symmetry

MDPI

Article

Monaural Singing Voice and Accompaniment Separation Based on Gated Nested U-Net Architecture

Haibo Geng [1,2], Ying Hu [1,2,*] and Hao Huang [1,3]

1 School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; ghb561323@stu.xju.edu.cn (H.G.); huanghao@xju.edu.cn (H.H.)
2 Key Laboratory of Signal Detection and Processing in Xinjiang Uygur Autonomous Region, Urumqi 830046, China
3 Key Laboratory of Multilingual Information Technology in Xinjiang Uygur Autonomous Region, Urumqi 830046, China
* Correspondence: huying@xju.edu.cn; Tel.: +86-186-0991-4062

Received: 31 May 2020; Accepted: 23 June 2020; Published: 24 June 2020

Abstract: This paper proposes a separation model adopting gated nested U-Net (GNU-Net) architecture, which is essentially a deeply supervised symmetric encoder–decoder network that can generate full-resolution feature maps. Through a series of nested skip pathways, it can reduce the semantic gap between the feature maps of encoder and decoder subnetworks. In the GNU-Net architecture, only the backbone not including nested part is applied with gated linear units (GLUs) instead of conventional convolutional networks. The outputs of GNU-Net are further fed into a time-frequency (T-F) mask layer to generate two masks of singing voice and accompaniment. Then, those two estimated masks along with the magnitude and phase spectra of mixture can be transformed into time-domain signals. We explored two types of T-F mask layer, discriminative training network and difference mask layer. The experiment results show the latter to be better. We evaluated our proposed model by comparing with three models, and also with ideal T-F masks. The results demonstrate that our proposed model outperforms compared models, and it's performance comes near to ideal ratio mask (IRM). More importantly, our proposed model can output separated singing voice and accompaniment simultaneously, while the three compared models can only separate one source with trained model.

Keywords: singing voice separation; nested U-Net; gated linear units; CNN; monaural source separation

1. Introduction

Singing voice separation attempts to isolate singing voice (also called vocal line) from a song. In recent years, this problem has attracted increasing attention with the demand for singer identification [1–3], automatic lyrics recognition [4,5] and alignment [6], singing pitch estimation [7], singing style visualization [8], and so on. Meanwhile, isolating pure accompaniment from a song also has great applications such as leading instrument detection [9] and drum source separation [10]. Although these tasks seem effortless to humans, it turns out to be very difficult for machines, especially when the singing voice is accompanied by musical instruments. However, such a requirement can be satisfied if successful separations of singing voice and accompaniment are used as preprocessing.

A popular song often has two major acoustic components that are singing voice and background accompaniment. Due to the harmony of a popular song, the singing voice and accompaniment are strongly correlated in both time and frequency [11], thus separating singing voice from a song

in single channel is a challenging task. Several approaches have been proposed for singing voice separation. Po-sen Huang et al. [12] proposed using robust principal component analysis for singing voice separation from music accompaniment. Hu and Liu proposed a system based on Non-negative Matrix Factorization (NMF) to separate singing voice from monaural music for singer identification [2]. It indeed helps to improve the performance of singer identification. However, the performance of singing voice separation still need to be boosted especially when the energy of accompaniment in a recording is larger than that of the singing voice.

With the development of deep learning, most recent methods based on deep learning show better performance [11,13]. Po-Sen Huang et al. explored using deep recurrent neural networks (RNN) for singing voice separation from monaural recordings [14]. Moreover, they proposed the joint optimization of mask functions and deep RNN, exploring a discriminative training criterion for neural networks to further enhance the separation performance [15]. Fan et al. proposed a monaural singing voice separation model using generative adversarial network (GAN) with a T-F masking function [16]. Generator G inputs a mixture spectra and generates realistic singing voice and accompaniment spectra, while discriminator D distinguishes the clean spectra from those generated spectra, which can be transformed into time-domain signals using the inverse short-time Fourier transform (ISTFT) with phase information. He et al. [17] also used the adversarial mechanism to improve the separation effect of monaural singing voice separation networks. The GAN's discriminator was introduced to measure the correlation between the latent variables of the vocals and music generated by the variational autoencoder probability encoder. Stoller et al. proposed a semisupervised approach, also using GAN on multitrack data for singing voice extraction [18].

The above supervised source separation approaches are all conducted in the time-frequency (T-F) domain [13–19]. These approaches reconstruct the target source signal in the time domain from the frequency domain using the phase of mixture by inverse short time Fourier transform (ISTFT). This paper also focuses on being conducted in the T-F domain.

Gating mechanisms were first proposed by Dauphin et al. for language modeling [20] in 2017. Since then, gating mechanism—also termed as gated linear units (GLUs)—has been broadly applied to the speech process field. Tan and Wang [21] extended the convolutional recurrent network and incorporated gated linear units (GLUs) for complex spectral mapping, which aims to estimate the real and imaginary spectrograms of clean speech from noisy speech for monaural speech enhancement. The convolutional neural network (CNN) model additionally incorporating gating mechanisms was proposed for speech enhancement [22], speech separation [23], and audio classification [24].

Various methods based on U-Net architecture have sprung up in various fields since the U-Net model was first proposed for biological cells segmentation by Ronneberger et al. [25]. Jansson et al. adopted U-Net architecture for the task of singing voice separation [26]. Stoller et al. investigated end-to-end audio source separation and introduced further architectural improvements on U-Net architecture [27]. They proposed Wave U-net, an adaptation of U-Net to the one-dimensional time domain. In addition, for image segmentation, Zhou et al. [28] made further improvements on the model structure and proposed a nested U-Net architecture model which was used for medical image segmentation and achieved better results.

Motivated by the success of the U-Net-based architecture model and gating mechanism as mentioned above, we further develop the nested U-Net (NU-Net) architecture by applying gated linear units on backbone, not including the nested part, to replace the conventional convolution network. We term the separation model based on NU-Net with gated linear units as gated nested U-Net (GNU-Net). The outputs of GNU-Net are further fed into a mask layer to generate two masks of singing voice and accompaniment. So, our proposed model can output singing voice and pure accompaniment simultaneously. We also explored two mask layers, discriminative training network (DTN) and difference mask layer (DML). The experimental results show that, on the whole, the latter is better.

The rest of this paper is organized as follows. We introduce the proposed separation model in Section 2. Section 3 presents the experimental setting. Section 4 presents the results for monaural singing voice and accompaniment separation. We then make our conclusions in Section 5.

2. Gated Nested U-Net Separation Model

We use a fully convolutional neural network that is comprised of a series of convolutional and deconvolutional layers. We first describe the proposed GNU-Net separation model and then detail the gated nested U-Net architecture and two kinds of mask layers.

