIR-1 K, and “M-99-2-DSD” represents the No. 2 evolved structure in the 99th generation of the multi-objective E-MRP-CNN on DSD100.

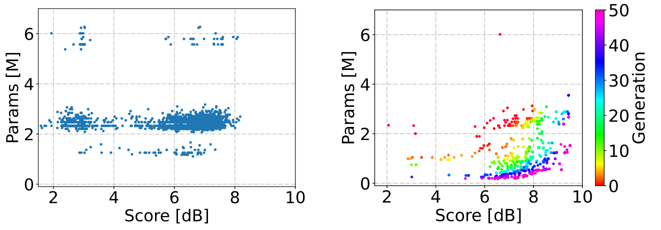
A. Trends and Results of Evolution

We recorded the dynamic evolution process of the E-MRP-CNN on MIR-1 K, DSD100, and MUSDB18, as shown in Fig. 3. The vertical axis in each figure represents the model complexity measured by the number of parameters (Params), and the horizontal axis represents the fitness score measured by SDR (SDR score). Each colored data point stands for a gene, i.e., an MRP-CNN structure. The genes of different generations are distinguished by colors changing from red (initial generation) to pink (highest generation). We set the highest evolution number to 99. The single-objective E-MRP-CNN stopped evolving at the 16th generation on MIR-1 K, 31st generation on DSD100, and 17th generation on MUSDB18 when the SDR of the best gene had no improvement for $S = 8$ consecutive generations. For the multi-objective E-MRP-CNN, we observed the evolution process of all 100 generations (i.e., $0 \leq \mathrm{G} \leq 99$) on the three datasets. By comparing the top and bottom panels in Fig. 3, we can find that the single-objective and multi-objective E-MRP-CNNs had different evolution trends.

As shown in Figs. 3(a)–3(c), the genes in the multi-objective E-MRP-CNN moved toward the Pareto front generation by generation during the evolution process. More specifically, we can see that the seed gene (represented with the black inverted triangle) had a relatively high model complexity and low SDR score. As the evolution proceeded, the new generations gradually moved to the Pareto-optimal front. For example, the first 10 generations in Figs. 3(a)–3(c) (red and yellow points) spread widely, the 10 to 40 generations (yellow and green points) started to move to the lower-right boundary, and the higher generations, e.g., 70 to 99 generations (blue and pink points), approximately converged to the Pareto-optimal front. These results indicate that better genes (in model complexity, in SDR, or in both) were obtained during the evolution process. Finally, a set of structures with better overall performance in model complexity and/or SDR were obtained, which can deal with different complexity requirements.

Compared with the multi-objective E-MRP-CNN, the model complexity of genes in the single-objective E-MRP-CNN (see Figs. 3(d)–3(f)) did not decrease during the evolution process since model complexity was not considered in it. In particular, we can see that the single-objective E-MRP-CNN, without the constraint of model complexity, could steadily improve SDR generation by generation. By comparing the single-objective and multi-objective E-MRP-CNNs on each dataset, we found that the single-objective E-MRP-CNN could achieve a similar SDR as the multi-objective E-MRP-CNN with much fewer generations. For example, on MIR-1 K, the single-objective E-MRP-CNN reached $\mathrm{SDR} = 11$ dB using only $10 \leq \mathrm{G} \leq 15$ generations, while this required at least 20 generations in the multi-objective E-MRP-CNN. Nevertheless, the multi-objective E-MRP-CNN had a lower model complexity than the single-objective E-MRP-CNN at $\mathrm{SDR} = 11$ dB. We also found that the single-objective E-MRP-CNN behaved differently on the three datasets. On MIR-1 K (Fig. 3(d)), model complexity significantly increased at a high SDR score while this phenomenon was not observed on DSD100 and MUSDB18 (see Figs. 3(e)–3(f)).

We labelled some representative genes of the multi-objective E-MRP-CNN in Figs. 3(a)–3(c), including the seed gene, gene of early generation, and genes of the final generation ($\mathrm{G} = 99$). It is clear that better genes (in model complexity, SDR, or both) were obtained during the evolution process. For the single-objective E-MRP-CNN, we intentionally continued the evolution process for a few more generations. Typical genes including the seed gene, gene of early generation, genes of the final generation, and genes after the final generation ($\mathrm{G} = 29$ for MIR-1 K, $\mathrm{G} = 49$ for DSD100, and $\mathrm{G} = 25$ for MUSDB18) are plotted. We found from Fig. 3(d) that the gene after final generation, i.e., S-29-1-MIR, provided higher SDR performance than the best gene of the final generation, i.e., S-16-1-MIR, on the testing subset $\mathscr{D}_{te}$. This phenomenon is also found in Fig. 3(f). The performance of these structures are evaluated in the next section.



(a) Random search

(b) Multi-objective E-MRP-CNN

Fig. 4. Comparison between random search and multi-objective E-MRP-CNN on MUSDB18.

The evolution process of the E-MRP-CNN is relatively time-consuming. In our experiments, we use six GPUs in parallel for evolution.[3] In the single-objective E-MRP-CNN, it took about 1.5 hours to perform one iteration on MIR-1 K and 2 hours on DSD100 and MUSDB18. In the multi-objective E-MRP-CNN, it took about 1 h to perform one iteration on MIR-1 K and 2 hours on DSD100 and MUSDB18. Nevertheless, it should be noted that some evolved structures in the multi-objective E-MRP-CNN can be trained much faster than SOTA methods on the full dataset because they have lower model complexity. More details are given in the following sections.

B. Comparison Between Random Search and Evolution

To verify the effectiveness of the E-MRP-CNN, we compared the multiple-objective E-MRP-CNN with random search on MUSDB18 using the same seed structure. As shown in Table III, we had 37 genes in the initial population and 45 (o_c+o_m) genes in each iteration. We set the maximum generation to 50 and thus 2287 structures were found. For fair comparison, we also searched 2287 structures with random search using the same seed structure. The performances of random search and the multiple-objective E-MRP-CNN under the same partial training and testing conditions are compared in Fig. 4.

