Time-Domain Audio Source Separation With Neural Networks Based on Multiresolution Analysis

Tomohiko Nakamura, *Member, IEEE*, Shihori Kozuka, and Hiroshi Saruwatari, *Member, IEEE*

***Abstract*—We propose a time-domain audio source separation method based on multiresolution analysis, which we call multiresolution deep layered analysis (MRDLA). The MRDLA model is based on one of the state-of-the-art time-domain deep neural networks (DNNs), Wave-U-Net, which successively down-samples features and up-samples them to have the original time resolution. From the signal processing viewpoint, we found that the down-sampling (DS) layers of Wave-U-Net cause aliasing and may discard information useful for source separation because they are implemented with decimation. These two problems are due to the decimation; thus, to achieve a more reliable source separation method, we should design DS layers capable of simultaneously overcoming these problems. With this motivation, focusing on the fact that the successive DS architecture of Wave-U-Net resembles that of multiresolution analysis, we develop DS layers based on discrete wavelet transforms (DWTs), which we call the DWT layers, because the DWTs have anti-aliasing filters and the perfect reconstruction property. We further extend the DWT layers such that their wavelet basis functions can be trained together with the other DNN components while maintaining the perfect reconstruction property. Since a straightforward trainable extension of the DWT layers does not guarantee the existence of anti-aliasing filters, we derive constraints for this guarantee in addition to the perfect reconstruction property. Through music source separation experiments including subjective evaluations, we show the efficacy of the proposed methods and the importance of simultaneously considering both the anti-aliasing filters and the perfect reconstruction property.**

***Index Terms*—Time-domain audio source separation, multiresolution analysis, discrete wavelet transform, deep neural networks.**

I. Introduction

AUDIO source separation is a technique of extracting individual source signals from an observed mixture, which has wide applications including automatic music transcription and music remixing according to user preferences. The recent advent of deep neural networks (DNNs) has notably improved the performance of audio source separation in supervised settings [1]. Most DNN-based methods perform source separation in the magnitude or power spectrogram domain [2]–[5]. Despite their success, the methods have several drawbacks. They ignore phase information in the separation process, which may result in suboptimal solutions. Furthermore, to reconstruct the separated time-domain signals, the methods typically use the noisy phase of the mixture, which may be inconsistent with the separated magnitude or power spectrogram of each source, i.e., no corresponding time-domain signal is guaranteed to exist [6]. Although several attempts to estimate an adequate phase have been made thus far [7]–[10], recent studies have shown that one promising direction is to adopt an end-to-end approach, which avoids phase estimation and directly deals with time-domain signals [11]–[16].

End-to-end DNNs can be roughly categorized into two approaches. One approach imitates the commonly used separation procedure in the spectrogram domain [12], [14], [16]. In this approach, the DNN architecture consists of a pair of an encoder and a decoder, which correspond to time-frequency transform and its inverse, respectively, and a mask estimator.

The other approach, unlike the above separation procedure, performs the separation in the time domain [11], [15]. Wave-U-Net is one of the state-of-the-art DNNs categorized into this approach [11], and it is a time-domain adaptation of the U-net architecture [17]. The key idea of this architecture is to effectively capture the long-term dependence of data by successively making the time resolution of features coarser and increasing the receptive fields of convolution layers located between down-sampling (DS) layers. Wave-U-Net consists of an encoder and a decoder. The encoder successively decimates features with DS layers, which we call the decimation layer to distinguish it from the layer we introduce in this paper. The decoder up-samples the features with linear up-sampling (US) layers such that the output of the decoder has the same time resolution as the input. The decoder can access the feature before the decimation at the same hierarchy through so-called skip connections.

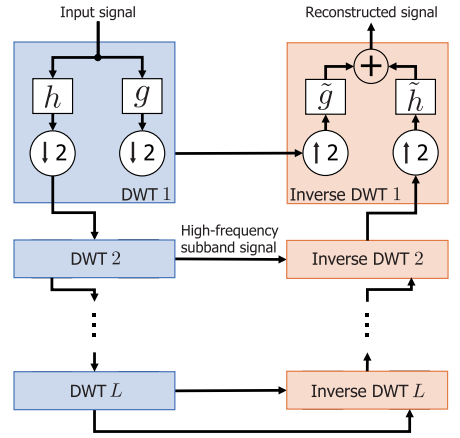
From the signal processing viewpoint, DNNs and features can be interpreted as stacked nonlinear systems and signals propagated inside these systems, respectively. This interpretation leads to the identification of the following two problems caused by decimation layers.

The decimation layers simply decimate the features, which apparently causes aliasing in the feature domain according to the Nyquist–Shannon sampling theorem. We call this aliasing the feature-domain aliasing to distinguish it from the aliasing

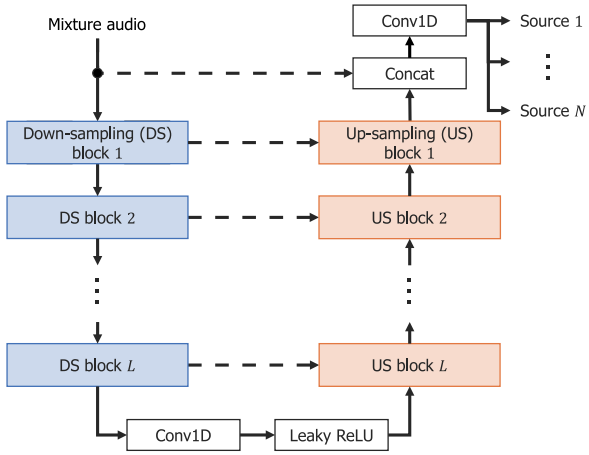
Manuscript received September 18, 2020; revised December 30, 2020 and March 3, 2021; accepted April 6, 2021. Date of publication April 13, 2021; date of current version May 19, 2021. This work was supported in part by JSPS KAKENHI under Grants JP19H01116 and JP20K19818, in part by JSPS-CAS Joint Research Program under Grant JPJSBP120197203, in part by Research Grant A of the Tateisi Science and Technology Foundation, and in part by the Research Grant of Kawai Foundation for Sound Technology and Music. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Isabel Barbancho. *(Corresponding author: Tomohiko Nakamura.)*

The authors are with the Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan (e-mail: tomohiko.nakamura.jp@ieee.org; kozuka-shihori001@g.ecc.u-tokyo.ac.jp; hiroshi_saruwatari@ipc.i.u-tokyo.ac.jp).

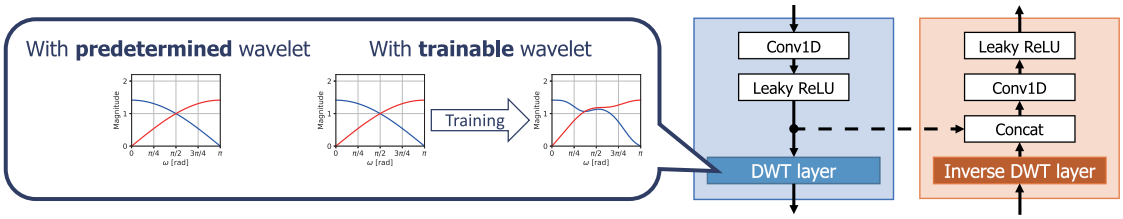
Digital Object Identifier 10.1109/TASLP.2021.3072496



(a) Multiresolution analysis and synthesis with L levels.



(b) MRDLA with L levels for N sources.



(c) DS and US blocks at each hierarchy.

Fig. 1. Illustrations of (a) multiresolution analysis, (b) proposed MRDLA model, and (c) architecture of DS and US blocks. The blue and orange regions represent DS and US modules, respectively. h and g ($\tilde{h}$ and $\tilde{g}$) respectively denote low- and high-pass (reconstruction) filters corresponding to the DWT (inverse DWT). "Conv1D," "Concat," and "Leaky ReLU" denote a one-dimensional convolution layer, a channel concatenation of two inputs, and a leaky rectified linear unit layer, respectively. The dashed lines represent skip connections.

of time-domain signals. Recent studies have shown that the feature-domain aliasing degrades the performance in audio recognition tasks [18] and image processing [19], which may be true for audio source separation, as we will experimentally show in Section V-E. One straightforward solution to reduce the feature-domain aliasing is to insert anti-aliasing filters before the decimation layers, which has been presented for image classification [20], [21]. In [21], the insertion of anti-aliasing filters has improved the image classification accuracy and has made the entire DNN robust against the diagonal shift of an input image. However, the other problem remains unsolved.

The decimation layers, even with the anti-aliasing filters, discard parts of input features. The discarded components may contain information useful for source separation, particularly in the early training stage, which may degrade the separation performance, as we will experimentally show in Section V-E. Although the skip connection allows the decoder to access the feature before DS, there are no guarantees that the following convolution layer always uses the feature coming from the skip connection. Since the weight of this convolution layer is determined by training, whether the lack of such information can be reduced depends on training.

The two problems may be alleviated if we can train Wave-U-Net as we wish, but there are no theoretical guarantees that such training can be carried out. Since the problems are due to decimation operations, to develop a more reliable source separation method, we should design a novel DS layer whose architecture ensures the reduction in feature-domain aliasing and the preservation of the entire information of input features.

To design such a DS layer, we focus on the fact that the successive DS architecture of Wave-U-Net resembles that of multiresolution analysis [22] (see Fig. 1(a)). Multiresolution analysis repeatedly down-samples a signal with a discrete wavelet transform (DWT), which decomposes the signal into low- and high-frequency subband signals with half the time resolution. By repeatedly applying an inverse DWT to subband signals, we can perfectly reconstruct the input signal, which shows that two subband signals obtained with the DWT include the entire information of an input signal. Since a DWT can be seen as a two-channel filterbank that consists of a pair of low- and high-pass filters and satisfies the perfect reconstruction property, the use of the DWT for DS allows us to overcome the two problems simultaneously.

On the basis of this idea, we develop DWT-based DS layers, which we call the DWT layers, to build a time-domain audio source separation method, multiresolution deep layered analysis (MRDLA) (see Fig. 1(b)). We also devise the US layers corresponding to the DWT layer, which we call the inverse DWT layer. The DWT layers can be efficiently implemented with a computation technique of DWTs called the lifting scheme [23], which also makes it easier to change wavelet basis functions. Since the basis functions determine the frequency responses of the filters of the DWTs, they may affect the separation performance of MRDLA. We further extend the DWT layers such that

their wavelet basis functions can be trained together with the other DNN components. However, only making the parameters of the DWT layers trainable cannot guarantee that this trainable extension has the anti-aliasing filters. For this purpose, we derive constraints that guarantee the existence of the anti-aliasing filters and the perfect reconstruction property even during training.

Additionally, we make an architectural change of the output layer of MRDLA for better performance. Wave-U-Net has a constraint that the sum of all source estimates equals the input signal. However, since this constraint restricts the range of the possible values of the source estimates during training, the estimation failure of one of the instruments may spill over to the estimations of the other instruments, which leads to performance degradation. For this reason, we do not introduce this constraint for MRDLA.

The contributions of this paper are summarized as follows:

- From the signal processing viewpoint, we found that decimation layers lack anti-aliasing filters for features and discard parts of them.
- Focusing on the resemblance of the successive DS architecture between Wave-U-Net and multiresolution analysis, we propose a time-domain audio source separation method, MRDLA, by developing the DWT and inverse DWT layers, which reduce the feature-domain aliasing and preserve the entire information of the features.
- We extend the DWT layers such that their wavelet basis functions are jointly trainable with the other DNN components.
- We derive constraints of the trainable DWT layer so that they have the anti-aliasing filters, in addition to the perfect reconstruction property, even during training.
- Through experiments on music source separation, we show the efficacy of the proposed models and the importance of simultaneously considering the anti-aliasing filters and the perfect reconstruction property for designing the DS layers.