2.1. Proposed GNU-Net Separation Model

Figure 1 shows the framework of proposed singing voice and accompaniment separation model. The mixed time-domain signals are converted into magnitude and phase spectra using short-time Fourier transformation (STFT). The magnitude spectra are fed into a gated nested U-Net with gated linear units, then the outputs are further fed into a mask layer to produce two masks of singing voice and accompaniment. These two estimated T-F masks are respectively applied to the magnitude spectrum of mixture to get two predicted spectra, which can be transformed into time-domain signals using inverse short-time Fourier transform (ISTFT) with phase information of mixture. Note that the dashed line arrow over the mask layer with dashed line box denotes that this data flow (mixture magnitude spectra) exists only in the training phase. While the full line arrow over the notation of multiply denotes that this data flow (mixture magnitude and phase spectra) exists only in the validation and test phases.

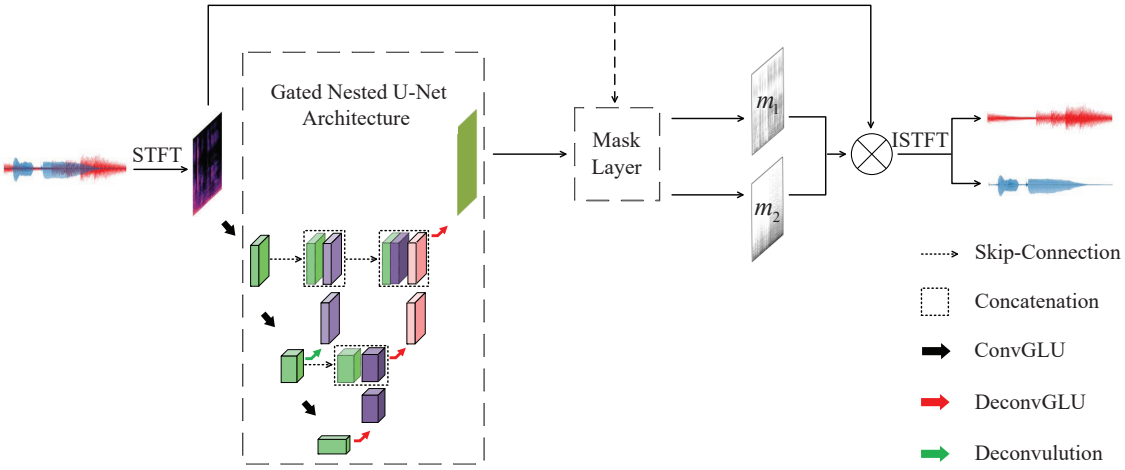


Figure 1. Proposed Separation Model.

As it can be seen in Figure 1, the nested U-Net architecture with dashed line box takes as input the magnitude spectrum of mixture and outputs a 2-dimensional (2-D) feature map (shown in green block) by a series of convolution and deconvolution layers. Those convolution layers and deconvolution layers accomplish the tasks of encoder and decoder respectively. Through the redesigned skip pathways (shown as dotted arrow), the encoder and decoder subnetworks are connected. The dotted box denotes the concatenation operation. The concatenated feature maps are taken as input to perform the deconvolution operation which outputs upsampled feature maps. The skip-connections have been shown to help recovering the full resolution at the network output, where the downsampling operation is performed in the encoder subnetwork and the upsampling operation in the decoder subnetwork.We denote the number of layers of encoder subnetwork as the number of levels of nested U-Net, for example, the nested U-Net in Figure 1 is a 3-level nested N-net, since there are a total of three downsampling operations in the encoder subnetwork.

The outputs of nested U-Net are fed into a mask layer to generate two masks of singing voice and accompaniment. Then, those two masks are applied with mixture spectrum by doing the dot product, respectively, to obtain two estimated source spectra. Through ISTFT operation with the phase of mixture, we can obtain the estimated singing voice and accompaniment time-domain waveform.

2.2. Gated Nested U-Net Architecture

The nested U-Net architecture in Figure 1 can be illustrated in detail through Figure 2, which is an illustration of a 6-level nested U-Net and clearly exhibits the details of operation and skip-connection. The triangularlike pink shadow area denotes nested encoder–decoder, which distinguishes nested U-Net from U-Net. In U-Net architecture for singing voice separation [26], the feature maps of the last convolution layer undergo deconvolution operation the same number of times as convolution. Before each deconvolution operation, it should take a concatenation operation between the outputs of previous deconvolution layers and of the same level convolution layer. This paper also adopts the same concatenation operation.

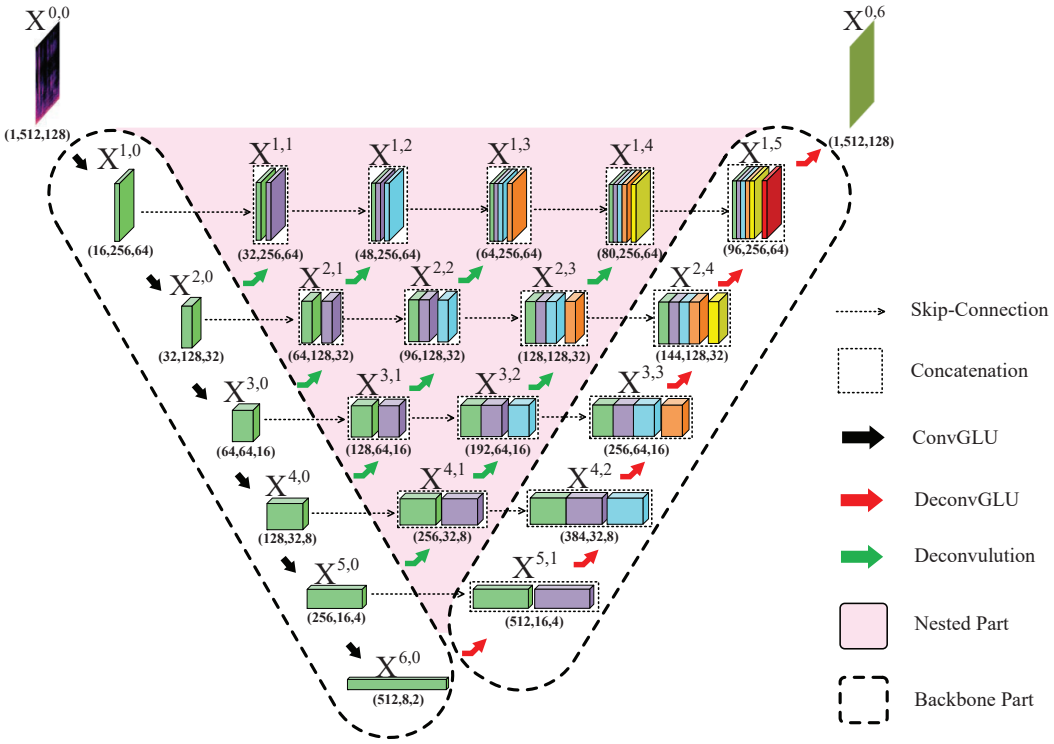


Figure 2. Illustration of 6-level nested U-Net architecture with gated linear units (GLUs) applied only on backbone. Dashed line columns denote the backbone of gated nested U-Net (GNU-Net), and the light-pink triangle denotes the nested part. Cubes denote the output of each layer or concatenation operation, except for $\mathbf{X}^{0,0}$ which denotes the input.