We can see that most structures found by random search had a model complexity of $2{\sim}3$ M. Their SDR scores were around $2{\sim}4$ dB and $5{\sim}8$ dB. Compared with random search, the performance of evolved structures in the multi-objective E-MRP-CNN improved generation by generation in the evolution process. Some structures in high generations had a much higher SDR score (>8 dB) and lower complexity (<2 M) than those in random search. These results indicate that random search cannot obtain the Pareto-optimal set as with the E-MRP-CNN, which verifies the advantage of the E-MRP-CNN over random search.

VII. Evaluation of Evolved Architectures

We evaluated typical evolved structures on the full MIR-1 K, DSD100, and MUSDB18 datasets. Their performances were analyzed with respect to separation performance, computational efficiency, and separation performance vs. computational efficiency.

[3] The six GPUs were two 1080Ti, one 2080Ti, one Titan RTX, one Titan V, and one Titan XP.

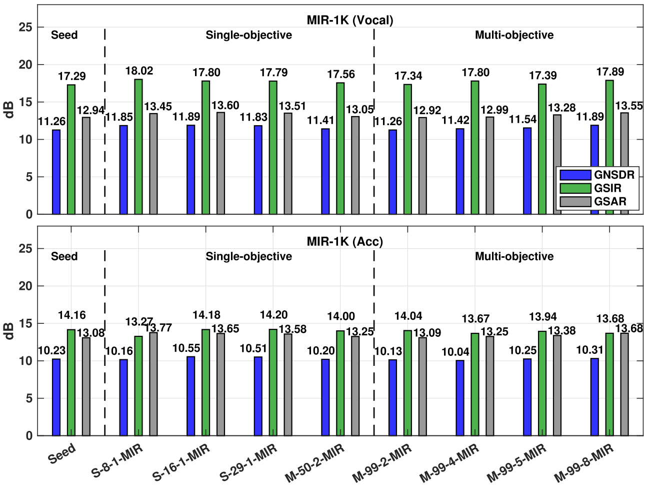


Fig. 5. Separation performance of E-MRP-CNN on MIR-1 K.

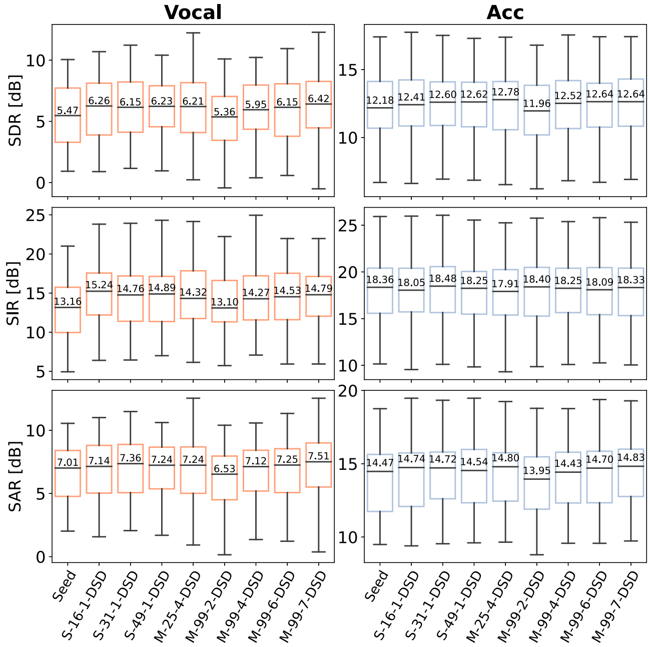


Fig. 6. Separation performance of E-MRP-CNN on DSD100.

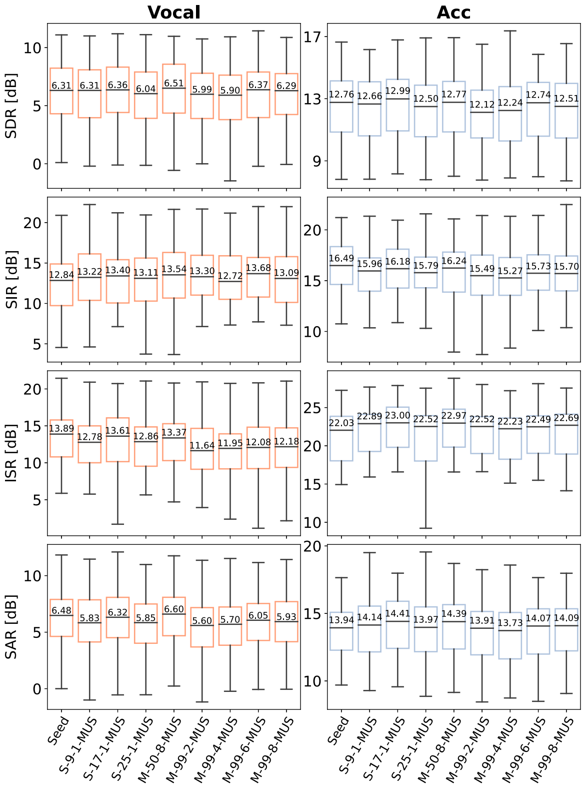


Fig. 7. Separation performance of E-MRP-CNN on MUSDB18.

A. Separation Performance

Separation performance was measured using the BSS-EVAL toolkit [36] with respect to three criteria: SDR, SIR, and SAR. The normalized SDR (NSDR) [50] indicates the improvement of SDR compared to the original mixture. To compare the E-MRP-CNN with current methods, we computed Global NSDR (GNSDR), Global SIR (GSIR), and Global SAR (GSAR) [10], [42] on MIR-1 K; for DSD100, we plotted the SDR, SIR, and SAR results with boxplots; for MUSDB18, we plotted the SDR, SIR, and SAR results as well as the source image-to-spatial distortion ratio (ISR) result with boxplots.

The separation performances of the evolved structures on MIR-1 K are shown in Fig. 5. We can see that the evolved structures in both single-objective and multi-objective E-MRP-CNNs achieved higher GNSDR, GSIR, and GSAR on the vocal (Vocal) source and higher GSAR on the accompaniment (Acc) source than the seed. The separation results for DSD100 are shown in Fig. 6. For the Vocal source, most evolved structures in the single-objective and multi-objective E-MRP-CNNs outperformed the seed in three evaluation metrics. For the Acc source, most evolved structures achieved higher SDR and SAR than the seed. The separation results for MUSDB18 are shown in Fig. 7. The evolved structures in the single-objective E-MRP-CNN performed better in SDR and SIR for Vocal and ISR and SAR for Acc. In the multi-objective E-MRP-CNN, the evolved structures did not show clear separation improvements. The main reason is that the multi-objective E-MRP-CNN compromises its separation performance to lower model complexity.