Note that commonly used DS layers (e.g., max pooling, average pooling, and strided convolution layers) lack either anti-aliasing filters or the perfect reconstruction property; thus, our proposed layers are beneficial for not only Wave-U-Net but also various research fields using DNNs.

The rest of this paper is organized as follows: In Section II, we clarify the relationships of the proposed layers and MRDLA with conventional DS layers and time-domain audio source separation methods, and review the lifting scheme. In Section III, we describe the motivations to develop the DWT and inverse DWT layers, and establish the MRDLA model. In Section IV, we extend the proposed layers to trainable ones and derive constraints to guarantee the existence of anti-aliasing filters in addition to the perfect reconstruction property. We conducted experimental evaluations on music source separation to show the efficacy of the MRDLA models in Section V and finally conclude this paper in Section VI.

Note that this paper is partially based on our international conference papers [24], [25]. This paper has the following seven additional contributions: (i) Although the DWT layer we presented in the conference paper is designed only for predetermined wavelets, we extend it to those jointly trainable with the other DNN modules. (ii) Since a straightforward trainable extension of the DWT layer has no guarantees to have low- and high-pass filters, we derive constraints to guarantee the existence of these filters in addition to the perfect reconstruction property. (iii) We entirely redesign experiments with modern data augmentation techniques, which we experimentally found to improve the separation performance, and (iv) evaluate the proposed DWT layer having predetermined wavelets and its trainable extensions with various model sizes. (v) We also assess the effects of guaranteeing the trainable DWT layers to have low- and high-pass filters during training through comparisons of weight initialization and architectures of these layers. (vi) We conducted objective and subjective experiments of the proposed method with conventional time-domain audio source separation methods to evaluate the effectiveness and perceptual quality of MRDLA, using statistical tests. (vii) We further show the details of the architectures of MRDLA and Wave-U-Net and those of the implementation of the DWT layer and its relationship with conventional DS layers, which are omitted in the conference papers due to space limitation.

II. Related Works

A. Conventional DS Layers

The performance degradation caused by the feature-domain aliasing has been observed in both audio processing [18] and image processing [19]. One straightforward method of reducing the feature-domain aliasing is to insert anti-aliasing filters, which has been adopted in previous studies for image processing [20], [21], [26]. Although one of such studies has shown a wavelet-based pooling layer called the wavelet pooling [26], it outputs only the second-order wavelet subband signals of a feature; hence, it lacks the perfect reconstruction property. In contrast, our proposed layers not only include anti-aliasing filters but also satisfy the perfect reconstruction property.

The squeezing operation has been used in a normalizing flow model for image generation [27]. The time-domain adaptation of this operation simply splits the feature into the even- and odd-sample components in time, and concatenates them along the channel axis. This operation has the perfect reconstruction property but apparently lacks anti-aliasing filters. Note that the relationship between the squeezing operation and the proposed layer will be shown in Section III-B2.

The subpixel convolution layer, a convolution layer followed by the inverse of the squeezing process, has been presented to reduce checkerboard artifacts caused by the US process using the transposed convolution layer for US [28]. Since the aliasing of our interests occurs in DS, the checkerboard artifacts are beyond the scope of our manuscript. Although we can consider a DS version of this US layer, it is defined by the squeezing operation followed by a convolution layer and is essentially the same as the squeezing operation. The use of this DS layer increases the number of parameters of the DNNs, which makes it difficult to distinguish the effect of this layer from that of the increase in model size. Thus, we did not consider it for the comparison.

For image processing, an invertible DNN named i-Revnet has been developed [29] and has recently been used as a trainable time-frequency transform of a DNN-based speech enhancement

system [30]. i-Revnet alternately performs the squeezing operation and the application of nonlinear functions to only half of each output of the squeezing operation. Although i-Revnet has the perfect reconstruction property, it lacks anti-aliasing filters. The network uses nonlinear functions for the components corresponding to the prediction operators, which makes it more difficult to discuss the filter characteristics of the components. The network requires that the number of channels of the features is the same for the forward and inverse paths; however, the Wave-U-Net architecture does not meet this requirement. Since changing the DNN architecture makes it difficult to distinguish the effects of the lack of anti-aliasing filters from those of the DNN architecture, we omitted the comparison of MRDLA with i-Revnet in Section V. We instead compared MRDLA with a Wave-U-Net variant that uses the time-domain adaptation of the squeezing operations as the DS layers, which allows us to reveal the effects of the lack of anti-aliasing filters.

B. Time-Domain Audio Source Separation

As described in Section I, Wave-U-Net uses the decimation layers for DS, which causes the feature-domain aliasing and the lack of the perfect reconstruction property. Similarly to MRDLA and Wave-U-Net, a noncausal extension of WaveNet, proposed in [15], directly performs separation in the time domain. This model uses dilated convolution layers with exponentially increasing dilation factors to capture the long-term dependence of music signals. Although the use of a dilated convolution layer can avoid the lack of the perfect reconstruction property, it requires a large amount of memory for training since this layer keeps the time steps of features. This layer is identical to a layer consisting of the following two steps: (i) converting a feature into a down-sampled feature with the same number of subbands as the dilation factor by using the squeezing operations and (ii) applying the convolution operations to each subband of the feature using the weights shared by all subbands. The layer thus causes the feature-domain aliasing unless the weights are adequately trained.

For speech separation, Conv-TasNet is one of the state-of-the-art DNNs in the approach of imitating the conventional spectrogram-domain separation procedure [14]. However, since it uses the dilated convolution layers with exponentially increasing dilation factors, the same problems as those of the above-mentioned WaveNet extension occur.

C. Lifting Scheme

A DWT is naively implemented as two concurrent finite impulse response (FIR) filters followed by decimation with a factor of 2. To efficiently implement a DWT, we can use the lifting scheme [31], which factorizes a DWT into a sequence of FIR filters. This scheme can be computed in an in-place manner and can reduce the computational complexity.

Let us consider a time-domain signal of length T, $\boldsymbol{x} \in \mathbb{R}^T$. For simplicity, we assume that T is even. The lifting scheme consists of four steps: time-split, prediction, update, and scaling steps. Firstly, the time-split step splits an input signal $\boldsymbol{x}$ into the even- and odd-sample signals, respectively denoted by $\boldsymbol{x}^{(\text{even})} \in \mathbb{R}^{T/2}$ and $\boldsymbol{x}^{(\text{odd})} \in \mathbb{R}^{T/2}$. We denote this operation as a split operator $\mathcal{S}$. Secondly, the prediction step predicts $\boldsymbol{x}^{(\text{odd})}$ with a linear FIR filter called the prediction operator $\mathcal{P}$ and outputs $\boldsymbol{x}^{(\text{even})}$ and a prediction error $\boldsymbol{d} \in \mathbb{R}^{T/2}$:

$$\boldsymbol{d} = \boldsymbol{x}^{(\text{odd})} - \mathcal{P}\boldsymbol{x}^{(\text{even})}. \tag{1}$$

Thirdly, to reduce the aliasing caused by the time-split step, the update step smoothens $\boldsymbol{x}^{(\text{even})}$ with a linear FIR filter called the update operator $\mathcal{U}$:

$$\boldsymbol{c} = \boldsymbol{x}^{(\text{even})} + \mathcal{U}\boldsymbol{d}, \tag{2}$$

where $\boldsymbol{c} \in \mathbb{R}^{T/2}$ is the smoothed even-sample signal. Finally, the scaling step scales $\boldsymbol{c}$ and $\boldsymbol{d}$ by a normalization constant $A > 0$ and its reciprocal, respectively. We hereafter write the scaled versions of $\boldsymbol{c}$ and $\boldsymbol{d}$ as $\tilde{\boldsymbol{c}}$ and $\tilde{\boldsymbol{d}}$, respectively. The above process only uses one prediction operator and one update operator. However, the number of prediction and update operators, namely, I, may increase, depending on the chosen wavelet; these operators may differ at the prediction and update steps, respectively. We hereafter add a subscript i to $\boldsymbol{c}, \boldsymbol{d}, \tilde{\boldsymbol{c}}, \tilde{\boldsymbol{d}}, \mathcal{P}$, and $\mathcal{U}$.

Since all the steps are apparently invertible, the architecture of the lifting scheme itself guarantees the perfect reconstruction property. Note that the perfect reconstruction property does not depend on the prediction and update operators, and we can instead use nonlinear functions for these operators. As described in Section II-A, this nonlinear extension is beyond the scope of this paper and we leave it as our future work.

On the other hand, whether the lifting scheme has the low- and high-pass filters depends on the choice of the prediction and update operators. In the rest of this section, we describe the relationship between the filters and the operators in the z-transform domain. Let $X^{(\text{even})}(z), X^{(\text{odd})}(z), \tilde{C}_i(z)$, and $\tilde{D}_i(z)$ denote the z-transforms of $\boldsymbol{x}^{(\text{even})}, \boldsymbol{x}^{(\text{odd})}, \tilde{\boldsymbol{c}}_i$ and $\tilde{\boldsymbol{d}}_i$, respectively. Since each of the prediction, update, and scaling steps can be represented as a 2×2 matrix, the output of the lifting scheme can be described using these matrices [32]:

$$\begin{bmatrix} \tilde{C}_I(z) \\ \tilde{D}_I(z) \end{bmatrix} = Q_I(z) \begin{bmatrix} X^{(\text{even})}(z) \\ X^{(\text{odd})}(z) \end{bmatrix}, \tag{3}$$

$$Q_I(z) = \underbrace{\begin{bmatrix} A & 0 \\ 0 & 1/A \end{bmatrix}}_{\text{Scaling step}} \prod_{i=1}^{I} \left(\underbrace{\begin{bmatrix} 1 & U_i(z) \\ 0 & 1 \end{bmatrix}}_{i\text{th update step}} \underbrace{\begin{bmatrix} 1 & 0 \\ -P_i(z) & 1 \end{bmatrix}}_{i\text{th prediction step}} \right), \tag{4}$$

where $\prod_i$ denotes a sequence of matrix multiplications over i. Importantly, it has been proven that a DWT with arbitrary FIR filters can be factorized into the above form [32].

Now, we derive another form of Eq. (4), starting with the z-transforms of the low- and high-pass filters of the corresponding DWT, say $H_I(z)$ and $G_I(z)$. Without the loss of generality, $H_I(z)$ ($G_I(z)$) can be represented with the even-order filter $H_I^{(\text{even})}(z)$ ($G_I^{(\text{even})}(z)$) and the odd-order filter $H_I^{(\text{odd})}(z)$ ($G_I^{(\text{odd})}(z)$):

$$H_I(z) = H_I^{(\text{even})}(z^2) + z^{-1} H_I^{(\text{odd})}(z^2), \tag{5}$$

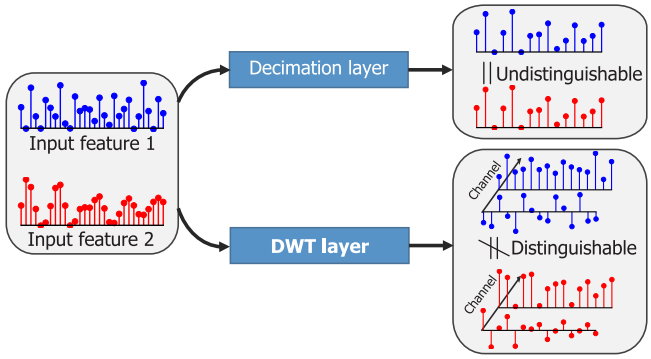


Fig. 2. Illustrative examples of down-sampled features obtained with decimation and DWT layers.

$$G_I(z) = G_I^{(\text{even})}(z^2) + z^{-1}G_I^{(\text{odd})}(z^2), \tag{6}$$

which are respectively called the polyphase representations of $H_I(z)$ and $G_I(z)$ (see [33] for details). With these notations, the DWT can be written as follows [32]:

$$\begin{bmatrix} \tilde{C}_I(z) \\ \tilde{D}_I(z) \end{bmatrix} = \begin{bmatrix} H_I^{(\text{even})}(z) & H_I^{(\text{odd})}(z) \\ G_I^{(\text{even})}(z) & G_I^{(\text{odd})}(z) \end{bmatrix} \begin{bmatrix} X^{(\text{even})}(z) \\ X^{(\text{odd})}(z) \end{bmatrix}. \tag{7}$$

Since the first matrix of the right-hand side of Eq. (7) equals $Q_I(z)$, we can confirm that $H_I(z)$ and $G_I(z)$ are parametrized by $P_i(z)$ and $U_i(z)$.