Cubes in Figure 2 denote the outputs of each layer or concatenation operation, except for $\mathbf{X}^{0,0}$, which denotes the input. The numbers below the cubes represent the dimension where (A, B, C) denotes the data has A channels and each feature map has the frequency dimension of B and time dimension of C. $\mathbf{X}^{0,0}$ and $\mathbf{X}^{0,6}$ are the mixture spectrum and output of GNU-net, respectively, the latter would be fed into mask layer to generate the masks. The other cubes are

$$\{\mathbf{X}^{i,j}, i \in \{1,2,3,4,5,6\}, j \in \{0,1,2,3,4,5\}, i+j \leq 6\}, \tag{1}$$

where i indexes the downsampling layer along the encoder and also the row index in nested U-Net architecture, and j indexes the upsampling layer along the decoder. Downsampling is the process

of encoding performed through a convolution operation while upsampling the process of decoding through deconvolution operation.

$$\{\mathbf{X}^{i,0}, i \in \{1,2,3,4,5,6\}\} \tag{2}$$

are the outputs of convolution layers.

$$\{\mathbf{X}^{i,j}, j \neq 0, 2 \leq i+j \leq 6\} \tag{3}$$

are stacks of outputs of convolution and deconvolution layers. They are computed as follows:

$$\mathbf{X}^{i,j} = \begin{cases} \mathcal{H}(\mathbf{X}^{i-1,j}), & j = 0, \quad 1 \leq i \leq 6 \\ \\ \left[\mathbf{X}^{i,j-1}, \mathcal{U}(\mathbf{X}^{i+1,j-1})\right], & j \neq 0, \quad 2 \leq i+j \leq 6, \end{cases} \tag{4}$$

where function $\mathcal{H}(\cdot)$ is convolution operation followed by leaky rectified linear units *(Relu)* activation function and then a batch normalization process, and $\mathcal{U}(\cdot)$ denotes deconvolution operation followed by *Relu* activation and batch normalization process. $[\cdot]$ denotes the concatenation operation, which is denoted by the dotted box in Figures 1 and 2. Specifically, $\mathbf{X}^{3,3} = [\mathbf{X}^{3,2}, \mathcal{U}(\mathbf{X}^{4,2})]$, $\mathbf{X}^{3,2} = [\mathbf{X}^{3,1}, \mathcal{U}(\mathbf{X}^{4,1})]$, $\mathbf{X}^{3,3}$, and $\mathbf{X}^{3,2}$ would undergo a deconvolution process to output a fraction of $\mathbf{X}^{2,4}$ and $\mathbf{X}^{2,3}$. Due to the symmetry of encoding and decoding, $\mathbf{X}^{i,0}$ and $\{\mathcal{U}(\mathbf{X}^{i+1,j}), i+j \leq 5\}$ own the same size.

In conclusion, in Equation (4), the upper-half formulates the encoding process and the outputs of convolution layer; while the lower-half formulates the decoding process and the outputs of concatenation operation. Note that the skip-connection in nested U-Net is designed to concatenate two boxes but not to sum directly, as for image segmentation in U-Net [25].

Owing to the nested skip pathways, nested U-Net could generates full-resolution feature maps at multiple semantic levels, $\{\mathbf{X}^{0,j}, j \in \{1,2,3,4,5\}\}$ (This part is not exit in Figure 2). However, for medical image segmentation, Zhou et al. [28] added a combination of binary cross-entropy and dice coefficient as the loss function to each of the full-resolution feature maps. According to the results of our experiment, $\mathbf{X}^{0,6}$ contains abundant information that is quite qualified for the subsequent mask estimation of each sources. So, our proposed GNU-Net does not include the deconvolution layers used to generate full-resolution feature maps, $\{\mathbf{X}^{0,j}, j \in \{1,2,3,4,5\}\}$.

2.3. Gated Linear Unit

The gating mechanism controls the information flow throughout the network, which potentially allows for modeling more sophisticated interactions [21]. The gated mechanism was first proposed for recurrent neural networks (RNNs) [29] and further developed for CNN [20]. Oord et al. [30] have shown the effectiveness of the LSTM-style gating, which be dubbed gated tahn unit (GTU):

$$tanh(X * W_1 + b_1) \odot \sigma(X * W_1 + b_1) = tanh(V_1) \odot \sigma(V_2), \tag{5}$$

where $V_1 = X * W_1 + b_1$ and $V_2 = X * W_2 + b_2$. W's and b's denote kernels and biases, respectively. σ represents sigmoid function, and $\odot$ means dot product. The gradient of GTUs is

$$\nabla[tanh(V_1) \odot \sigma(V_2)] = tanh'(V_1)\nabla V_1 \odot \sigma(V_2) + \sigma'(V_2)\nabla V_2 \odot tanh(V_1). \tag{6}$$

The gradient gradually vanishes as the network depth increases because of the downscaling factors $tanh'(x)$ and $\sigma'(x)$. To tackle this problem, Dauphin et al. [20] introduced the gated linear unit (GLU):

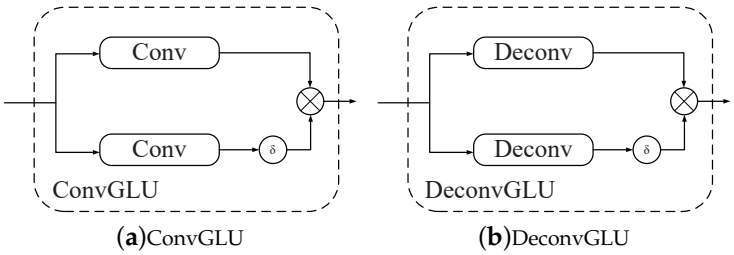
$$(X * W_1 + b_1) \odot \sigma(X * W_2 + b_2) = V_1 \odot \sigma(V_2). \tag{7}$$

The gradient of the GLUs,

$$\nabla[V_1 \odot \sigma(V_2)] = \nabla V_1 \odot \sigma(V_2) + \sigma'(V_2)\nabla V_2 \odot V_1, \tag{8}$$

has a path $\nabla v_1 \odot \sigma(v_2)$ without downscaling for the activated gating units in $\sigma(v_2)$. This can be regarded as a multiplicative skip-connection which helps gradients flow through the layers.