As mentioned above, S-29-1-MIR (a structure of a later generation after the stopping criterion was satisfied) provided a higher SDR than the final evolved S-16-1-MIR on the testing subset $\mathscr{D}_{te}$ (see Fig. 3(d)). This gene did not outperform S-16-1-MIR on GNSDR in either Acc or Vocal on the full MIR-1 K dataset, as shown in Fig. 5. Similarly, as shown in Fig. 7, S-25-1-MUS did not outperform S-17-1-MUS on SDR in either Acc or Vocal on the full MUSDB18 dataset. These results verified the effectiveness of our stopping criterion of the single-objective E-MRP-CNN.

To determine whether the separation improvements of single-objective and multi-objective structures are statistically significant to the seed, we conducted a one-way analysis of variance (ANOVA)-based F-test [51] on MIR-1 K and a Kruskal-Wallis [52] based H-test on DSD100 and MUSDB18. The one-way ANOVA (for MIR-1 K) test can be used to compare means

TABLE IV
STATISTICAL SIGNIFICANCE EVALUATION OF NSDR (FOR MIR-1 K) AND SDR (FOR DSD 100 AND MUSDB18) OF SINGLE-OBJECTIVE AND MULTI-OBJECTIVE E-MRP-CNN STRUCTURES AND SEED

MIR-1K (NSDR, mean)					**DSD100 (SDR, median)**					**MUSDB18 (SDR, median)**				
Structure	Acc		Vocal		Structure	Acc		Vocal		Structure	Acc		Vocal	
	F-value	p-value	F-value	p-value		H-value	p-value	H-value	p-value		H-value	p-value	H-value	p-value
S-8-1-MIR	0.19	0.67	18.38	1.92e-05	S-16-1-DSD	3.32	0.07	39.15	3.93e-10	S-9-1-MUS	1.61	0.20	0.81	0.37
S-16-1-MIR	5.18	0.02	19.42	1.11e-05	S-31-1-DSD	10.04	1.53e-03	18.11	2.08e-05	S-17-1-MUS	12.51	4.05e-04	6.85	8.87e-03
S-29-1-MIR	4.13	0.04	16.81	4.33e-05	S-49-1-DSD	2.88	0.09	16.49	4.89e-05	S-25-1-MUS	8.23	4.12e-03	11.58	6.67e-04
M-50-2-MIR	0.04	0.85	1.04	0.31	M-25-4-DSD	1.09	0.30	17.27	3.24e-05	M-50-8-MUS	2.68	0.10	4.10	0.04
M-99-2-MIR	0.48	0.49	1.21e-03	0.97	M-99-2-DSD	25.51	4.41e-07	24.23	8.56e-07	M-99-2-MUS	37.59	8.73e-10	26.59	2.52e-07
M-99-4-MIR	1.53	0.22	1.37	0.24	M-99-4-DSD	0.17	0.68	0.14	0.71	M-99-4-MUS	45.97	1.20e-11	37.88	7.51e-10
M-99-5-MIR	0.03	0.86	3.80	0.05	M-99-6-DSD	2.30	0.13	8.90	2.86e-03	M-99-6-MUS	5.16	0.02	2.65	0.10
M-99-8-MIR	0.38	0.54	20.04	8.10e-06	M-99-7-DSD	4.81	0.03	19.95	7.95e-06	M-99-8-MUS	13.66	2.19e-04	6.04	0.01

of two or more samples (using the F distribution), and it tests the null hypothesis that two groups have the same population mean. The Kruskal-Wallis test (for DSD100 and MUSDB18) can be seen technically as a comparison of the ranks of the data points, rather than the data points themselves, and it tests the null hypothesis that the population median of two groups are equal. If the p-value is lower than 0.05 (5% significance level), the null hypothesis is rejected, which means the given results are statistically significant.

The ANOVA test was conducted using all 825 testing clips in MIR-1 K. For DSD100 and MUSDB18, the original audio tracks were cut to a duration of 1.0 s, then all 8696 (DSD100) and 9633 (MUSDB18) testing clips[4] were used for the Kruskal-Wallis test. The evaluation results for NSDR on MIR-1 K and SDR on DSD100 and MUSDB18 are listed in Table IV. For both Acc and Vocal, the p-values of all final evolved structures in the single-objective E-MRP-CNN (i.e., S-16-1-MIR, S-31-1-DSD, S-17-1-MUS) were smaller than 0.05, indicating that the separation improvements were statistically significant. Compared with these structures, the structures in the early generation, e.g., S-8-1-MIR and S-16-1-DSD, only achieved significant improvement on the Vocal source. On MUSDB18, the SDR of S-9-1-MUS was lower than that of the seed on Acc (see Fig. 7), while this discrepancy was not statistically significant, as shown in Table IV. The structures in the multi-objective E-MRP-CNN took into account both separation performance and model complexity; thus, only a few structures had significant separation improvements, e.g., M-99-7-DSD. It should be noted that the statistical analysis results are affected by the effect size, and the current results should be used for reference only.

In addition to these results, some audio samples are also available at *https://tuxzz.org/emrpcnn-ckpt/*, so that the reader can listen to them for a qualitative comparison on separation performance.

B. Computational Efficiency

Computational efficiency was calculated in theory and measured in a real hardware/software environment. The theoretical efficiency was given by Params and FLOPs, where Params denotes the number of trainable parameters of each structure and FLOPs represents the floating-point operations for training (inferring) using batch size of 1. In practice, two structures with similar Params and FLOPs may have different computation speeds; thus, computational efficiency was also measured in a real hardware/software environment.[5] The real computational efficiency in training and inferring was given in bat./s., that is, number of batches per second.