III. Multiresolution Deep Layered Analysis

A. Motivation and Strategy

In this section, we describe the motivation and strategy to develop the proposed DWT layer. As described in Section I, we found, from the signal processing viewpoint, that the decimation layers lack the anti-aliasing filters for the features and discard parts of input features.

Fig. 2 shows illustrative examples of down-sampled features. As shown in this figure, the decimation layer can output the same features for different input features owing to the feature-domain aliasing. Once the same features are obtained, the layers at hierarchies higher than the decimation layer cannot distinguish them, which means that these layers do not affect the discarded components. Thus, there is no choice but to drive only the layers at the lower hierarchies to handle the components discarded by the decimation layer.

The feature-domain aliasing can be alleviated by introducing the low-pass filters before the decimation layers, but the other problem remains unsolved. If the layers preceding the decimation layers can completely pack information useful for source separation into the decimated features, i.e., the information in the discarded components can be recovered from the decimated features, the model works well. However, whether such layers can be obtained strongly depends on the training. Although the skip connections allow the decoder to access the features inputted to the decimation layers, there are no guarantees that the following convolution layers always use the features coming from the skip connections. This is because the weight of the following convolution layer is determined by training and there is no constraint that this layer always uses the features coming from these connections. For example, the trained weight may be zeros for the features coming from the skip connections. To access the high-frequency components of the features coming from the skip connections, at every channel corresponding to these features, the trained weights must be high-pass frequency characteristics. In addition, the convolution layer, owing to its translation invariance, processes the even- and odd-indexed components of the feature coming from the skip connection in exactly the same manner, despite the fact that only the odd-indexed components are discarded by the decimation layer. This fact clearly means that the convolution layer itself cannot identify which elements of the feature are discarded or not.

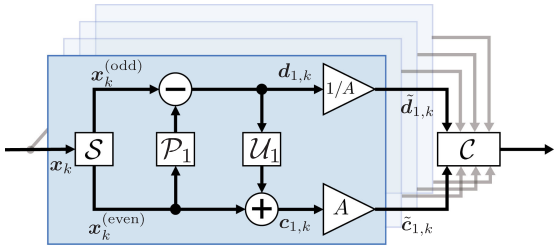
Since these two problems are caused by the decimation operation, we take an approach of designing a DS layer that is guaranteed to reduce the feature-domain aliasing and preserve the entire information of the feature. Inspired by the resemblance of the successive DS architecture between Wave-U-Net and multiresolution analysis, we use the DWT as a DS operation, which has an anti-aliasing filter and the perfect reconstruction property. Owing to these characteristics, the DWT outputs distinguishable features for different input features unlike the decimation layer, as shown in Fig. 2. This means that the DWT can preserve the entire information of the input feature.

B. DWT Layer

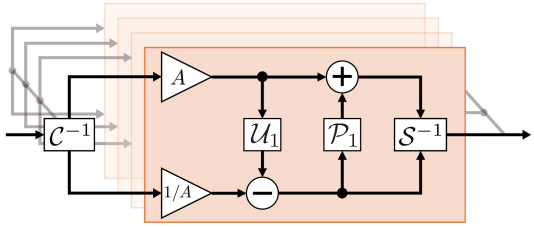
1) Implementation With Lifting Scheme: In this section, we describe an efficient implementation of the DWT layer. The DWT layer first applies the DWT to each channel slice of a feature using the lifting scheme [31] and concatenates the low- and high-frequency components obtained with the DWT along the channel axis. For simplicity, we consider the lifting scheme with $I = 1$, but the following discussion can be easily extended for $I > 1$.

Let us consider the feature of K channels and T time length and that T is even for simplicity. We denote a feature channel index as $k = 1, \ldots, K$. To avoid abuse of notations, we use the same notations used in Section II-C for the variables of the lifting scheme part of the DWT layer except for adding a subscript k to these variables. By interpreting the kth channel slice of the feature, $\boldsymbol{x}_k \in \mathbb{R}^T$, as a time-domain signal of length T, we can apply the lifting scheme to it. As described in Section II-C, $\boldsymbol{x}_k$ is split into the even- and odd-sample components, respectively denoted by $\boldsymbol{x}_k^{(\text{even})} \in \mathbb{R}^{T/2}$ and $\boldsymbol{x}_k^{(\text{odd})} \in \mathbb{R}^{T/2}$, in the time-split step. Subsequently, the prediction and update steps output the high- and low-frequency components $\boldsymbol{d}_{1,k} \in \mathbb{R}^{T/2}$ and $\boldsymbol{c}_{1,k} \in \mathbb{R}^{T/2}$ with the prediction and update operators $\mathcal{P}_1$ and $\mathcal{U}_1$, respectively. Finally, the scaling step scales $\boldsymbol{c}_{1,k}$ and $\boldsymbol{d}_{1,k}$ are scaled by the normalization constant A and its reciprocal to yield their scaled versions $\tilde{\boldsymbol{c}}_{1,k} \in \mathbb{R}^{T/2}$ and $\tilde{\boldsymbol{d}}_{1,k} \in \mathbb{R}^{T/2}$, respectively. The lifting scheme part is summarized as follows:

$$\begin{aligned} \text{Time-split step:} \quad & \boldsymbol{x}^{(\text{even})}, \boldsymbol{x}^{(\text{odd})} = \mathcal{S}\boldsymbol{x}, \\ \text{Prediction step:} \quad & \boldsymbol{d}_{1,k} = \boldsymbol{x}_k^{(\text{odd})} - \mathcal{P}_1 \boldsymbol{x}_k^{(\text{even})}, \\ \text{Update step:} \quad & \boldsymbol{c}_{1,k} = \boldsymbol{x}_k^{(\text{even})} + \mathcal{U}_1 \boldsymbol{d}_{1,k}, \end{aligned}$$



(a) DWT layer.



(b) Inverse DWT layer.

Fig. 3. Block diagrams of proposed layers. The blue and orange regions respectively correspond to the lifting scheme and its inverse. $\mathcal{C}^{-1}$ and $\mathcal{S}^{-1}$ denote the inverse operations of $\mathcal{C}$ and $\mathcal{S}$, respectively.

$$\text{Scaling step:}\quad \tilde{\boldsymbol{d}}_{1,k} = \frac{\boldsymbol{d}_{1,k}}{A},\quad \tilde{\boldsymbol{c}}_{1,k} = A\boldsymbol{c}_{1,k}.$$

After the lifting scheme, the DWT layer concatenates $\tilde{\boldsymbol{c}}_{1,k}$ and $\tilde{\boldsymbol{d}}_{1,k}$ for all channels along the channel axis to form a down-sampled version of the input feature, which has $2K$ channels and $T/2$ time length. We call this step the channel concatenation step and denote its operation by $\mathcal{C}$. The overall block diagram of the DWT layer is shown in Fig. 3(a). Since the lifting scheme and the channel concatenation step are apparently invertible, the DWT layer has the perfect reconstruction property. Owing to this property, we can define the corresponding US layer, i.e., the inverse DWT layer, by the reverse process of the DWT layer, whose block diagram is shown in Fig. 3(b).

For odd T, we insert a padding layer before the time-split step, which pads the last time entry of the feature and makes the padded feature to have an even time length. In this case, we remove the last time entry of the output of the inverse DWT layer at the same hierarchy to keep the correct time length. Through preliminary experiments, we observed no marked difference in separation performance between zero and reflection paddings, and decided to use the reflection padding.

Compared with the decimation and linear US layers, the DWT and inverse DWT layers require additional computations for the prediction, update, and scaling steps. However, the use of these layers does not significantly increase the processing time. Since the prediction and update operators can be implemented as the usual convolution layers without bias terms, these computations are easily parallelizable at each step. Furthermore, the proposed layers consist of differentiable operations, which makes it easier to implement them with modern deep learning frameworks, e.g., PyTorch and TensorFlow, owing to their automatic differentiation mechanisms.

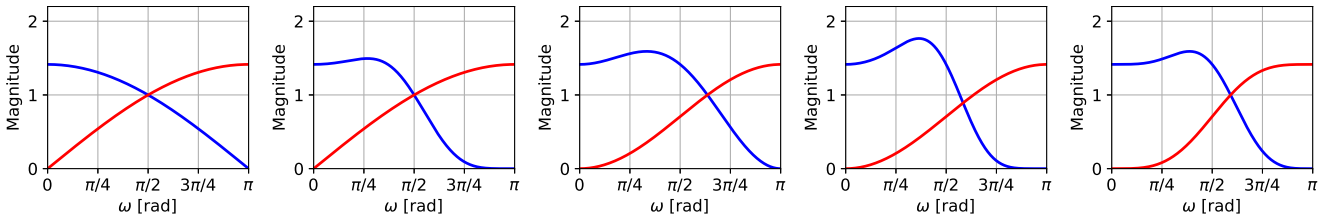
2) Relationship With Conventional DS Layers: We here describe the relationship of the DWT layer with the average pooling layer and the squeezing operation [27], with which we will compare the proposed layers in Section V. Here, let us consider the DWT layer whose $\mathcal{P}_1$ is an all-stop filter, $\mathcal{U}_1$ outputs an input as is, and A equals 2. In this case, the low-frequency component $\tilde{\boldsymbol{c}}_{1,k}$ of this layer is computed as $\boldsymbol{c}_{1,k} = (\boldsymbol{x}_{1,k}^{(\mathrm{odd})} + \boldsymbol{x}_{1,k}^{(\mathrm{even})})/2$, which is the same operation as that for the average pooling layer with a kernel size of 2 and a stride of 2. We can reduce the DWT layer to the average pooling layer by discarding the high-frequency component $\tilde{\boldsymbol{d}}_{1,k}$ before the concatenation step and directly outputting the low-frequency component $\tilde{\boldsymbol{c}}_{1,k}$. Thus, the average pooling layer has an anti-aliasing filter but lacks the perfect reconstruction property.

If $\mathcal{P}_1$ and $\mathcal{U}_1$ are all-stop filters and $A = 1$, the DWT layer consists of only the time-split and concatenation steps, which is the same as the squeezing operation. These two steps are invertible, and the squeezing operation has the perfect reconstruction property. However, since the outputs of the time-split step, $\boldsymbol{x}_k^{(\mathrm{even})}$ and $\boldsymbol{x}_k^{(\mathrm{odd})}$, are aliased, the squeezing operation lacks the anti-aliasing filter.