A convolutional GLU block (denoted as "ConvGLU") is illustrated in Figure 3a. A deconvolutional GLU block (denoted as "DeconvGLU") is analogous, except that the convolutional layers are replaced by deconvolutional layers, as shown in Figure 3b.



(a) ConvGLU **(b)**DeconvGLU

Figure 3. Diagrams of a convolutional GLU block and a deconvolutional GLU block, where σ denotes a sigmoid function.

In our proposed GNU-Net model, only the backbone of GNU-Net (two dashed line columns shown in Figure 2) is applied with GLUs, not including nested subnetworks. We use convolution GLU block (black arrow show in Figure 2) and deconvolution GLU block (red arrow show in Figure 2) instead of convolution layer and deconvolution layer in the backbone part. Figure 2 clearly exhibits the details of concatenation operation, skip-connection, and GLU blocks. The triangularlike shadow area same as nested part. So, we rewrite Equation (4) as follows:

$$\mathbf{X}^{i,j} = \begin{cases} \mathcal{H}_{GLU}(\mathbf{X}^{i-1,j}), & j = 0, \quad 1 \le i \le 6 \\ \left[\mathbf{X}^{i,j-1}, \mathcal{U}(\mathbf{X}^{i+1,j-1})\right], & j \ne 0, \quad 2 \le i+j < 6 \\ \left[\mathbf{X}^{i,j-1}, \mathcal{U}_{GLU}(\mathbf{X}^{i+1,j-1})\right], & 1 \le i,j \le 6, \quad i+j = 6 \end{cases}, \tag{9}$$

where function $\mathcal{H}_{GLU}(\cdot)$ and $\mathcal{U}_{GLU}(\cdot)$ are convolution GLU block and deconvolution GLU block, respectively. They are all followed by leaky *Relu* activation function and then a batch normalization process. $\mathcal{U}(\cdot)$ denotes conventional deconvolution operation also followed by *Relu* activation and batch normalization process. Take the same examples as Section 2.2, where $\mathbf{X}^{3,3} = [\mathbf{X}^{3,2}, \mathcal{U}_{GLU}(\mathbf{X}^{4,2})]$, $\mathbf{X}^{3,2} = [\mathbf{X}^{3,1}, \mathcal{U}(\mathbf{X}^{4,1})]$.

2.4. Mask Layer

Ronneberger [25] chose to train two distinctive separation models for two sources exploiting U-Net model. Our goal is to separate singing voice and accompaniment from a mixture simultaneously; so, instead of learning one of the sources as the target, we propose to simultaneously model all the sources. The output of GNU-Net, $\mathbf{X}^{0,6}$, is fed into a mask layer to generate two masks of singing voice and accompaniment. In this paper, we explore two kinds of mask layer, discriminative training network and difference output layer.

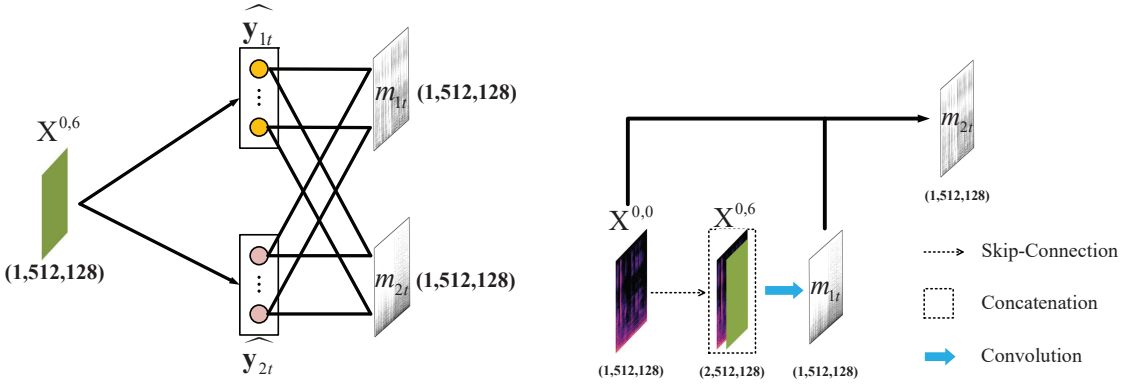
A. Discriminative Training Network (DTN)

Discriminative training network was proposed to jointly train the network with T-F mask function by Po-Sen Huang et al. [15]. In our proposed separation model, the output of GNU-Net, $\mathbf{X}^{0,6}$, is fed into two linear layers, each followed with a *Relu* activation operation. These two linear layers output

magnitude predictions of two sources, $\hat{\mathbf{y}}_{1t}$ and $\hat{\mathbf{y}}_{2t}$, as shown in Figure 4a. Here, we also add an extra layer to the output of the linear layers as

$$\tilde{\mathbf{y}}_{1t} = \frac{\hat{\mathbf{y}_{1t}}}{\hat{\mathbf{y}}_{1t} + \hat{\mathbf{y}}_{2t}} \odot \mathbf{z}_t, \ \tilde{\mathbf{y}}_{2t} = \frac{\hat{\mathbf{y}_{2t}}}{\hat{\mathbf{y}}_{1t} + \hat{\mathbf{y}}_{2t}} \odot \mathbf{z}_t, \tag{10}$$

where the addition, division, and $\odot$ (Hadamard product) operators are elementwise operations. $\mathbf{z}_t$ denotes magnitude spectra of the mixture signals. $\tilde{\mathbf{y}}_{1t}$ and $\tilde{\mathbf{y}}_{2t}$ are two estimated magnitudes of sources $\mathbf{y}_{1t}$ and $\mathbf{y}_{2t}$ through a soft mask function, t = 1, 2, 3, ..., T, where T is the frame length of an input sequence.



(**a**)Discriminative Training Network (DTN) (**b**)Difference Mask Layer (DML)

Figure 4. Two kinds of mask layer.

Equation (10) enforces the constraint that the sum of prediction results is equal to the original mixture. This implies a soft T-F mask function

$$m_{1t} = \frac{\hat{\mathbf{y}_{1t}}}{\hat{\mathbf{y}_{1t}} + \hat{\mathbf{y}_{2t}}}, \qquad m_{2t} = \frac{\hat{\mathbf{y}_{2t}}}{\hat{\mathbf{y}_{1t}} + \hat{\mathbf{y}_{2t}}}. \tag{11}$$

Here, two predictions $\hat{\mathbf{y}}_{1t}$ and $\hat{\mathbf{y}}_{2t}$ should be positive because of *Relu* activation function. Equation (11) implies that $m_{1t} + m_{2t} = 1$. In this way, we integrate the constraints into the network and optimize the network with the masking functions jointly. Although this extra layer is a deterministic layer, the network weights are optimized for the error metric of Equation (11). Thus, it also can be considered that the discriminative training network outputs two masks of singing voice and accompaniment, as shown in Figure 4a.