The computational efficiency on the three datasets are plotted in Fig. 8. We used the same seed gene for the three datasets; however, the model complexity of the seed gene for MIR-1 K and DSD100 was lower than that of MUSDB18. This is because the block size (512 × 64) for MIR-1 K and DSD100 was smaller than that of MUSDB18 (2822 × 64). In particular, since a dense layer was used for MUSDB18, the Params of MUSDB18 was higher than those of MIR-1 K and DSD100.

On MIR-1K, the model complexity of the evolved structures in the single-objective E-MRP-CNN increased generation by generation and most structures had higher model complexities than the seed. In the multi-objective E-MRP-CNN, multiple structures were provided in one generation with varying model complexities, e.g., M-99-2/4/5/8-MIR. Most evolved structures had similar or even lower model complexities compared with the seed and those in the single-objective E-MRP-CNN. On DSD100 and MUSDB18, the model complexities of evolved structures in the single-objective E-MRP-CNN increased slightly. In the multi-objective E-MRP-CNN, most evolved structures on DSD100 (M-99-2/4/6-DSD) and MUSDB18 (M-99-2/4/6/8-MUS) had lower model complexities compared with the seed and those in the single-objective E-MRP-CNN.

C. Separation Performance vs. Computational Efficiency

When comparing the single-objective E-MRP-CNN with the seed, we found that the single-objective E-MRP-CNN could achieve much better separation with a slightly higher model complexity. For example, on MIR-1 K (see Figs. 5 and 8), S-16-1-MIR, which had 0.63 dB GNSDR improvement on Vocal with respect to the seed, only needed an additional cost of 4.43 M

[4] Since the standard BSS-EVAL toolbox uses 1.0-s segments to compute SDR (https://sigsep.github.io/datasets/musdb.html), we followed this setting to run the statistical tests.

[5] The GPU we used was 2080Ti, CPU was Intel Core i9 9900 K, and memory was 4×16 G DDR4 (3200 MHz). In the Linux operating system, we used TensorFlow 2.0 with CUDA 10.1 and cuDNN 7.6.

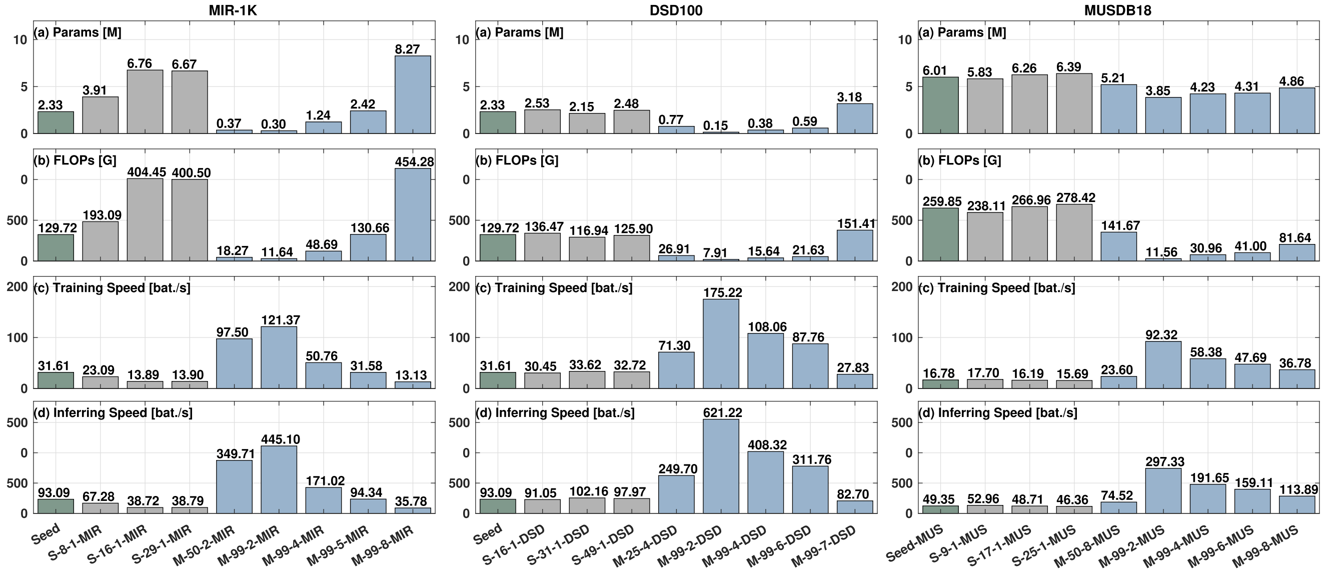


Fig. 8. Computational efficiency of E-MRP-CNN on the three datasets.

Params and 274.73 G FLOPs. On DSD100 (see Figs. 6 and 8), the single-objective E-MRP-CNN achieved much better separation with similar or even lower model complexity to the seed, e.g., S-31-1-DSD. On MUSDB18 (see Figs. 7 and 8), the final evolved structure, i.e., S-17-1-MUS, achieved better separation than the seed in most evaluation metrics while maintaining a similar model complexity.

When comparing the multi-objective E-MRP-CNN with the seed, we found that it could achieve better separation than the seed with a lower model complexity. For example on DSD100, M-99-6-DSD achieved 0.68 dB improvement in SDR on Vocal with respect to the seed using only 25.3% Params and 16.7% FLOPs of the seed. In the real environment, this structure was also 2.77 times (Training) and 3.35 times (Inferring) faster than the seed. Similar results were obtained on MIR-1 K, e.g., M-99-5-MIR and M-50-2-MIR outperformed the seed in most evaluation metrics with lower and similar model complexities. Different from these results, the final evolved structures on MUSDB18, which significantly reduced model complexity, did not achieve clear separation improvement compared with the seed.

By comparing Figs. 5-7 and Fig. 8, we can also see that within the same generation of the multi-objective E-MRP-CNN, the structures with higher model complexities usually provided better separation, e.g. from M-99-2-MIR to M-99-8-MIR and from M-99-2-DSD to M-99-7-DSD.