C. Architecture

By using the proposed DWT and inverse DWT layers, we build the MRDLA model on the basis of the best architecture of Wave-U-Net reported in [11]. Figs. 1(b) and (c) show schematic illustrations of the architecture of the MRDLA model, which features an encoder–decoder architecture with L levels, a bottleneck block, and an output block. Let the level index be $l = 1, \ldots, L$, the number of sources be N, and the number of channels of the input signal be $C^{(\mathrm{in})}$. The encoder first takes a mixture audio signal and successively down-samples it and features with L DS blocks. DS block l consists of a convolution layer with $C^{(\mathrm{e})}l$ filters of size $f^{(\mathrm{e})}$ and the DWT layer. The bottleneck block processes the output of DS block L with a convolution layer with $C^{(\mathrm{m})}$ filters of size $f^{(\mathrm{e})}$. The decoder then up-samples the features with L US blocks, accessing the inputs of the DWT layers at the same hierarchy through the skip connections from the encoder. US block l up-samples a feature with the inverse DWT layer and passes it to a convolution layer with $C^{(\mathrm{d})}l$ filters of size $f^{(\mathrm{d})}$. Ahead of the convolution layer, this block concatenates the up-sampled feature and the input of the DWT layer of DS block l after cropping the latter in time to match the time lengths of the two inputs. The above convolution layers are followed by leaky ReLU activations with a leakiness of 0.2. The decoder is finally followed by a convolution layer with $C^{(\mathrm{in})}$ filters of size 1, which outputs N source estimates with $C^{(\mathrm{in})}$ channels. The convolution layers are intentionally without any paddings since zero padding may cause artifacts at the edges of the estimated audio signals [11].

The detailed architecture is summarized in Table I, where LeakyReLU and Tanh denote the leaky ReLU and the tangent hyperbolic function, respectively. Here, Conv1D(a, b) denotes a one-dimensional convolution layer with a filters of size b, and Input and DS feature l respectively represent the input signal and the feature before the DWT layer of DS block l. Concat(DS feature l) denotes the concatenation of the output of the preceding layer and DS feature l.



(a) Haar wavelet. (b) CDF(1,5) wavelet. (c) CDF(2,2) wavelet. (d) CDF(2,6) wavelet. (e) DD4 wavelet.

Fig. 4. Frequency responses of Haar, CDF, and DD wavelets. The first and second numbers in parentheses following "CDF" represent the multiplicity of zeros of the analysis and synthesis high-pass filters at $z = 1$, respectively. The blue and red curves respectively correspond to the low- and high-pass filters.

TABLE I
DETAILED ARCHITECTURES OF MRDLA MODEL AND WAVE-U-NET

Block	MRDLA	Wave-U-Net [11]
	Input	Input
DS blocks for $l = 1$ to L	Conv1D($C^{(\mathrm{e})}l, f^{(\mathrm{e})}$) LeakyReLU **DWT layer**	Conv1D($C^{(\mathrm{e})}l, f^{(\mathrm{e})}$) LeakyReLU **Decimation layer**
Bottleneck block	Conv1D($C^{(\mathrm{m})}, f^{(\mathrm{e})}$) LeakyReLU	Conv1D($C^{(\mathrm{m})}, f^{(\mathrm{e})}$) LeakyReLU
US blocks for $l = L$ to 1	**Inverse DWT layer** Concat(DS feature l) Conv1D($C^{(\mathrm{d})}l, f^{(\mathrm{d})}$) LeakyReLU	**Linear US layer** Concat(DS feature l) Conv1D($C^{(\mathrm{d})}l, f^{(\mathrm{d})}$) LeakyReLU
Output block	Concat(Input) Conv1D($NC^{(\mathrm{in})}, 1$) -	Concat(Input) Conv1D($(N-1)C^{(\mathrm{in})}, 1$) Tanh

In the estimation phase, the MRDLA model works as Wave-U-Net. Let us denote the input and output signal lengths of the MRDLA model as $T^{(\mathrm{out})}$ and $T^{(\mathrm{in})}$, respectively, where $T^{(\mathrm{out})} < T^{(\mathrm{in})}$ since the convolution layers do not use any padding. We take overlapping audio segments of $T^{(\mathrm{in})}$ lengths with a hopsize of $T^{(\mathrm{out})}$ from a mixture audio signal, apply the MRDLA model to these segments independently, and concatenate them to obtain the estimated source signals.

D. Modifications of Output Block

If we respectively replace the DWT and inverse DWT layers with the decimation and linear US layers, the proposed model is reduced to Wave-U-Net except for the following two modifications. One modification is to change the output of Wave-U-Net from $N-1$ estimates to N estimates. As shown in Table I, the original Wave-U-Net outputs the Nth source estimate by subtracting the sum of $N-1$ source estimates from the input signal to ensure that the sum of N source estimates always equals the input signal. However, since this output method restricts the range of the possible values of the source estimates during training, the estimation failure of one of the instruments heavily affects the estimations of other instruments, which may result in performance degradation. We experimentally found this modification to improve the separation performance.

The other modification is to remove the tangent hyperbolic function followed by the last convolution layer of Wave-U-Net. As one of the data augmentation techniques, we used data standardization, which normalizes a time-domain signal of each track so that its mean and variance are zero and one, respectively. This technique may increase the values of the training audio signals outside a range of $[-1, 1]$, although the tangent hyperbolic function squashes the values into $[-1, 1]$.

IV. EXTENSION TO DWT LAYERS WITH TRAINABLE WAVELET BASIS FUNCTIONS

A. Motivation and Strategy

The DWT layer can use a wide variety of well-known wavelets, e.g., Haar, Cohen–Daubechies–Feauveau (CDF) [34], and Deslauriers–Deubuc (DD) [35] wavelets. The frequency responses of these wavelets are shown in Fig. 4, where ω denotes the normalized angular frequency. However, the existing wavelets are not designed for audio source separation, which may limit the performance of the DWT layer. As described in Section II-C, the design of the wavelet is equivalent to that of the prediction and update operations. Since the optimal prediction and update operators may depend on the target sources and the network architecture, here we aim to train them simultaneously with the other DNN components.

The prediction and update operators can be implemented with the convolution layers with 1 filter, whose weights correspond to the time-reversed impulse responses of these operators. By making these weights trainable, we can obtain a trainable extension of the DWT layer, which we call the trainable DWT (TDWT) layer. Even during training, all the steps constituting the DWT layer are invertible and the TDWT layer is guaranteed to have the perfect reconstruction property. However, the frequency responses of the filters depend on the prediction and update operators and no anti-aliasing filters may exist when randomly determining the impulse responses of these operators. For this reason, the TDWT layer is not guaranteed to have the anti-aliasing filters during training, which may increase the dependence on training particularly when randomly initializing the weights. To reduce this dependence, in the subsequent section, we derive constraints of the weights to guarantee that the TDWT layer has the anti-aliasing filters during training.

B. Weight-Normalized Trainable DWT Layer

1) Constraints of Single Prediction and Update Operators: According to the definition of the low-pass filter, the condition of $H_I(z)$ to be a low-pass filter is given by

$$|H_I(1)| > 0, \quad H_I(-1) = 0. \tag{8}$$

Similarly, the condition of $G_1(z)$ to be a high-pass filter is given by

$$|G_I(-1)| > 0, \quad G_I(1) = 0. \tag{9}$$

Substituting Eqs. (5) and (6) into conditions (8) and (9), we can convert these conditions into those of the even- and odd-order filters, respectively:

$$|H_I^{(\text{even})}(1)| > 0, \qquad H_I^{(\text{even})}(1) = H_I^{(\text{odd})}(1), \tag{10}$$

$$|G_I^{(\text{even})}(1)| > 0, \qquad G_I^{(\text{even})}(1) = -G_I^{(\text{odd})}(1). \tag{11}$$

In this section, we derive constraints of the prediction and update operators to satisfy conditions (10) and (11) when using one prediction operator and one update operator, i.e., $I = 1$.

Let $p_{I,t}$ and $u_{I,t}$ denote the impulse responses of the Ith prediction and update operators, respectively, i.e.,

$$P_I(z) = \sum_t p_{I,t} z^{-t}, \tag{12}$$

$$U_I(z) = \sum_t u_{I,t} z^{-t}, \tag{13}$$

where t is the discrete time index. For the impulse responses, we can derive the following lemma:

Lemma 1: The filters of the lifting scheme with $I = 1$, $H_1(z)$ and $G_1(z)$, are respectively low- and high-pass filters if and only if

$$\sum_t p_{1,t} = 1, \quad \sum_t u_{1,t} = \frac{1}{2}. \tag{14}$$

Proof: When $I = 1$, Eq. (4) is written as

$$Q_1(z) = \begin{bmatrix} A(1 - P_1(z)U_1(z)) & AU_1(z) \\ P_1(z)/A & 1/A \end{bmatrix}. \tag{15}$$

Since the first matrix of the right-hand side of Eq. (7) equals $Q_I(z)$, we obtain

$$H_1^{(\text{even})}(z) = A(1 - P_1(z)U_1(z)), \tag{16}$$

$$H_1^{(\text{odd})}(z) = AU_1(z), \tag{17}$$

$$G_1^{(\text{even})}(z) = \frac{P_1(z)}{A}, \tag{18}$$

$$G_1^{(\text{odd})}(z) = \frac{1}{A}. \tag{19}$$

Comparing Eq. (16) with Eq. (19) with conditions (10) and (11) yields

$$P_1(1) = 1, \quad U_1(1) = \frac{1}{2}. \tag{20}$$

Here, if condition (20) is satisfied, the inequalities in conditions (10) and (11) are also satisfied since $A > 0$. By substituting Eqs. (12) and (13) at $z = 1$ into condition (20), we can obtain condition (14). ■

This derivation reveals that to satisfy conditions (10) and (11), we simply introduce a constraint that the weights of the convolution layers of the prediction are normalized according to Eq. (14).

2) Constraints of Multiple Prediction and Update Operators: Although the above derivation can be generalized in the case where $I > 1$, $Q_I(z)$ is much more complicated than Eq. (15), and a derivation similar to that in Section IV-B1 is impractical. In this section, we instead derive a sufficient condition in a simple form by using the property of the lifting scheme, which can systematically construct new wavelets from the existing ones by inserting the prediction and update steps.