To reduce the interference from other sources, we adopt the discriminative network training criterion with a simple and useful form [14,15]:

$$\begin{aligned} J_{\mathbf{DIS}} = \frac{1}{2}\sum_{t=1}^{T}(||\mathbf{y}_{1t} - \tilde{\mathbf{y}}_{1t}||^2 + ||\mathbf{y}_{2t} - \tilde{\mathbf{y}}_{2t}||^2 \\ -\gamma||\mathbf{y}_{1t} - \tilde{\mathbf{y}}_{2t}||^2 - \gamma||\mathbf{y}_{2t} - \tilde{\mathbf{y}}_{1t}||^2). \end{aligned} \tag{12}$$

The first half of Equation (12) is general mean squared error (MSE), which directly optimizes the reconstruction objective, adding the extra term $-\gamma||\mathbf{y}_{1t} - \tilde{\mathbf{y}}_{2t}||^2 - \gamma||\mathbf{y}_{2t} - \tilde{\mathbf{y}}_{1t}||^2$ further penalizes the interference from the other source. For our experimental results, we generally achieved higher source-to-interference ratio (SIR) and source-to-distortion ratio (SDR) while slightly lower source-to-artifacts ratio (SAR). We think that an appropriate value of γ would further improve the performance.

B. Difference Mask Layer (DML)

To speed-up learning and improve performance, difference output layer was proposed by Stoller et al. [27]. Similarly, we adopt a difference mask layer (DML) to constrain the mask M_{jt} for source j at time t. If a mixture includes K sources, then enforce $\Sigma_{j=1}^{K} M_{jt} = 1$ that only $K-1$ convolutional filters with a size of 1 are applied to the last feature map of the network, followed by a sigmoid nonlinearity function to estimate the first $K-1$ mask of source signals. The last mask is then simply computed as

$$M_{Kt} = 1 - \Sigma_{j=1}^{K-1} M_{jt}. \tag{13}$$

In our singing voice and accompaniment separation tasks, there are just two sources, so $K = 2$. So, as shown in Figure 4b, the output $\mathbf{X}^{0,6}$ and the mixture spectrum input $\mathbf{X}^{0,0}$ are concatenated, forming a feature map with dimensions $2 \times 512 \times 128$. Through a convolutional network with the filter size of $2 \times 1 \times 1$ followed with a *Sigmoid* activation operation, the difference mask layer outputs a mask of source 1 with the dimensions $1 \times 512 \times 128$. M_{2t}, computed by Equation (13), can be obtained as the mask of source 2 simultaneously.

3. Experiments

3.1. Dataset and Preprocessing

The iKala dataset has been used as a standardized evaluation for the annual Music Information Retrieval Evaluation (MIREX) campaign for several years, so there are many existing results that can be used for comparison. The iKala dataset [31] includes 352 30-second song clips with a sample rate of 44,100 Hz. These clips are recorded from Chinese popular songs performed by professional singers. Only 252 song clips are released as a public subset for evaluation. Each song clip is a stereo recording, with one channel for singing voice and the other for accompaniment. We first downsample the input audio to the same sampling frequency of 8192 Hz as per U-Net model [25], then extract the magnitude spectrum using a 1024-point STFT with 75% overlap. All sample clips are cut into roughly 11 s so that the number of time frame of each patch can be set with 128 (a power of 2 times). The magnitude spectrograms are normalized by $x \rightarrow log(1+x)$. (See Supplementary Materials).

3.2. Evaluation Metrics

To measure the quality of estimated time-domain signal $\hat{v}$ with respect to the original signal v, the source-to-interference ratio (SIR), source-to-artifacts ratio (SAR), and source-to-distortion ratio (SDR) [32] provided in the commonly used BSS EVAL toolbox. The source-to-distortion ratio (SDR) is computed as follows:

$$SDR(\hat{v}, v) = 10log_{10}\left[\frac{< \hat{v}, v >^2 >}{\|\hat{v}\|^2\|v\|^2 - < \hat{v}, v >^2}\right]. \tag{14}$$

Normalized SDR (NSDR) is the improvement of SDR from the original mixture x to the separated singing voice $\hat{v}$, and is commonly used to measure the separation performance for each mixture [12,26]:

$$NSDR(\hat{v}, v, x) = SDR(\hat{v}, v) - SDR(x, v), \tag{15}$$

where $\hat{v}$ is the estimated source signal, v is the reference source signal, and x is the mixed signal.

3.3. Experiment Configurations

The networks are trained on 11-second-long segments. Mean squared error (MSE) is exploited as loss function. ADAM [33] is used as optimizer. The learning rate is set to 10^{-5} with decay rates $\beta_1 = 0.9$. Batch size is 4. Stride size of 2 is used in the convolutional encoder. γ of discriminative training network in Equation (12) is set to 0.05.

The detailed description of GNU-Net is shown in Table 1. The column *Shape* represents the dimension of outputs (cubes in Figure 2). The column *Operation* represents the different neural network operations. F_c equals 16. ConvGLU-2D (A), Deconv-2D (A), and DeconvGLU-2D (A) denote the operations; and A is the output channels of each operation. The filter size is 5×5. $Concat(A, B)$ denotes the concatenation operation of A and B. i in the row *Encoder block* refers to the number of downsampling process. $j = 0, 1 \leq i \leq L$. Note that the Decoder blocks are applied in reverse order, so that j is from level L to 1, $j \neq 0, 2 \leq i + j < L$. In nested part, $1 \leq i, j \leq L, i + j = L$ in backbone part. As it is shown in Figure 2, $L = 6$.

Table 1. Schematic diagram of the proposed GNU-Net architecture.

Block		Operation	Shape
Input			$\mathbf{X}^{0,0}$ = (1,512,128)
Encoder i = 1,...,L		ConvGLU2D($F_c \times 2^{i-1}$)	$\mathbf{X}^{L,0}$ = (512,8,2)
Decoder i = L,...,1	Nested Part	Concat[$\mathbf{X}^{i,j-1}, \mathcal{U}(\mathbf{X}^{i+1,j-1})$] Deconv2D($F_c \times 2^i$)	$\mathbf{X}^{1,L-1}$ = (96,256,64)
	Backbone Part	Concat[$\mathbf{X}^{i,j-1}, \mathcal{U}_{GLU}(\mathbf{X}^{i+1,j-1})$] DeconvGLU2D($F_c \times 2^{i-1}$)	
	Output	DeconvGLU2D(1)	$\mathbf{X}^{0,L}$ = (1,512,128)

ConvGLU-2D (A) denotes the convolutional GLU operation with stride of 2 followed with leaky rectified linear units (ReLU) activation, obtaining that leakiness is 0.2. Deconv-2D (A) and DeconvGLU-2D (A) denote the conventional–deconvolutional and deconvolutional GLU operations, respectively. The deconvolutional operations are both followed with a batch normalization operation and leaky ReLU activation with leakiness of 0.2. Note in decoder block, before the deconvolutional operation we should concatenate the output of the last deconvolutional operation and the previous output in the same level.