When comparing the single-objective E-MRP-CNN with the multi-objective E-MRP-CNN, we found that the multi-objective E-MRP-CNN could sometimes find more effective and efficient structures (similar or lower model complexity but better separation performance) than the single-objective E-MRP-CNN. For example, M-99-6-DSD had a higher SDR than S-31-1-DSD on Acc but with only 27.4% Params and 18.5% FLOPs of S-31-1-DSD. In the real environment, M-99-6-DSD was 2.61 times (Training) and 3.05 times (Inferring) faster than S-31-1-DSD. These observations suggest that the multi-objective E-MRP-CNN can greatly reduce model complexity while maintaining acceptable separation. Such a phenomenon was also observed on MIR-1 K, e.g., M-99-4-MIR, which had much higher computational efficiency than S-16-1-MIR and had the same SDR on Vocal.

VIII. Comparative Evaluations

A. On MIR-1 K and DSD100 Datasets

We first compared the best structures in the single-objective and multi-objective E-MRP-CNNs with several typical MR-CNN-based MSVS methods: MR-FCNN [13], SHN [10], and SA-SHN [7], on MIR-1 K and DSD100. The computational efficiencies and separation performances are listed in Table V, where we use SHN-n and SA-SHN-n to represent the n-layer SHN and n-layer SA-SHN, respectively.

We can see that the MR-FCNN has an acceptable model complexity, however, its separation performance was much lower than those of the other methods on both datasets. On MIR-1 K, the single-objective structure, i.e., S-16-1-MIR, achieved the best separation in most metrics while maintaining a similar model complexity as SHN and SA-SHN. The multi-objective structure, i.e., M-99-5-MIR, which had much lower model complexity than SHN and SA-SHN, achieved comparable separation performance to SHN and SA-SHN. On DSD100, both the single-objective structure (S-31-1-DSD) and multi-objective structure (M-99-7-DSD) achieved lower model complexities than SHN and SA-SHN, and their separation performances were better than most of the SHN and SA-SHN models. The statistical results of the above methods are listed in Table VI. Our structures achieved significant improvement on MIR-1 K and DSD100 compared with most of the SHN and SA-SHN models. These results verified the effectiveness of the E-MRP-CNN.

We also compared the E-MRP-CNN with other MSVS methods on MIR-1 K and DSD100. Since the model complexity

TABLE V
COMPARISON OF COMPUTATIONAL EFFICIENCY AND SEPARATION PERFORMANCE BETWEEN E-MRP-CNN AND CURRENT MR-CNN-BASED MSVS METHODS (MR-FCNN, SHN-∗, AND SA-SHN-∗)

Method	Computational efficiency				MIR-1K						DSD100					
	Params [M]	FLOPs [G]	Training [bat./s]	Inferring [bat./s]	Acc			Vocal			Acc (Median)			Vocal (Median)		
					GNSDR	GSIR	GSAR	GNSDR	GSIR	GSAR	SDR	SIR	SAR	SDR	SIR	SAR
MR-FCNN	0.56	36.56	9.03	18.59	8.65	11.65	12.35	9.66	15.72	11.40	11.28	16.48	13.59	4.76	12.43	5.83
SHN-1	9.06	168.29	29.94	87.70	9.85	13.66	12.85	10.88	16.63	12.71	12.11	17.78	14.20	5.42	13.46	6.66
SHN-2	17.46	292.87	16.70	49.19	9.94	13.67	12.96	11.10	17.13	12.82	12.01	17.95	14.43	5.67	13.80	6.76
SHN-4	34.18	537.66	8.84	26.09	9.97	13.65	13.08	11.13	17.09	12.89	12.17	17.63	14.61	5.85	14.29	7.07
SA-SHN-1	9.85	197.29	14.41	40.08	10.12	13.78	13.25	11.32	17.15	13.10	12.17	17.71	14.73	5.91	14.76	7.17
SA-SHN-2	19.03	350.87	7.56	20.95	10.34	13.99	13.46	11.71	17.58	13.44	12.33	18.06	14.73	6.11	14.79	7.27
SA-SHN-4	37.33	653.67	3.87	10.70	10.53	**14.54**	13.38	11.75	**17.87**	13.40	12.63	18.04	**14.90**	6.24	**15.14**	7.31
S-16-1-MIR	6.76	404.45	13.89	38.72	**10.55**	14.18	**13.65**	**11.89**	17.80	**13.60**	–	–	–	–	–	–
M-99-5-MIR	2.42	130.66	31.58	94.34	10.25	13.94	13.38	11.54	17.39	13.28	–	–	–	–	–	–
S-31-1-DSD	2.15	116.94	33.62	102.16	–	–	–	–	–	–	12.60	**18.48**	14.72	6.15	14.76	7.36
M-99-7-DSD	3.18	151.41	27.83	82.70	–	–	–	–	–	–	**12.64**	18.33	14.83	**6.42**	14.79	**7.51**

TABLE VI
STATISTICAL SIGNIFICANCE EVALUATION OF NSDR AND SDR BETWEEN E-MRP-CNN AND MR-FCNN, SHN-∗, AND SA-SHN-∗ ON MIR-1 K AND DSD100

Method	MIR-1K (NSDR)								DSD100 (SDR)							
	S-16-1-MIR				M-99-5-MIR				S-31-1-DSD				M-99-7-DSD			
	Acc		Vocal		Acc		Vocal		Acc		Vocal		Acc		Vocal	
	F-value	p-value	F-value	p-value	F-value	p-value	F-value	p-value	H-value	p-value	H-value	p-value	H-value	p-value	H-value	p-value
MR-FCNN	190.33	4.69e-41	244.01	2.11e-51	132.09	1.83e-29	175.33	3.79e-38	398.47	1.19e-88	147.08	7.54e-34	361.78	1.15e-80	150.12	1.63e-34
SHN-1	23.60	1.30e-06	49.57	2.80e-12	7.52	6.18e-03	21.40	4.02e-06	54.69	1.41e-13	39.70	2.96e-10	41.52	1.17e-10	42.07	8.82e-11
SHN-2	18.26	2.03e-05	29.80	5.52e-08	4.62	0.03	9.24	2.41e-03	34.23	4.89e-09	11.81	5.90e-04	23.98	9.75e-07	13.37	2.56e-04
SHN-4	16.29	5.69e-05	27.50	1.77e-07	3.74	0.05	8.00	4.72e-03	22.00	2.73e-06	2.00	0.16	13.97	1.86e-04	2.75	0.10
SA-SHN-1	8.84	2.98e-03	15.44	8.86e-05	0.78	0.38	2.19	0.14	18.26	1.93e-05	0.01	0.92	11.01	9.06e-04	0.14	0.71
SA-SHN-2	2.19	0.14	1.58	0.21	0.36	0.55	1.51	0.22	14.00	1.83e-04	0.15	0.70	7.74	5.40e-03	0.01	0.91
SA-SHN-4	0.01	0.91	0.90	0.34	3.76	0.05	2.35	0.13	0.05	0.82	17.04	3.67e-05	1.40	0.24	14.40	1.47e-04