We can derive the following lemma of the inserted prediction and update steps by starting with the lifting scheme whose $H_I(z)$ and $G_I(z)$ satisfy conditions (10) and (11):

Lemma 2: Consider the lifting scheme having the $I \geq 1$ pairs of the prediction and update steps, whose filters are denoted by $H_I(z)$ and $G_I(z)$, and extend it by inserting the $(I+1)$th prediction and update steps after the Ith prediction and update steps. If $H_I(z)$ and $G_I(z)$ satisfy conditions (10) and (11), the filters of the extended lifting scheme, $H_{I+1}(z)$ and $G_{I+1}(z)$, also satisfy conditions (10) and (11) if

$$\sum_t p_{I+1,t} = 0, \quad \sum_t u_{I+1,t} = 0. \tag{21}$$

Proof: Suppose that $H_I(z)$ and $G_I(z)$ satisfy conditions (10) and (11). By inserting the $(I+1)$th prediction and update steps after the Ith prediction and update steps, we can write $Q_{I+1}(z)$ in a recurrent form:

$$Q_{I+1}(z) = \begin{bmatrix} A & 0 \\ 0 & 1/A \end{bmatrix} \begin{bmatrix} 1 & U_{I+1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -P_{I+1}(z) & 1 \end{bmatrix} \times \begin{bmatrix} A & 0 \\ 0 & 1/A \end{bmatrix}^{-1} Q_I(z), \tag{22}$$

$$= \begin{bmatrix} 1 - P_{I+1}(z)U_{I+1}(z) & A^2 U_{I+1}(z) \\ -P_{I+1}(z)/A^2 & 1 \end{bmatrix} Q_I(z). \tag{23}$$

Here, the second last matrix of Eq. (22) represents the inverse of the scaling step since $Q_I(z)$ includes the operation of the scaling step. Since the first matrix of the right-hand side of Eq. (7) equals $Q_I(z)$, Eq. (23) can be seen as a recurrent form from $H_I^{(\text{even})}(z)$, $H_I^{(\text{odd})}(z)$, $G_I^{(\text{even})}(z)$, and $G_I^{(\text{odd})}(z)$ to the corresponding $(I+1)$th filters. Straightforward calculus yields

$$H_{I+1}^{(\text{even})}(z) = H_I^{(\text{even})}(z)(1 - P_{I+1}(z)U_{I+1}(z)) + A^2 G_I^{(\text{even})}(z)U_{I+1}(z), \tag{24}$$

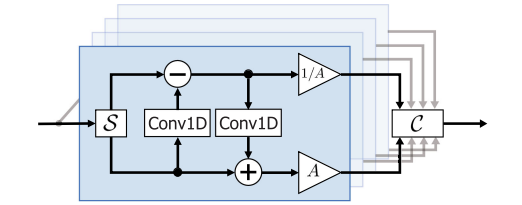
$$H_{I+1}^{(\text{odd})}(z) = H_I^{(\text{odd})}(z)(1 - P_{I+1}(z)U_{I+1}(z)) + A^2 G_I^{(\text{odd})}(z)U_{I+1}(z), \tag{25}$$

$$G_{I+1}^{(\text{even})}(z) = G_I^{(\text{even})}(z) - \frac{P_{I+1}(z)H_I^{(\text{even})}(z)}{A^2}, \tag{26}$$

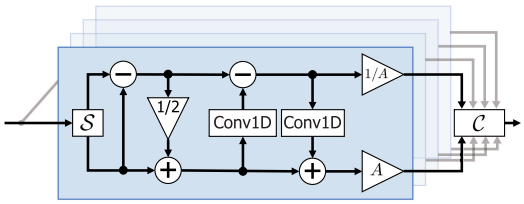
$$G_{I+1}^{(\text{odd})}(z) = G_I^{(\text{odd})}(z) - \frac{P_{I+1}(z)H_I^{(\text{odd})}(z)}{A^2}. \tag{27}$$

Invoking $H_I^{(\text{even})}(1) = H_I^{(\text{odd})}(1)$ and $G_I^{(\text{even})}(1) = -G_I^{(\text{odd})}(1)$, Eqs. (25) and (27) at $z = 1$ are respectively converted into

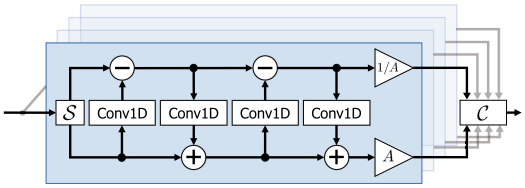
$$H_{I+1}^{(\text{odd})}(1) = H_I^{(\text{even})}(1)(1 - P_{I+1}(1)U_{I+1}(1))$$



(a) Type A.



(b) Type B.



(c) Type C.

Fig. 5. Architectures of TDWT and WN-TDWT layers used in experiments. The notations of the components are the same as those in Figs. 1 and 3.

$$- A^2 G_I^{(\text{even})}(1) U_{I+1}(1), \tag{28}$$

$$G_{I+1}^{(\text{odd})}(1) = -G_I^{(\text{even})}(1) - \frac{P_{I+1}(1) H_I^{(\text{even})}(1)}{A^2}. \tag{29}$$

By comparing Eqs. (24) and (26) at $z = 1$ with Eqs. (28), and (29), respectively, we can derive the conditions that $H_{I+1}(z)$ and $G_{I+1}(z)$ satisfy conditions (10) and (11) for any $H_I(z)$ and $G_I(z)$ satisfying conditions (10) and (11):

$$P_{I+1}(1) = 0, \quad U_{I+1}(1) = 0. \tag{30}$$

Substituting Eqs. (12) and (13) into condition (30) yields condition (21). ■

This lemma shows that we should simply normalize the weights of the convolution layers corresponding to the prediction and update operators as in Section IV-B1.

Consequently, guaranteeing that the DWT layer has the anti-aliasing filter and the perfect reconstruction property even during training can be achieved by normalizing the weights of the convolution layers according to Eq. (14) for the first prediction and update operators and according to Eq. (21) for the subsequent ones. We call the trainable DWT layer the weight-normalized trainable DWT (WN-TDWT) layer.

C. Architecture of Proposed Trainable Layers

The two proposed trainable layers do not force all the prediction and update operators to be trainable, and they allow these operators to be partially frozen. For example, we can use the architectures of the layers shown in Fig. 5. The simplest architecture has one pair of trainable prediction and update operators (see Fig. 5(a)). An example of the architecture with the frozen and trainable operators is shown in Fig. 5(b). It has the trainable operators after the prediction and update steps of Haar wavelets, which are frozen during training. This can be seen as modifying the Haar wavelets on the basis of the lifting scheme in a data-driven manner. The last example shown in Fig. 5(c) has two trainable prediction and update operators. We call the three architectures Types A, B, and C and will use them in Section V. As in the DWT layers, we can develop the corresponding US layers of the TDWT and WN-TDWT layers, which we call the inverse TDWT and inverse WN-TDWT layers, respectively.

The TDWT and WN-TDWT layers have the same computational complexity as the DWT layer except for the training cost of the prediction and update operators. The number of parameters of each operator equals only the filter length, and the increase in model size caused by the introduction of the trainable DWT layers is negligible compared with the number of parameters of the other DNN components.

V. Experimental Evaluation

A. Experimental Settings

1) Data Preparation and Training Procedure: To evaluate the importance of the anti-aliasing filters and the perfect reconstruction property in the feature domain, we conducted experiments using the MUSDB18 dataset [36], which consists of 100 training and 50 test tracks. The genres of the tracks vary widely, and for each track, four musical instruments (*vocals, bass, drums,* and *other*) were separately recorded. The audio signals were down-sampled to 22.05 kHz in the stereo format, i.e., $C^{(\text{in})} = 2$.

We used the same experimental settings as those in [11] except for data augmentation techniques. We augmented the training data by the standardization of the mixture audio signal to have zero mean and unit variance for each track, the random cropping of 6.68-s (147 443 samples) training audio segments, the random amplification within [0.75,1.25], the random channel swapping, and the random intertrack shuffling of instruments in 20% of the minibatch [37]. These augmentations except for the data standardization were performed on the fly during training. The batch size was 16 and the loss function was the mean squared error function. We used the Adam optimizer with a learning rate of 1.0×10^{-4} and decay rates of $\beta = 0.9$ and $\beta_2 = 0.999$. We defined one epoch by 2000 iterations and employed an early stopping technique to reduce the effect of overfitting similarly in [11]. We first continued to train each model until the validation loss was no longer improved for 20 successive epochs. Subsequently, we changed the batch size and learning rate to 32 and 1.0×10^{-5}, respectively, and fine-tuned the model with the same stopping criterion. We finally selected the trained model with the lowest validation loss. The hyperparameters were set as $A = \sqrt{2}$, $L = 12$, $C^{(\text{m})} = 312$, $C^{(\text{d})} = 24$, $f^{(\text{e})} = 15$, and $f^{(\text{d})} = 5$ for all models.

2) Evaluation Metric: We evaluated all models with a four-fold cross-validation scheme. For each data split and each instrument, we computed source-to-distortion ratios (SDRs) of the source estimates every one second for each track, took the

TABLE II
AVERAGES AND STANDARD ERRORS OF MEDIAN SDRS OBTAINED WITH MRDLA MODELS WITH HAAR, CDF, AND DD WAVELETS

Wavelet	Instrument			
	vocals	bass	drums	other
Haar	4.92 ± 0.13	4.52 ± 0.03	5.48 ± 0.03	3.09 ± 0.04
CDF(1,5)	5.07 ± 0.09	4.50 ± 0.18	5.52 ± 0.04	3.11 ± 0.05
CDF(2,2)	4.93 ± 0.11	4.22 ± 0.23	5.41 ± 0.06	3.07 ± 0.07
CDF(2,6)	4.88 ± 0.03	4.07 ± 0.13	5.29 ± 0.08	3.08 ± 0.04
DD4	4.99 ± 0.10	4.23 ± 0.16	5.44 ± 0.05	3.08 ± 0.12

median SDRs (trackwise SDRs), and obtained the median trackwise SDR over all the tracks. Similarly, the median source-to-interference ratios (SIRs) and source-to-artifacts ratios (SARs) were computed. Letting $\boldsymbol{s}_n \in \mathbb{R}^{T'}$ and $\hat{\boldsymbol{s}}_n \in \mathbb{R}^{T'}$ respectively denote the ground truth and estimated signals of source n, the SDR, SIR, and SAR for source n are given as follows [38]:

$$\mathrm{SDR}_n = 10 \log_{10} \frac{\|\boldsymbol{s}_{\mathrm{target},n}\|^2}{\|\boldsymbol{e}_{\mathrm{interf},n} + \boldsymbol{e}_{\mathrm{artif},n}\|^2}, \tag{31}$$

$$\mathrm{SIR}_n = 10 \log_{10} \frac{\|\boldsymbol{s}_{\mathrm{target},n}\|^2}{\|\boldsymbol{e}_{\mathrm{interf},n}\|^2}, \tag{32}$$

$$\mathrm{SAR}_n = 10 \log_{10} \frac{\|\boldsymbol{s}_{\mathrm{target},n} + \boldsymbol{e}_{\mathrm{interf},n}\|^2}{\|\boldsymbol{e}_{\mathrm{artif},n}\|^2}. \tag{33}$$

Here, $\boldsymbol{s}_{\mathrm{target},n}$, $\boldsymbol{s}_{\mathrm{interf},n}$ and $\boldsymbol{s}_{\mathrm{artif},n}$ are respectively defined by

$$\boldsymbol{s}_{\mathrm{target},n} = \mathcal{O}(\boldsymbol{s}_n)\hat{\boldsymbol{s}}_n, \tag{34}$$

$$\boldsymbol{s}_{\mathrm{interf},n} = \mathcal{O}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N)\hat{\boldsymbol{s}}_n - \mathcal{O}(\boldsymbol{s}_n)\hat{\boldsymbol{s}}_n, \tag{35}$$

$$\boldsymbol{s}_{\mathrm{artif},n} = \boldsymbol{s}_n - \mathcal{O}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N)\hat{\boldsymbol{s}}_n, \tag{36}$$

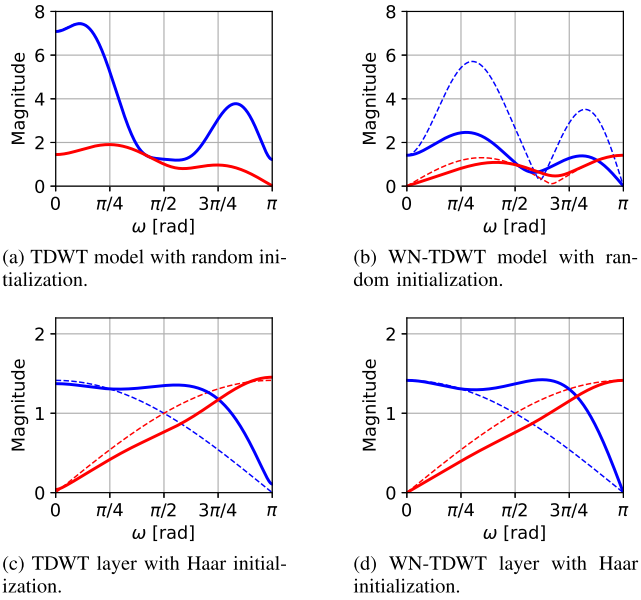
where $\mathcal{O}(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N)$ denotes an orthogonal projector onto the subspace spanned by $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N$. These metrics were computed after optimally matching the ground truth and estimated signals by a linear time-invariant filter, which compensates for the linear mismatches between the ground truth and the estimated signals. This procedure was used in the signal separation evaluation campaign 2018 (see [1], [38] for the details). We used the averages and standard errors of the median SDRs, SIRs, and SARs over the four data split as evaluation metrics.