Our implementation is similar to that of U-Net [26]. Each encoder layer consists of a convolution layer with stride of 2 instead of pooling process [26]. In the decoder, ReLU is used as activation function. Dropout of 50% is only applied to the first three nested deconvolution layers, $\mathcal{U}(\mathbf{X}^{6,0})$,$[\mathcal{U}(\mathbf{X}^{5,0}), \mathcal{U}(\mathbf{X}^{5,1})]$, $[\mathcal{U}(\mathbf{X}^{4,0}), \mathcal{U}(\mathbf{X}^{4,1}), \mathcal{U}(\mathbf{X}^{4,2})]$.

For the difference mask layer in Section 2.4, the output of last DeconvGLU operation, $\mathbf{X}^{0,6}$, and the input mixture spectrum, $\mathbf{X}^{0,0}$, are concatenated and further fed into a 2-D conventional convolution layer with a stride of 1and kernel size of $2 \times 1 \times 1$ followed with a *Sigmoid* activation, as shown in Figure 4b.

3.4. Comparison with Ideal Time-Frequency Masks

Following the common configurations in [34,35], the ideal time-frequency masks were calculated using STFT with a 32-ms window size and 8-ms hop size with a Hanning window. The ideal masks include the ideal binary mask (IBM), ideal ratio mask (IRM), and Wiener filterlike mask (WFM), which are defined for source i as

$$IBM_i(f,t) = \begin{cases} 1, |S_i(f,t)| > |S_{j \neq i}(f,t)| \\ 0, otherwise, \end{cases} \tag{16}$$

$$IRM_i(f,t) = \frac{|S_i(f,t)|}{\Sigma_{j=1}^{C} |S_j(f,t)|}, \tag{17}$$

$$WFM_i(f,t) = \frac{|S_i(f,t)|^2}{\Sigma_{j=1}^{C}|S_j(f,t)|^2}, \tag{18}$$

where $S_i(f,t) \in \mathbb{C}^{F\times T}$ are the complex-valued spectrograms of clean sources $i = 1, ..., C$.

4. Results

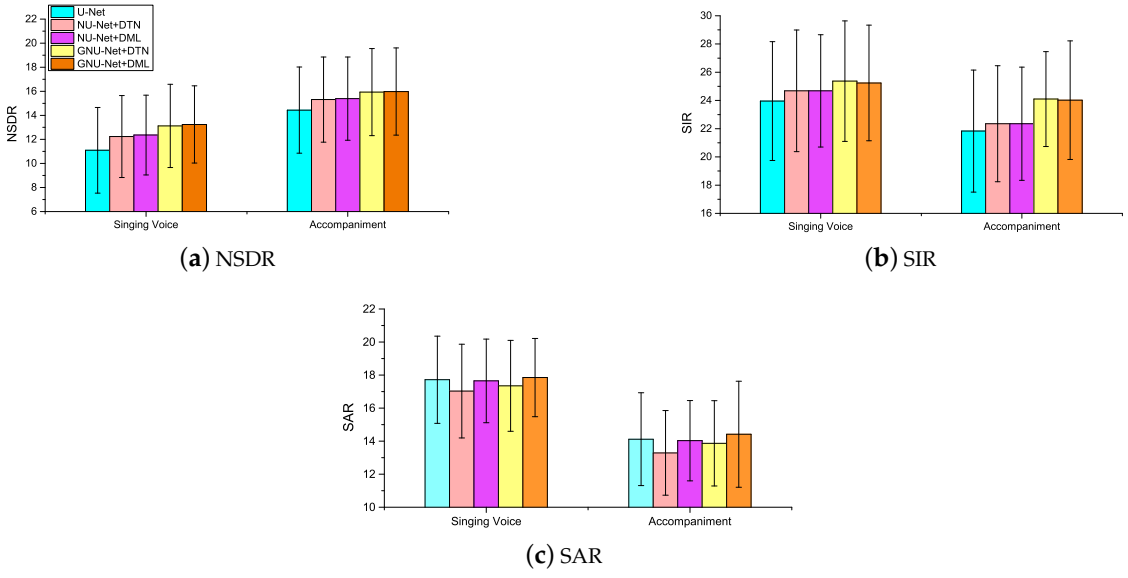
Firstly, the proposed GNU-Net model and two kinds of mask layer were verified by the separation performance, and the effect of the nested U-Net was assessed by comparing with U-Net [26]. Then, a comparison of various networks levels was made on model parameter and system performance to select a proper network level. Finally, the performance of GNU-Net separation model was compared with three models and ideal T-F masks on the iKala dataset.

4.1. Optimizing the Network Model

The performance of GNU-Net separation model was evaluated on iKala dataset. Table 2 shows the performance scores of various models with 6-level nested U-Net and 6-level U-Net [26]. In the first row, the results of singing voice and accompaniment are based on two U-Net separation models, as the U-Net [26] model can output only one source signal, while our proposed model can output estimated singing voice and accompaniment simultaneously. NU-Net denotes nested U-Net without introducing GLUs. The contents of Table 2 are exhibited in another form in Figure 5, which can help to intuitively distinguish various models by the means and variances of various evaluation metrics. From Table 2 and Figure 5, we can conclude the following statements:

Table 2. Comparison between various network models and mask layer on iKala dataset.

	Singing Voice			Accompaniment		
	NSDR	**SIR**	**SAR**	**NSDR**	**SIR**	**SAR**
U-Net [26]	11.09	23.96	17.72	14.44	21.83	14.12
NU-Net+DTN	12.24	24.68	17.03	15.31	22.35	13.29
NU-Net+DML	12.36	24.54	17.62	15.39	22.31	14.03
GNU-Net+DTN	13.12	**25.37**	17.37	15.93	**24.10**	13.87
GNU-Net+DML	**13.24**	25.24	**17.85**	**15.98**	24.02	**14.42**