TABLE VII
COMPARISON BETWEEN E-MRP-CNN AND OTHER MSVS METHODS ON MIR-1 K, WHERE "–" MEANS THAT CORRESPONDING RESULTS WERE NOT PROVIDED WITH

Method	Vocal			Acc		
	GNSDR	GSIR	GSAR	GNSDR	GSIR	GSAR
MLRR [43]	3.85	5.63	10.70	4.19	7.80	8.22
DRNN [42]	7.45	13.08	9.68	–	–	–
ModGD [53]	7.50	13.73	9.45	–	–	–
U-Net [11]	7.43	11.79	10.42	7.45	11.43	10.41
S-16-1-MIR	11.89	17.80	13.60	10.55	14.18	13.65
M-99-5-MIR	11.54	17.39	13.28	10.25	13.94	13.38

TABLE VIII
COMPARISON OF MEDIAN SDR VALUES BETWEEN E-MRP-CNN AND OTHER MSVS METHODS ON DSD100

Method	Vocal	Acc
DeepNMF [54]	2.75	8.90
wRPCA [55]	3.92	9.45
NUG [56]	4.55	10.29
BLEND [57]	5.23	11.70
MM-DenseNet [58]	6.00	12.10
S-31-1-DSD	6.15	12.60
M-99-7-DSD	6.42	12.64

was not reported in these compared methods, we only discuss separation performance. As shown in Tables VII-VIII, the E-MRP-CNN gave superior performance compared with these methods.

B. On MUSDB18 Dataset

We compared the E-MRP-CNN with Open-Unmix [44] on MUSDB18. Since the E-MRP-CNN is a monaural separation method, we estimated vocal and accompaniment respectively from the left-channel and right-channel of the mixture signal in MUSDB18. Different from Open-Unmix, which cropped the input spectrogram to 16 kHz for separation, the E-MRP-CNN cropped the input spectrogram to 8 kHz to speed up computation. In addition, the Open-Unmix used stronger data augmentation methods compared with our simple augmentation methods (gain and sliding). It also used normalization and input/output scalar to improve performance, which we did not use.

The results of the E-MRP-CNN and Open-Unmix (mono) are compared in Table IX, where the results of standard Open-Unmix (stereo), i.e., using stereo mixture for separation, are reported for reference. The E-MRP-CNN had lower Params but much higher FLOPs than Open-Unmix (mono). In the real environment, Open-Unmix (mono) had similar training and inferring speed as M-99-6-MUS. For separation performance, the E-MRP-CNN achieved better results than Open-Unmix (mono) and Open-Unmix (stereo) in SDR/SIR/SAR for Vocal and SDR/ISR/SAR for Acc.

We also conducted a statistical significance evaluation on the E-MRP-CNN and Open-Unmix, as shown in Table X. For both Vocal and Acc, all p-values were smaller than 0.05, suggesting that the E-MRP-CNN achieved significant improvement in separation performance compared with Open-Unmix (mono and stereo).

TABLE IX
COMPARISON BETWEEN E-MRP-CNN AND OPEN-UNMIX ON MUSDB18

Method	**Computational efficiency**				**Separation performance**							
	Params	FLOPs	Training	Inferring	Vocal				Acc			
	[M]	[G]	[bat./s]	[bat./s]	SDR	ISR	SIR	SAR	SDR	ISR	SIR	SAR
Open-Unmix (mono)	7.07	1.20	45.52	146.70	5.57	14.07	12.19	5.98	11.66	19.06	19.62	12.54
Open-Unmix (stereo)	8.89	0.76	118.86	348.25	5.57	**14.37**	12.48	5.72	11.88	18.95	**20.43**	12.29
S-17-1-MUS	6.26	266.96	16.19	48.71	6.36	13.61	13.40	**6.32**	**12.99**	**23.00**	16.18	**14.41**
M-99-6-MUS	4.31	41.00	47.69	159.11	**6.37**	12.08	**13.68**	6.05	12.74	22.49	15.73	14.07

TABLE X
KRUSKAL-WALLIS-BASED STATISTICAL SIGNIFICANCE EVALUATION OF SDR BETWEEN E-MRP-CNN AND OPEN-UNMIX (MONO AND STEREO)

Method		Acc		Vocal	
		H-value	p-value	H-value	p-value
S-17-1-MUS	Open-Unmix (mono)	238.38	8.89e-54	169.54	9.31e-39
	Open-Unmix (stereo)	214.35	1.54e-48	153.48	3.01e-35
M-99-6-MUS	Open-Unmix (mono)	103.25	2.96e-24	79.31	5.30e-19
	Open-Unmix (stereo)	87.20	9.80e-21	68.18	1.49e-16

IX. CONCLUSION

As the first attempt in the field of MSVS, we proposed an evolving framework, i.e., the E-MRP-CNN, to automatically find effective neural networks for MSVS. The E-MRP-CNN is based on a novel MR-CNN called MRP-CNN, which uses various-sized average pooling operators for feature extraction. Compared with current MR-CNNs, the MRP-CNN has a low computational complexity and can effectively extract multi-resolution features for MSVS. We derived the E-MRP-CNN using single-objective and multiple-objective genetic algorithms. The single-objective E-MRP-CNN takes into account only separation performance while the multi-objective E-MRP-CNN takes into account both separation performance and model complexity, thus providing a set of solutions to handle different separation performance and/or model complexity requirements. Experimental results on the MIR-1 K, DSD100, and MUSDB18 datasets indicate that the E-MRP-CNN (especially the multi-objective one) is more effective and efficient than the SOTA MSVS methods, verifying its effectiveness.