B. Effect of Wavelet Basis Functions

We first evaluated the effects of wavelet basis functions of the DWT layer for MRDLA. We set $C^{(\mathrm{e})} = 18$ and used the Haar, CDF(1,5), CDF(2,2), CDF(2,6), and fourth-order DD (DD4) wavelets as the wavelet basis functions, whose frequency responses are shown in Fig. 4. Table II shows the separation performance characteristics of the MRDLA models. These models gave similar separation performance characteristics, showing the robustness of MRDLA against the wavelet variations. In the following experiments, we used the Haar wavelet for the DWT layer with the predetermined weights.

C. Trainable Extensions of DWT Layers

We evaluated the effect of the trainable extensions of the DWT layer. We call the MRDLA model having the TDWT (WN-TDWT) layer the TDWT (WN-TDWT) model. We adopted the



(a) TDWT model with random initialization.

(b) WN-TDWT model with random initialization.

(c) TDWT layer with Haar initialization.

(d) WN-TDWT layer with Haar initialization.

Fig. 6. Frequency responses of $H_1(z)$ (blue) and $G_1(z)$ (red) of Type-A TDWT and WN-TDWT layers with random and Haar initializations. The solid and dashed curves represent the initial and trained frequency responses, respectively.

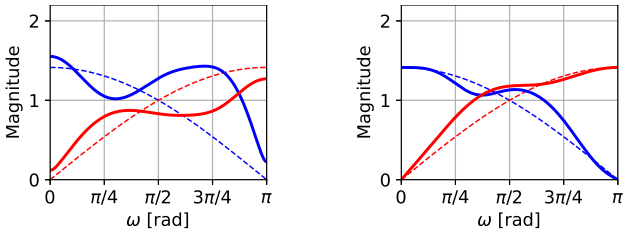
architectures of the trainable layers named Types A, B, and C, as shown in Fig. 5, and set the filter size of the convolution layers to 3 for the trainable prediction and update operators. The weights of these convolution layers were shared between all the trainable DWT and inverse DWT layers of each network.

We first compared two weight initialization methods for the Type-A TDWT and WN-TDWT layers: the random initialization and the initialization by the prediction and update operators of the Haar wavelets (Haar initialization). Table III shows the averages and standard errors of the median SDRs for the TDWT and WN-TDWT models, where we set $C^{(\mathrm{e})} = 18$. For the TDWT model, the weight initialization methods heavily affected the numerical stability and separation performance. With the random initialization, we often encountered the sudden rise of training and validation losses, although they did not diverge. The use of the Haar initialization did not cause such a rise of the losses, and it greatly increased the separation performance, which shows that the TDWT layer is sensitive to the initial values of the prediction and update operators. On the other hand, the WN-TDWT models provided a consistent performance regardless of the initializations, clearly showing that guaranteeing the existence of the anti-aliasing filters during training reduces the dependence on the weight initialization. As shown in Fig. 6, with the random initialization, the frequency responses of the TDWT layer did not change before and after training, whereas those of the WN-TDWT layer changed, particularly in their magnitudes. Interestingly, the frequency responses of the trained Type-A TDWT and WN-TDWT layers with the Haar initialization were similar but different from those of the Haar wavelets.

We examined the effect of the architectures of the proposed trainable layers. For Type B, the weights of the convolution layers were initialized by zeros. For Type C, the first two convolution layers were initialized by the Haar wavelet and

TABLE III
SEPARATION PERFORMANCE CHARACTERISTICS OF MRDLA MODELS WITH PROPOSED TRAINABLE DWT LAYERS

Initialization	Architecture	DS layer	Instrument			
			vocals	bass	drums	other
Random	Type A	TDWT	2.68 ± 0.03	3.07 ± 0.13	3.73 ± 0.05	1.83 ± 0.07
		WN-TDWT	$\mathbf{4.62 \pm 0.12}$	$\mathbf{4.17 \pm 0.04}$	$\mathbf{5.50 \pm 0.05}$	$\mathbf{2.94 \pm 0.06}$
Haar	Type A	TDWT	4.82 ± 0.14	$\mathbf{4.47 \pm 0.15}$	5.50 ± 0.04	3.00 ± 0.06
		WN-TDWT	4.87 ± 0.11	4.44 ± 0.14	5.49 ± 0.07	3.08 ± 0.08
	Type B	TDWT	4.72 ± 0.11	4.38 ± 0.25	5.37 ± 0.13	3.07 ± 0.11
		WN-TDWT	$\mathbf{4.99 \pm 0.06}$	4.46 ± 0.13	$\mathbf{5.59 \pm 0.07}$	$\mathbf{3.17 \pm 0.04}$
	Type C	TDWT	4.82 ± 0.16	4.30 ± 0.14	5.44 ± 0.07	3.05 ± 0.06
		WN-TDWT	4.90 ± 0.08	4.36 ± 0.06	5.47 ± 0.05	3.09 ± 0.07



(a) Type-C TDWT layer. (b) Type-B WN-TDWT layer.

Fig. 7. Frequency responses of $H_2(z)$ and $G_2(z)$ of Type-C TDWT and Type-B WN-TDWT layers with Haar initialization. The lines and colors similarly represent the responses in Fig. 6.

the other weights were initialized by zeros. These initializations correspond to the Haar initialization. As summarized in Table III, the architectures of the TDWT and WN-TDWT layers did not greatly affect SDRs. Since the TDWT layer is not guaranteed to have the low- and high-pass filters, the $H_I(z)$ ($G_I(z)$) of the trained TDWT layer, particularly the layer with the Type C architecture, was not strictly zero at $z = -1$ ($z = 1$), as shown in Fig. 7(a). Nevertheless, the trained TDWT layers showed nearly low- and high-pass frequency responses, which may be one of the reasons why the TDWT models showed a similar performance to the WN-TDWT models. The trained WN-TDWT layers showed the low- and high-pass filters consistently with the theoretical results given in Section IV-B. The Type-B WN-TDWT model provided the highest SDRs for *vocals*, *drums*, and *other*, and the trained WN-TDWT layer showed the slightly different frequency responses from the Haar wavelet, as shown in Fig. 7(b).

D. Comparison of MRDLA and Wave-U-Net

To evaluate the advantage of MRDLA, we compared the proposed MRDLAs having the DWT layers and the WN-TDWT layers of Type B with Wave-U-Net. We call the former MRDLA *Proposed* and the latter one *Proposed w/ WN-TDWT*. The DWT and WN-TDWT layers double the channel size of the feature, whereas the decimation layer leaves it unchanged, which makes it not easy to exactly match the model sizes of MRDLA and Wave-U-Net. For a fair comparison, we compared these methods at various $C^{(e)}$ values, as shown in Table IV, where the Average Pooling and Squeezing models will be used for the comparison in Section V-E. Note that we removed the tangent hyperbolic function located at the end of Wave-U-Net since the data standardization may increase the values of the training audio signals outside the range of $[-1, 1]$.

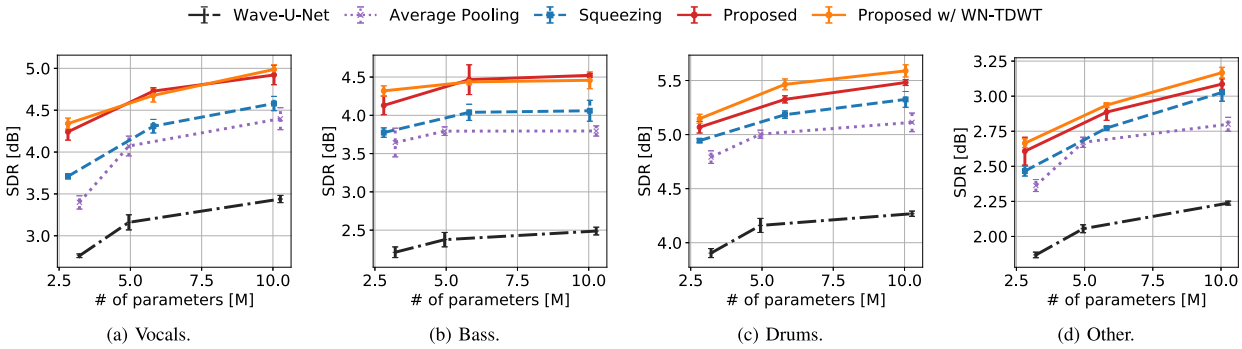
TABLE IV
FEATURES OF PROPOSED AND CONVENTIONAL MODELS. "AAF" AND "PRP" ARE ABBREVIATIONS OF ANTI-ALIASING FILTERS AND THE PERFECT RECONSTRUCTION PROPERTY, RESPECTIVELY

Method	$C^{(e)}$	DS layer	
		Have AAF	Have PRP
Wave-U-Net [11]	6, 12, 24	No	No
Proposed	6, 12, 18	**Yes**	**Yes**
Proposed w/ WN-TDWT	6, 12, 18	**Yes**	**Yes**
Average Pooling	6, 12, 24	**Yes**	No
Squeezing	6, 12, 18	No	**Yes**

Figs. 8, 9, and 10 respectively show the average and standard errors of the median SDRs, SIRs, and SARs. For all instruments, the proposed methods gave a comparable performance with a smaller model size than Wave-U-Net and consistently provided a higher performance with a similar model size in terms of SDR and SIR. Compared with Wave-U-Net, the SARs of the proposed methods were comparable for *vocals* and *bass* and lower for *drums* and *other*. However, the improvement of the SIRs of the proposed methods is sufficiently large to counteract the SAR degradation and enhance the overall separation quality, which leads to an SDR improvement of approximately 1.5 dB. To examine its perceptual effect, we will show a subjective evaluation in Section V-F. One may think that it is unnatural that the SAR degradation occurred despite the fact that the proposed DWT layers have the anti-aliasing filters. However, once the nonlinear layer is applied to the down-sampled feature, the feature processed by the nonlinear layer may include artifacts. The down-sampled feature is processed by the nonlinear layers at least once, and thus it is not guaranteed that the output of the MRDLA model does not include artifacts in the waveform domain even though the DS layer has the anti-aliasing filter in the feature domain. For this reason, the DWT layers do not necessarily directly affect the SARs of the source estimates.