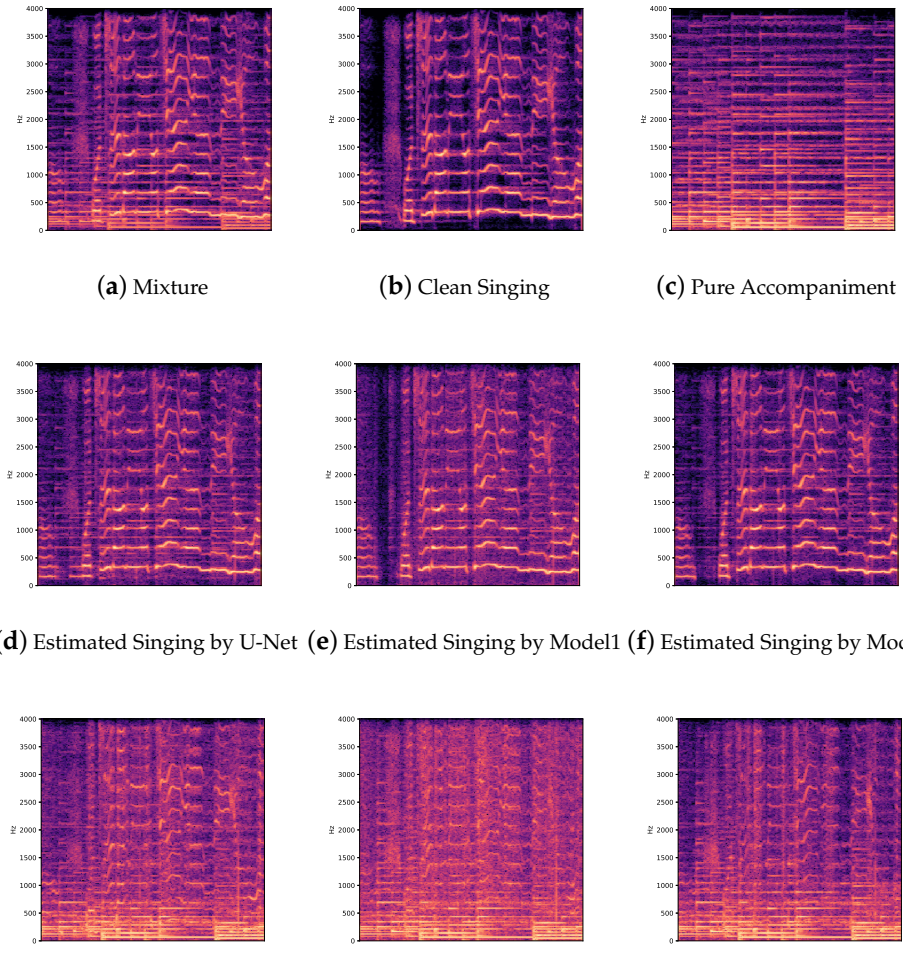
(**a**) NSDR (**b**) SIR

(**c**) SAR

Figure 5. Three evaluation metrics of estimated singing voice and accompaniment by various network models.

(i) Nested U-Net architecture outperforms U-Net architecture, this results verifies that the nested decoder subnetworks can remedy the information loss caused by previous downsampling operations.
(ii) Introducing gated mechanisms can noticeably improve system performance.
(iii) As mask layer, difference mask layer (DML) is superior to discriminative training network (DTN).
(iv) On the whole, the NSDR scores of accompaniment outperform that of singing voice. This may be because in the most general case, the intensity of the accompaniment is greater than that of the singing voice, and accompaniment has more continuous components over time.

Figure 6 shows the magnitude spectra comparison between the estimated sources and original sources. From the estimated magnitude spectra of estimated singing voice, we can noticeably distinguish that our proposed models outperform U-Net model.



(**a**) Mixture (**b**) Clean Singing (**c**) Pure Accompaniment

(**d**) Estimated Singing by U-Net (**e**) Estimated Singing by Model1 (**f**) Estimated Singing by Model2

(**g**) Estimated Accom by U-Net (**h**) Estimated Accom by Model1 (**i**) Estimated Accom by Model2

Figure 6. (**a**) The mixture magnitude spectrogram of a clip in iKala dataset; (**b**,**c**) the ground truth spectra of clean singing voice and pure accompaniment; (**d**–**f**) the magnitude spectra of estimated singing voice by U-Net model and our proposed two models; (**g**–**i**) The magnitude spectra of estimated accompaniment by U-Net model and our proposed two models (model1, NU-Net+DML; model2, GNU-Net+DML). Accom denotes accompaniment.

Some experiments were performed for selection of the depth of network. Table 3 shows the model size and system performances of U-Net [26] architecture and our proposed method (NU-Net and

GNU-Net) with the mask layer of difference mask layer (DML). The numbers of parameters in different methods are based on our implementations. The results of U-Net [26] by our implementation is basically the same as their reported results. The GNU-Net model has the biggest model size compared with U-Net and NU-Net at the same network level and have the best separation performance on NSDR, SIR, and SAR. Compromise the system performance and complexity, 6-level network was selected to adopt for the GNU-Net separation model.

Table 3. Comparison of model size and evaluation results.

Model	Levels	Model Size	Singing Voice			Accompaniment		
			NSDR	SIR	SAR	NSDR	SIR	SAR
U-Net	4	0.61M	9.71	22.72	15.15	13.22	17.78	12.22
	5	2.45M	10.37	23.25	16.87	14.96	21.24	13.53
	6	9.82M	11.09	23.96	17.72	15.31	21.83	14.12
NU-Net+DML	4	0.72M	9.78	22.68	15.23	13.23	17.69	12.33
	5	3.00M	11.98	23.34	16.82	15.04	21.21	13.40
	6	12.07M	12.36	24.54	17.62	15.39	22.31	14.03
GNU-Net+DML	4	1.32M	10.28	22.14	15.88	13.37	22.82	12.46
	5	5.47M	12.91	24.47	17.05	15.64	23.73	13.90
	6	22.21M	**13.24**	**25.24**	**17.85**	**15.98**	**24.02**	**14.42**

4.2. Comparison of Proposed Method with Previous Methods

Finally, the proposed models were also compared to the *RPCA* [12] and *Chimera* [36] models, which produced the highest evaluation scores in the 2016 MIREX Source Separation campaign. Table 4 shows the means of evaluation metrics using iKala dataset. The results of first row of *RPCA* are from their reported paper. The second row shows the results reported in Reference [26], the results are run by the Chimera web server using the improved Chimera network [36]. *NU-Net+DML* and *GNU-Net+DML* denote our proposed methods, which separate singing voice and accompaniment simultaneously, while *Chimera* and *U-Net* separate singing voice and accompaniment using two distinct trained separation models. We can see from Table 4 that the separation performance of our proposed GNU-Net with the mask layer of DML approaches the results of IBM, especially the NSDR of separated singing voice. Our proposed separation model even surpasses IBM in SAR metric for both singing voice and accompaniment.

Table 4. Comparison of proposed methods (NU-Net+DML and GNU-Net+DML) and previous methods using iKala dataset.