In this study, we took the linear-scaled STFT spectrogram as input. It is known that nonlinear-scaled spectrograms, e.g., Mel-based or constant-Q transform (CQT)-based spectrograms, approximate the auditory system better than the linear-scaled spectrogram; therefore, we will evaluate the E-MRP-CNN by taking nonlinear-scaled spectrogram as the input. We also intend to apply it to other audio-source-separation problems in our future work.

REFERENCES

[1] Z. Rafii, A. Liutkus, F. Stöter, S. I. Mimilakis, D. FitzGerald, and B. Pardo, "An overview of lead and accompaniment separation in music," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 26, no. 8, pp. 1307–1335, Aug. 2018.

[2] Y. Li and D. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 15, no. 4, pp. 1475–1487, May 2007.

[3] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] A. J. R. Simpson, G. Roma, and M. D. Plumbley, "Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network," in *Proc. 12th Int. Conf. Latent Var. Anal. Signal Separation*, 2015, pp. 429–436.

[5] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[6] E. M. Grais and M. D. Plumbley, "Single channel audio source separation using convolutional denoising autoencoders," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2017, pp. 1265–1269.

[7] W. Yuan, S. Wang, X. Li, M. Unoki, and W. Wang, "A skip attention mechanism for monaural singing voice separation," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1481–1485, Oct. 2019.

[8] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis," in *Proc. 11th Int. Soc. Music Inf. Ret. Conf.*, 2010, pp. 625–636.

[9] C. A. Huang *et al.*, "Music transformer: Generating music with long term structure," in *Proc. 7th Int. Conf. Learn. Representations*, 2019, pp. 1–15.

[10] S. Park, T. Kim, K. Lee, and N. Kwak, "Music source separation using stacked hourglass networks," in *Proc. 19th Int. Soc. Music Inf. Ret. Conf.*, 2018, pp. 289–296.

[11] A. Jansson, E. J. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *Proc. 18th Int. Soc. Music Inf. Ret. Conf.*, 2017, pp. 745–751.

[12] E. M. Grais, H. Wierstorf, D. Ward, and M. D. Plumbley, "Multi-resolution fully convolutional neural networks for monaural audio source separation," in *Proc. 14th Int. Conf. Latent Variable Anal. Signal Separation*, 2018, pp. 340–350.

[13] E. M. Grais, D. Ward, and M. D. Plumbley, "Raw multi-channel audio source separation using multi-resolution convolutional auto-encoders," in *Proc. 26th Eur. Signal Process. Conf.*, 2018, pp. 1577–1581.

[14] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[15] A. Stoutchinin, F. Conti, and L. Benini, "Optimally scheduling CNN convolutions for efficient memory access," *Computing Res. Repository (CoRR)*, vol. abs/1902.01492, 2019.

[16] H. Lee, P. T. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1096–1104.

[17] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6979–6983.

[18] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[19] 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," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 721–725.

[20] S. I. Mimilakis, K. Drossos, T. Virtanen, and G. Schuller, "A recurrent encoder-decoder approach with skip-filtering connections for monaural singing voice separation," in *Proc. Int. Workshop Mach. Learn. Signal Process.*, 2017, pp. 1–6.

[21] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *Proc. Conf. Int. Soc. Music Inf. Retrieval*, 2018, pp. 334–340.

[22] O. Slizovskaia, L. Kim, G. Haro, and E. Gómez, "End-to-end sound source separation conditioned on instrument labels," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 306–310.

[23] J. Liu and Y. Yang, "Dilated convolution with dilated GRU for music source separation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4718–4724.

[24] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[25] M. Michelashvili, S. Benaim, and L. Wolf, "Semi-supervised monaural singing voice separation with a masking network trained on synthetic mixtures," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 291–295.

[26] Y. C. Sübakan and P. Smaragdis, "Generative adversarial source separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 26–30.

[27] Y. Luo, Z. Chen, J. R. Hershey, J. L. Roux, and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 61–65.

[28] S. I. Mimilakis, K. Drossos, E. Cano, and G. Schuller, "Examining the mapping functions of denoising autoencoders in singing voice separation," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 28, pp. 266–278, Jan. 2020.

[29] D. So, Q. Le, and C. Liang, "The evolved transformer," in *Proc. 36th Int. Conf. Mach. Learn*., 2019, pp. 5877–5886.

[30] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, pp. 55:1–55:21, 2019.

[31] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, pp. 99–127, 2001.

[32] X. Gong, S. Chang, Y. Jiang, and Z. Wang, "Autogan: Neural architecture search for generative adversarial networks," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2019, pp. 3223–3233.

[33] Y. Chen *et al.*, "RENAS: Reinforced evolutionary neural architecture search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 4787–4796.

[34] Z. Lu *et al.*, "Nsga-Net: Neural architecture search using multi-objective genetic algorithm (*extended abstract*)," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2020, pp. 4750–4754.

[35] Z. Lu *et al.*, "Nsga-Net: Neural architecture search using multi-objective genetic algorithm," in *Proc. Genet. Evol. Comput. Conf.*, 2019, pp. 419–427.

[36] 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, Jul. 2006.

[37] K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, "A survey of fpga-based neural network inference accelerators," *TRETS*, vol. 12, no. 1, pp. 2:1–2:26, 2019.

[38] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

[39] C. Hsu and J. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1 K dataset," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 18, no. 2, pp. 310–319, Feb. 2010.

[40] N. Ono, Z. Rafii, D. Kitamura, N. Ito, and A. Liutkus, "The 2015 signal separation evaluation campaign," in *Proc. 12th Int. Conf. Latent Variable Anal. Signal Separation*, 2015, pp. 387–395.

[41] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[42] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[43] Y. Yang, "Low-rank representation of both singing voice and music accompaniment via learned dictionaries," in *Proc. 14th Int. Soc. Music Inf. Ret. Conf.*, 2013, pp. 427–432.

[44] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix—A reference implementation for music source separation," *J. Open Source Softw.*, vol. 4, no. 41, p. 1667, 2019.

[45] A. Liutkus and R. Badeau, "Generalized wiener filtering with fractional power spectrograms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 266–270.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015. pp. 1–15.