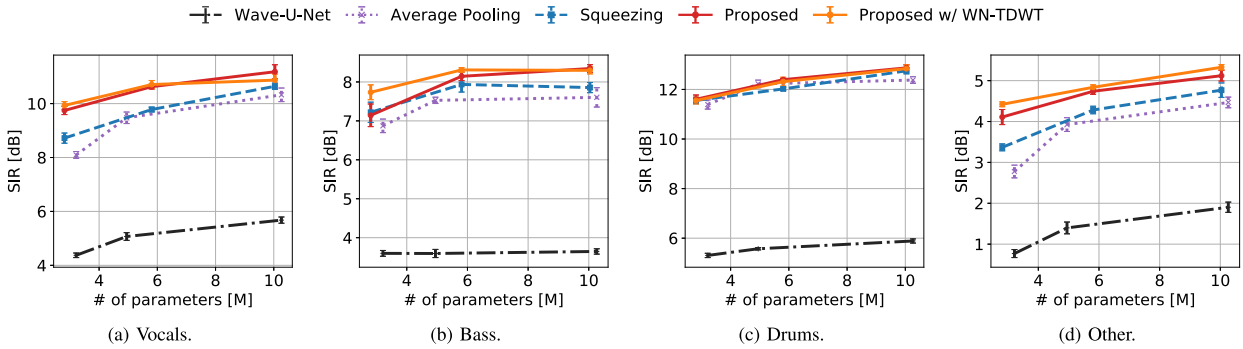
Proposed w/ WN-TDWT gave comparable SDRs for *vocals* and *bass* and slightly higher SDRs than *Proposed* for *drums* and *other*. This observation shows the efficacy of the trainable extension of the DWT layer and that taking into account the anti-aliasing filters and the perfect reconstruction property is more important than the wavelet basis functions.

To examine the statistical significance of SDRs between *Proposed w/ WN-TDWT* and Wave-U-Net, we performed the Wilcoxon signed-rank test of trackwise SDRs for each instrument and each data split. As a result, since the p-values of all instruments and data splits were far below 1.0×10^{-3}, we confirm that MRDLA significantly improves the separation performance compared with Wave-U-Net.



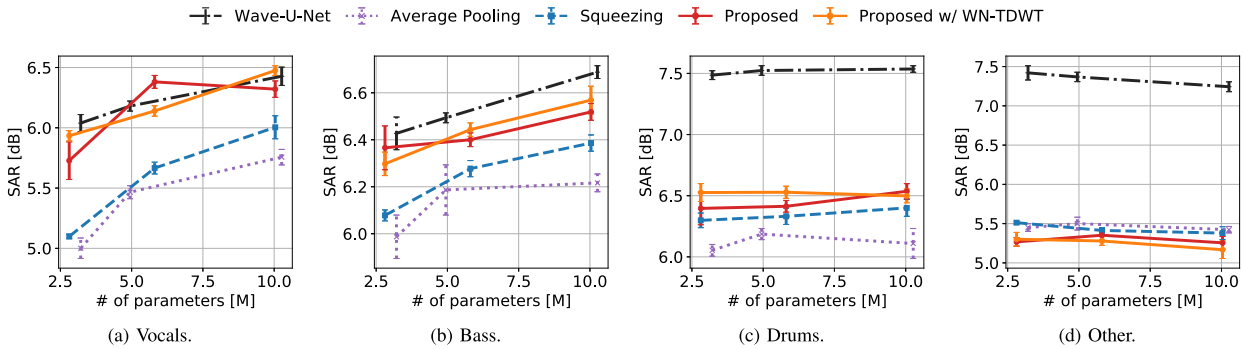
(a) Vocals. (b) Bass. (c) Drums. (d) Other.

Fig. 8. Averages and standard errors of median SDRs of proposed and conventional models.



(a) Vocals. (b) Bass. (c) Drums. (d) Other.

Fig. 9. Averages and standard errors of median SIRs of proposed and conventional models.



(a) Vocals. (b) Bass. (c) Drums. (d) Other.

Fig. 10. Averages and standard errors of median SARs of proposed and conventional models.

E. Effects of Anti-Aliasing Filters and Perfect Reconstruction Property in Feature Domain

We separately evaluated the effects of the anti-aliasing filters and the perfect reconstruction property by comparing the following two variants of MRDLA, *Average Pooling* and *Squeezing*. *Average Pooling* uses the average pooling and linear US layers instead of the DWT and inverse DWT layers in *Proposed*, and *Squeezing* has the same architecture as *Proposed* while using the squeezing operation and its inverse as the DS and US layers. As summarized in Table IV, these models lack either one of the anti-aliasing filters and the perfect reconstruction property.

Before the performance comparison, we examined whether the feature-domain aliasing occurs. Fig. 11 shows the spectra of the input feature of the first DS layer of the trained *Squeezing* model. The spectra had high energies in the frequency band

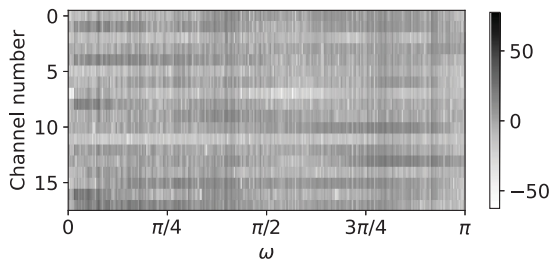


Fig. 11. Spectra [dB] of input feature of first DS layer of trained *Squeezing* model.

over $\omega > \pi/2$, which shows that the down-sampled features were aliased unless the following DS layer had an anti-aliasing filter. We observed that the energies of the input features were

distributed over $\omega > \pi/2$ in the other trained models. This observation shows that the feature-domain aliasing can frequently appear in the DNNs of time-domain audio source separation. Furthermore, we found that some filters of the trained convolution layers of the DS blocks had high energies over $\omega > \pi/2$. This observation shows that the trained convolution layers did not play a role of anti-aliasing filters, which is consistent with the observation in audio recognition tasks [18].

The separation performance characteristics of the two models are shown in Figs. 8, 9, and 10. The performance characteristics of *Average Pooling* and *Squeezing* did not reach those of *Proposed* at most of the model sizes, showing the advantage of simultaneously featuring the anti-aliasing filters and the perfect reconstruction property. *Squeezing* achieved the second best performance at most of the model sizes and for most instruments, which shows that the perfect reconstruction property is more important for the separation performance than the anti-aliasing filters. The performance gap between *Proposed* and *Squeezing* was greater for *vocals* and *bass* than for *drums*, suggesting that the feature-domain aliasing affects the pitched sounds more greatly than the percussive sounds.

F. Comparison With State-of-The-Art Methods

1) Separation Performance: To evaluate the effectiveness of MRDLA, we finally compared the best MRDLA model, *Proposed w/ WN-TDWT*, with two conventional time-domain audio source separation methods in addition to Wave-U-Net: a noncausal WaveNet [15], which we call WaveNet, and Conv-TasNet [14]. Since the model sizes of the original Conv-TasNet and WaveNet were smaller than those of *Proposed w/ WN-TDWT*, for a fair comparison, we implemented Conv-TasNet and WaveNet variants, namely, Conv-TasNet+ and WaveNet+, respectively, by increasing their model sizes up to those of *Proposed w/ WN-TDWT*. All the models were trained with the same dataset and data augmentation as in Section V-A.

Proposed w/ WN-TDWT: We used the MRDLA model with the Type-B WN-TDWT layer and $C^{(\mathrm{e})} = 36$, which achieved the best separation performance in Section V-D.

Wave-U-Net: We used the Wave-U-Net model with $C^{(\mathrm{e})} = 24$, which provided the best performance in the Wave-U-Net models in Section V-D.

WaveNet: Since WaveNet was originally designed for monaural inputs, we respectively doubled the input and output channel sizes of the first and last convolutions so that they can deal with stereo signals. We used the same settings for the early stopping technique as in [15].

WaveNet+: To increase the model size of the original WaveNet to that of *Proposed w/ WN-TDWT*, we changed the number of channels of the residual blocks, which is denoted by k in [15], from 64 to 164.

Conv-TasNet: For the same reason as that for WaveNet, we respectively doubled the input and output channel sizes of the first and last convolutions so that it can deal with stereo input and outputs. To reduce the overfitting, we employed the early stopping technique with the stopping criterion that the validation losses did not decrease for 20 successive epochs. The loss function of Conv-TasNet is scale-invariant and we experimentally found a large gap in scale between the source estimates and the ground truths. Since this mismatch greatly decreased SDRs, as a postprocessing, we scaled the source estimates by instrument-wise factors to minimize the mean squared error between the input mixture and the sum of all instrument estimates for each track.

Conv-TasNet+: To increase the model size of the original Conv-TasNet to that of *Proposed w/ WN-TDWT*, we doubled the number of channels of the bottleneck layers of the mask estimator, which is denoted as B in [14].

The other experimental settings were the same as those described in the literature of each method.

Table V shows the separation performance characteristics of all the methods. For all instruments, *Proposed w/ WN-TDWT* achieved the highest SDRs and SIRs, clearly showing the effectiveness of MRDLA. As described in Section V-D, although the SARs of *Proposed w/ WN-TDWT* were comparable to but lower than those of Wave-U-Net, the large improvement of *Proposed w/ WN-TDWT* in SIR greatly enhances the overall separation quality by around 1.5 dB in SDR. We examined the statistical significance of the difference in separation performance between MRDLA and Conv-TasNet, which provided the highest average SDRs over all instruments in the conventional methods, by performing the Wilcoxon signed-rank test of trackwise SDRs. We confirmed that the p-values were far below 1.0×10^{-3} for each instrument and each data split, which clearly shows that MRDLA significantly outperforms the conventional methods in SDR.

2) Perceptual Quality: Furthermore, to evaluate the perceptual quality of the separation results, we conducted a preference test by comparing MRDLA with Conv-TasNet, which achieved the highest average SDRs over all instruments in the conventional methods. Ten tracks were randomly chosen from 40 to 50 s of the 50 test tracks of the MUSDB18 dataset. For each track, we randomly chose one of the data splits and prepared the separation results of the chosen data split. In addition to the separated audio signals, we prepared the so-called minus-one audio signals of each method, which were computed by subtracting the separated audio signals from the mixtures. We consider that the minus-one audio signals are helpful for the participants to check the leakage of the target sources to the residuals. The mixtures and ground truth signals were also provided as references, and the participants were blind to the method names. For each track and each instrument, 12 participants were asked to listen to the separation results and choose the one with a higher overall separation quality, taking into account (i) the sound quality of the target source, (ii) the distortions of the target source, (iii) the naturalness of the interferences in the separated signals, and (iv) the leakage of the target sources to the residuals. They could listen to the audio signals as many times as they wanted. Similarly to the above, we also conducted a preference test to compare MRDLA with Wave-U-Net, which gave the highest SARs for *bass* and *drums*. Note that the same tracks and data splits were used in the two preference tests.