[48] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. 5th Int. Conf. Learn. Representations*, 2017, pp. 1–16.

[49] P. Goyal *et al.*, "Accurate, large minibatch SGD: training imagenet in 1 hour," *Computing Res. Repository (CoRR)*, vol. abs/1706.02677, 2017.

[50] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 15, no. 5, pp. 1564–1578, Jul. 2007.

[51] P. G. Hoel, *Elementary Statistics*. Hoboken, NJ, USA: John Wiley & Sons, 1976.

[52] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *J. Amer. Stat. Assoc.*, vol. 47, no. 260, pp. 583–621, 1952.

[53] J. Sebastian and H. A. Murthy, "Group delay based music source separation using deep recurrent neural networks," in *Proc. Int. Conf. Signal Process. Commun.*, 2016, pp. 1–5.

[54] J. L. Roux, J. R. Hershey, and F. Weninger, "Deep NMF for speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 66–70.

[55] I. Jeong and K. Lee, "Singing voice separation using RPCA with weighted l_1 -norm," in *Proc. 13th Int. Conf. Latent Variable Anal. Signal Separation*, 2017, pp. 553–562.

[56] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel music separation with deep neural networks," in *Proc. 24th Eur. Signal Process. Conf.*, 2016, pp. 1748–1752.

[57] S. Uhlich *et al.*, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 261–265.

[58] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *Proc. IEEE Workshop App. Signal Process. Audio Acoust.*, 2017, pp. 21–25.

Weitao Yuan received the B.S. and M.S. degrees from the School of Information Science and Engineering, Yanshan University, Qinhuangdao, China, in 2000 and 2003, respectively, and the Ph.D. degree from the School of Mathematical Sciences, Peking University (PKU), Beijing, China, in 2008. From 2009 to 2014, he was with Jinan University and Yanshan University. Since 2015, he has been with the School of Computer Science and Technology, Tiangong University, Tianjin, China. His current research interests include convex analysis, deep learning, and music source separation.

Bofei Dong is currently working toward the B.S. degree with the School of Computer Science and Technology, Tiangong University, Tianjin, China, since 2017. His current research interests include speech signal processing and deep learning.

Shengbei Wang received the B.S. and M.S. degrees in signal processing from Tianjin Polytechnic University, Tianjin, China, in 2009 and 2012, respectively, and the Ph.D. degree in information science from the Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2015. Her main research interests include signal processing, digital audio or speech watermarking, and deep learning-based source separation. She is currently an Associate Professor with the School of Computer Science and Technology with Tiangong University, Tianjin, China.

Masashi Unoki (Member, IEEE) received the M.S. and Ph.D. degrees from the Japan Advanced Institute of Science and Technology (JAIST), Nomi, Japan, in 1996 and 1999, respectively. From 1998 to 2001, he was a Research Fellow with Japan Society for the Promotion of Science. From 1999 to 2000, he was a Visiting Researcher with ATR Human Information Processing Laboratories, and from 2000 to 2001, he was a Visiting Research Associate with Centre for the Neural Basis of Hearing, Department of Physiology, University of Cambridge, Cambridge, U.K. Since 2001, he has been with the Faculty of the School of Information Science, JAIST, and is currently a Full Professor. He is a member of the Research Institute of Signal Processing, Institute of Electronics, Information and Communication Engineers of Japan, the Acoustical Society of America, the Acoustical Society of Japan, and the International Speech Communication Association. He was the recipient of the Sato Prize from the ASJ in 1999, 2010, and 2013 for an Outstanding Paper and the Yamashita Taro Young Researcher Prize from the Yamashita Taro Research Foundation in 2005.

Wenwu Wang (Senior Member, IEEE) received the B.Sc., M.E., and the Ph.D. degrees from the College of Automation, Harbin Engineering University, Harbin, China, in 1997, 2000, and 2002, respectively.

He was with King's College London, London, U.K., from 2002 to 2003, Cardiff University, Cardif, U.K., from 2004 to 2005, Tao Group Ltd. (now Antix Labs Ltd.), U.K., from 2005 to 2006, Creative Labs from 2006 to 2007, and in May 2007, before joining the University of Surrey, Guildford, U.K., where he is currently a Professor of signal processing and machine learning, and the Co-Director with the Machine Audition Lab, Centre for Vision Speech and Signal Processing, Guildford, U.K. He is also a Guest Professor with Tianjin University, Tianjin, China, and the Qingdao University of Science and Technology, Qingdao, China. He was a Visiting Scholar with Ohio State University, Columbus, OH, USA, in 2008. He has authored or coauthored more than 250 publications in his research fields, which include blind signal processing, sparse signal processing, audio–visual signal processing, machine learning and perception, artificial intelligence, machine audition (listening), and statistical anomaly detection.

He was the co-recipient of more than ten awards, including the Judges Award on the DCASE 2020, the Reproducible System Award on the DCASE 2019 and the DCASE 2020, the Best Student Paper Award on the LVA/ICA 2018, the Best Oral Presentation on the FSDM 2016, the TVB Europe Award for Best Achievement in Sound in 2016, and the Best Solution Award on the Dstl Challenge Under-sampled Signal Recognition in 2012. He achieved the first place in the 2020 DCASE Challenge on Urban Sound Tagging with Spatio-Temporal Context, and the first place in the 2017 DCASE Challenge on Large-scale Weakly Supervised Sound Event Detection for Smart Cars.

Since 2019, he has been a Senior Area Editor, and from 2014 to 2018, was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. Since 2020, he has been an Associate Editor for the IEEE/ACM TRANSACTIONS ON AUDIO SPEECH AND LANGUAGE PROCESSING, and since 2019, he has been an Associate Editor for the EURASIP *Journal on Audio Speech and Music Processing*. Since 2020, he has been the Specialty Editor-in-Chief of *Frontiers in Signal Processing*. He was the Publication Co-Chair for ICASSP 2019, Brighton, U.K. Since 2021, he has been elected as a member of the IEEE Signal Processing Theory and Methods Technical Committee and the IEEE Machine Learning for Signal Processing Technical Committee. Since 2019, he has also been a member of the International Steering Committee of Latent Variable Analysis and Signal Separation.