TABLE V
SEPARATION PERFORMANCE CHARACTERISTICS OF MRDLA AND CONVENTIONAL TIME-DOMAIN AUDIO SOURCE SEPARATION METHODS

Metric	Method	# of parameters [M]	Instrument			
			vocals	bass	drums	other
SDR	Wave-U-Net [11]	10.26	3.44 ± 0.05	2.49 ± 0.06	4.27 ± 0.03	2.24 ± 0.02
	WaveNet [15]	3.16	2.99 ± 0.15	3.01 ± 0.11	3.52 ± 0.16	1.65 ± 0.15
	WaveNet+	10.35	1.94 ± 0.35	2.98 ± 0.17	3.23 ± 0.06	0.79 ± 0.34
	Conv-TasNet [14]	5.26	4.02 ± 0.24	3.59 ± 0.15	4.52 ± 0.14	2.29 ± 0.08
	Conv-TasNet+	10.32	2.73 ± 0.89	4.14 ± 0.22	4.66 ± 0.22	1.15 ± 0.70
	Proposed w/ WN-TDWT	10.04	$\mathbf{4.99 \pm 0.06}$	$\mathbf{4.46 \pm 0.13}$	$\mathbf{5.59 \pm 0.07}$	$\mathbf{3.17 \pm 0.04}$
SIR	Wave-U-Net [11]	10.26	5.68 ± 0.13	3.65 ± 0.08	5.89 ± 0.10	1.90 ± 0.15
	WaveNet [15]	3.16	8.64 ± 0.18	5.48 ± 0.26	10.50 ± 0.26	1.51 ± 0.49
	WaveNet+	10.35	8.27 ± 0.93	5.63 ± 0.49	8.82 ± 0.71	0.09 ± 0.43
	Conv-TasNet [14]	5.26	9.24 ± 0.58	6.52 ± 0.75	11.13 ± 0.78	3.76 ± 0.67
	Conv-TasNet+	10.32	9.23 ± 0.26	8.08 ± 0.05	$\mathbf{13.26 \pm 0.39}$	0.68 ± 1.24
	Proposed w/ WN-TDWT	10.04	$\mathbf{10.87 \pm 0.26}$	$\mathbf{8.30 \pm 0.10}$	12.84 ± 0.17	$\mathbf{5.33 \pm 0.07}$
SAR	Wave-U-Net [11]	10.26	6.43 ± 0.09	$\mathbf{6.69 \pm 0.03}$	$\mathbf{7.54 \pm 0.03}$	7.24 ± 0.07
	WaveNet [15]	3.16	4.14 ± 0.12	4.76 ± 0.13	4.79 ± 0.07	5.93 ± 0.70
	WaveNet+	10.35	2.48 ± 0.79	4.88 ± 0.32	5.24 ± 0.21	7.34 ± 0.87
	Conv-TasNet [14]	5.26	5.56 ± 0.14	6.05 ± 0.20	5.57 ± 0.05	4.62 ± 0.24
	Conv-TasNet+	10.32	5.31 ± 0.14	5.90 ± 0.12	5.83 ± 0.06	$\mathbf{7.60 \pm 1.40}$
	Proposed w/ WN-TDWT	10.04	$\mathbf{6.48 \pm 0.05}$	6.57 ± 0.07	6.50 ± 0.07	5.17 ± 0.13

TABLE VI
RESULTS OF PREFERENCE TEST OF MRDLA AND CONV-TASNET

Instrument	Preference score [%]		p-value
	Conv-TasNet	MRDLA	
vocals	13.33	**86.67**	$< 10^{-15}$
bass	13.33	**86.67**	$< 10^{-15}$
drums	10.83	**89.17**	$< 10^{-17}$
other	22.50	**77.50**	$< 10^{-8}$

TABLE VII
RESULTS OF PREFERENCE TEST OF MRDLA AND WAVE-U-NET

Instrument	Preference score [%]		p-value
	Wave-U-Net	MRDLA	
vocals	29.17	**70.83**	$< 10^{-5}$
bass	6.67	**93.33**	$< 10^{-20}$
drums	19.17	**80.83**	$< 10^{-10}$
other	26.67	**73.33**	$< 10^{-6}$

Tables VI and VII show the results of the preference test of MRDLA and Conv-TasNet, and those of MRDLA and Wave-U-Net, respectively. Here, the p-values were computed by Pearson's chi-squared test. The results apparently show that MRDLA significantly outperforms Conv-TasNet and Wave-U-Net in perceptual separation quality for all instruments. When listening to the separation results, we observed that the separated audio signals of Conv-TasNet had high-frequency noises, particularly for *bass* and *drums*, and the leakage of the target sources to the residuals often occurred. Although the separated audio signals of Wave-U-Net had less noises, they included a large amount of interference sounds at audible levels for all instruments, which apparently degrades the separation quality. On the other hand, we observed that the separated audio signals of MRDLA include the target sources from low to high frequency bands and sound more clearly than those of the other methods. Note that the separated audio signals of WaveNet sound choppy and had distortions, particularly for *bass* and *drums*. Some audio examples of MRDLA and the conventional methods are available at http://tomohikonakamura.github.io/Tomohiko-Nakamura/demo/MRDLA/index.html.

VI. CONCLUSION

We presented a novel time-domain audio source separation method, MRDLA, based on multiresolution analysis by developing the DWT and inverse DWT layers. The basis for developing the proposed layers was from our observation that the successive DS architecture of Wave-U-Net resembles that of multiresolution analysis. From the signal processing viewpoint, we found that the decimation layer causes the feature-domain aliasing and discards parts of input features, which may degrade the separation performance. To simultaneously overcome these two problems, we designed the proposed layers by using a DWT for DS because a DWT has the anti-aliasing filter and the perfect reconstruction property.

We further presented the TDWT and WN-TDWT layers by extending the DWT layer so that its prediction and update operators can be trained simultaneously with the other DNN components. We derive the constraints of the weights of the convolution layers corresponding to the prediction and update operators to guarantee that the WN-DWT layer has the anti-aliasing filter in addition to the perfect reconstruction property even during training. Through systematic experiments on music source separation, we showed the efficacy of the MRDLA models and the importance of taking into account the anti-aliasing filters and the perfect reconstruction property in the feature domain. We further showed that the use of the trainable DWT layers can slightly improve the separation performance. The experimental analysis revealed that maintaining the anti-aliasing filters during the training can reduce the dependence on the training and the initial values of the prediction and update operators. Through the objective and subjective experiments, we confirmed that MRDLA significantly outperformed the conventional time-domain audio separation methods in SDR and perceptual quality.

REFERENCES

[1] F. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, Jul. 2018, pp. 293–305.

[2] J. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 31–35.

[3] A. Jansson, E. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-Net convolutional networks," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Oct. 2017.

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

[5] N. Takahashi, N. Goswami, and Y. Mitsufuji, "Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation," in *Proc. Int. Workshop Acoust. Signal Enhancement*, Sep. 2018, pp. 106–110.

[6] J. Le Roux, N. Ono, and S. Sagayama, "Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction," in *Proc. Workshop Statist. Perceptual Audition*, Sep. 2008, pp. 23–28.

[7] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, Apr. 1984.

[8] Z. Průša, P. Balazs, and P. L. Søndergaard, "A noniterative method for reconstruction of phase from STFT magnitude," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 5, pp. 1154–1164, Mar. 2017.

[9] S. Takamichi, Y. Saito, N. Takamune, D. Kitamura, and H. Saruwatari, "Phase reconstruction from amplitude spectrograms based on directional-statistics deep neural networks," *Signal Process.*, vol. 169, Apr. 2020, Art. no. 107368.

[10] J. Le Roux, G. Wichern, S. Watanabe, A. Sarroff, and J. R. Hershey, "Phasebook and friends: Leveraging discrete representations for source separation," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 2, pp. 370–382, May 2019.

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

[12] S. Venkataramani, J. Casebeer, and P. Smaragdis, "End-to-end source separation with adaptive front-ends," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Oct. 2018, pp. 684–688.

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

[14] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, May 2019.

[15] F. Lluís, J. Pons, and X. Serra, "End-to-end music source separation: Is it possible in the waveform domain?," in *Proc. INTERSPEECH*, Sep. 2019, pp. 4619–4623.

[16] I. Kavalerov *et al.*, "Universal sound separation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2019, pp. 175–179.

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

[18] Y. Gong and C. Poellabauer, "Impact of aliasing on deep CNN-based end-to-end acoustic models," in *Proc. INTERSPEECH*, Sep. 2018, pp. 2698–2702.

[19] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 818–833.

[20] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, "Convolutional Kernel networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 2627–2635.

[21] R. Zhang, "Making convolutional networks shift-invariant again," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, Jul. 2019, pp. 7324–7334.

[22] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[23] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, Apr. 1996.

[24] T. Nakamura and H. Saruwatari, "Time-domain audio source separation based on wave-U-Net combined with discrete wavelet transform," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2020, pp. 386–390.

[25] S. Kozuka, T. Nakamura, and H. Saruwatari, "Investigation on wavelet basis function of DNN-based time domain audio source separation inspired by multiresolution analysis," in *Proc. Int. Congr. Expo. Noise Control Eng.*, Aug. 2020, pp. 4013–4022.

[26] T. Williams and R. Li, "Wavelet pooling for convolutional neural networks," in *Proc. Int. Conf. Learn. Representations*, May 2018.

[27] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. Int. Conf. Learn. Representations*, Apr. 2017.

[28] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.

[29] J.-H. Jacobsen, A. Smeulders, and E. Oyallon, "i-RevNet: Deep invertible networks," in *Proc. Int. Conf. Learn. Representations*, Apr. 2018.

[30] D. Takeuchi, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Invertible dnn-based nonlinear time-frequency transform for speech enhancement," in *IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6644–6648.

[31] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, Mar. 1998.

[32] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, May 1998.

[33] A. Abbate, C. DeCusatis, and P. K. Das, *Wavelets and Subbands: Fundamentals and Applications. Birkhäuser*, Boston, MA, 2002.

[34] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, no. 5, pp. 485–560, Jun. 1992.

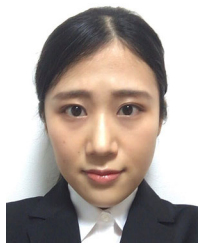
[35] G. Deslauriers and S. Dubuc, "Symmetric iterative interpolation processes," in *Constructive Approx.*, Dec. 1989, pp. 49–68.

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

[37] N. Takahashi, M. Gygli, B. Pfister, and L. V. Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *Proc. INTERSPEECH*, 2016, pp. 2982–2986.

[38] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.

Tomohiko Nakamura received the B.E., M.S., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 2011, 2013, and 2016, respectively. He was with SECOM Intelligent Systems Laboratory, Tokyo, Japan, in 2016, and has been a Project Research Associate with the University of Tokyo, Tokyo, Japan, since 2019. His research interests include audio signal processing, music signal and information processing, and machine learning. He is a member of the IEEE Signal Processing Society (SPS), the Acoustical Society of Japan (ASJ), and the Information Processing Society of Japan (IPSJ). He was the recipient of the IPSJ Yamashita SIG Research Award in 2015, the SICE Best Paper Award (Takeda Award) in 2016, and the Graduate School Dean's Award from the University of Tokyo in 2016.



Shihori Kozuka received the B.E. degree in engineering from the University of Tokyo, Tokyo, Japan, in 2020. She is currently working toward the M.S. degree in information physics and computing with the University of Tokyo, Tokyo, Japan. Her research interests include audio signal processing, source separation, non-linear acoustics, and noise reduction.

Hiroshi Saruwatari received the B.E., M.E., and Ph.D. degrees from Nagoya University, Japan, in 1991, 1993, and 2000, respectively. He joined SECOM IS Laboratory, Japan, in 1993, and Nara Institute of Science and Technology, Japan, in 2000. From 2014, he is currently a Professor of The University of Tokyo, Japan. His research interests include statistical speech signal processing, blind source separation (BSS), audio enhancement, and robot audition. He has successfully achieved his carrier, especially on BSS researches, and put his research into the world's first commercially available Independent-Component-Analysis-based BSS microphone in 2007. He was the recipient of the Paper Awards from IEICE in 2001 and 2006, from TAF in 2004, 2009 and 2012, from IEEE-IROS2005 in 2006, and from APSIPA in 2013 and 2018. He received DOCOMO Mobile Science Award in 2011, Ichimura Award in 2013, The Commendation for Science and Technology by the Minister of Education in 2015, Achievement Award from IEICE in 2017, and Hattori-Hoko Award in 2018. He won the first prize in IEEE MLSP2007 BSS Competition. He has been professionally involved in various volunteer works for IEEE, EURASIP, IEICE, and ASJ, including chair posts of international conferences and Associate Editor of journals.