	SingingVoice			Accompaniment		
	NSDR	SIR	SAR	NSDR	SIR	SAR
RPCA [12]	6.32	8.14	12.53	0.75	3.23	7.00
Chimera [36]	8.75	21.30	15.64	11.63	20.48	11.54
U-Net [26]	11.09	23.96	17.72	14.44	21.83	14.12
NU-Net+DML	12.36	24.54	17.62	15.39	22.31	13.03
GNU-Net+DML	13.24	25.24	17.85	15.98	24.02	14.42
IBM	14.06	29.00	16.80	16.56	32.78	14.18
IRM	15.25	26.45	18.34	17.48	22.18	16.02
WFM	15.66	28.57	18.55	18.23	29.18	16.00

5. Conclusions

We propose a separation model based on GNU-Net architecture. The outputs of GNU-Net are further fed into a T-F mask layer to generate two masks of singing voice and accompaniment. Then,

those masks along with the magnitude and phase spectra of mixture are transformed into time-domain waveform. We explored two types of T-F mask layer, discriminative training network (DTN) and difference mask layer (DML). The experimental results demonstrate the following:

(i) The nested U-Net architecture outperforms U-Net architecture.
(ii) Introducing gated mechanisms can improve system performance.
(iii) DML is superior to DTN.
(iv) Our proposed GNU-Net separation model outperforms three compared models on three evaluation metrics—NSDR, SIR, and SAR.
(v) Our proposed GNU-Net approaches IBM on NSDR metric and even outperforms IBM on SAR.

More importantly, our proposed model can output the two sources of singing voice and accompaniment simultaneously.

Meanwhile, we have observed that if the prelude or interlude are a little longer, or the energy of accompaniment is much larger than that of singing voice, the system does not perform well. This is what we would focus on and strive to resolve in the future work.

Supplementary Materials: The following are available online at http://www.mdpi.com/2073-8994/12/6/1051/s1.

Author Contributions: Conceptualization, Y.H. and H.G.; methodology, Y.H. and H.G.; software, H.G.; validation, H.G. and Y.H.; resources, Y.H. and H.H.; writing–original draft preparation, Y.H. and H.G.; supervision, Y.H. and H.H.; funding acquisition, Y.H. and H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Natural Science Foundation of China (NSFC) (U1903213, 61761041, 61663044), Natural Science Foundation of the Xinjiang (2016D01C061) and University Scientific Research Project of Xinjiang (XJEDU2017T002).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sharma, B.; Das, R.K.; Li, H. On the importance of audio-source separation for singer identification in polyphonic music. In Proceedings of the Interspeech 2019, Graz, Austria, 15–19 September 2019.
2. Hu, Y.; Liu, G. Separation of singing voice using nonnegative matrix partial co-factorization for singer identification. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, 23, 643–653. [CrossRef]
3. Mesaros, A.; Virtanen, T.; Klapuri, A. Singer identification in polyphonic music using vocal separation and pattern recognition methods. *ISMIR* **2007**, 375–378.
4. Kruspe, A.M.; Fraunhofer, I. Retrieval of textual song lyrics from sung inputs. In Proceedings of the Interspeech 2016, San Francisco, CA, USA, 8–12 September 2016; pp. 2140–2144.
5. Mesaros, A.; Virtanen, T. Automatic recognition of lyrics in singing. *EURASIP J. Audio Speech Music. Process.* **2010**, 546047. [CrossRef]
6. Wang, Y.; Kan, M.Y.; Nwe, T.L.; Shenoy, A.; Yin, J. Lyrically: automatic synchronization of acoustic musical signals and textual lyrics. In Proceedings of the 12th annual ACM international conference on Multimedia, New York, NY, USA, 10–16 October 2004; pp. 212–219.
7. Ikemiya, Y.; Itoyama, K.; Yoshii, K. Singing voice separation and vocal f0 estimation based on mutual combination of robust principal component analysis and subharmonic summation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 2084–2095. [CrossRef]
8. Lin, K.W.E.; Anderson, H.; Agus, N.; So, C.; Lui, S. Visualising singing style under common musical events using pitch-dynamics trajectories and modified traclus clustering. In Proceedings of the 2014 13th International Conference on Machine Learning and Applications, Detroit, MI, USA, 3–5 December 2014; pp. 237–242.
9. Uhlich, S.; Giron, F.; Mitsufuji, Y. Deep neural network based instrument extraction from music. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 2135–2139.

10. Yoo, J.; Kim, M.; Kang, K.; Choi, S. Nonnegative matrix partial co-factorization for drum source separation. In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, USA, 14–19 March 2010; pp. 1942–1945.
11. Rafii, Z.; Liutkus, A.; Stoter, F.R.; Mimilakis, S.I.; FitzGerald, D.; Pardo, B. An overview of lead and accompaniment separation in music. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1307–1335. [CrossRef]
12. Huang, P.S.; Chen, S.D.; Smaragdis, P.; Hasegawa-Johnson, M. Singing-voice separation from monaural recordings using robust principal component analysis. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 57–60.
13. Wang, D L.; Chen, J. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1702–1726. [CrossRef] [PubMed]
14. Huang, P.S.; Kim, M.; Hasegawa-Johnson, M.; Smaragdis, P. Singing-voice separation from monaural recordings using deep recurrent neural networks. In Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014), Taipei, Taiwan, 27–31 October 2014; pp. 477–482.
15. Huang, P.S.; Kim, M.; Hasegawa-Johnson, M.; Smaragdis, P. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 2136–2147. [CrossRef]
16. Fan, Z.C.; Lai, Y.L.; Jang, J.S.R. Svsgan: Singing voice separation via generative adversarial network. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 726–730.
17. He, B.; Wang, S.; Yuan, W.; Wang, J.; Unoki, M. Data augmentation for monaural singing voice separation based on variational autoencoder-generative adversarial network. In Proceedings of the 2019 IEEE International Conference on Multimedia and Expo (ICME), Shanghai, China, 8–12 July 2019; pp. 1354–1359.
18. Stoller, D.; Ewert, S.; Dixon, S. Adversarial semi-supervised audio source separation applied to singing voice extraction. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2391–2395.
19. Mimilakis, S.I.; Drossos, K.; Santos, J.F.; Schuller, G.; Virtanen, T.; Bengio, Y. Monaural singing voice separation with skip-filtering connections and recurrent inference of time-frequency mask. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 721–725.
20. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, Sydney, Australia, 6–11 August 2017; pp. 933–941.
21. Tan, K.; Wang, D.L. Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *28*, 380–390. [CrossRef]
22. Tan, K.; Chen, J.; Wang, D.L. Gated Residual Networks with Dilated Convolutions for Monaural Speech Enhancement. *IEEE/ACM IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 189–198. [CrossRef] [PubMed]
23. Shi, Z.; Lin, H.; Liu, L.; Liu, R.; Han, J.; Shi, A. Deep attention gated dilated temporal convolutional networks with intra-parallel convolutional modules for end-to-end monaural speech separation. *Proc. Interspeech* **2019**, 3183–3187. [CrossRef]
24. Xu, Y.; Kong, Q.; Wang, W.; Plumbley, M.D. Large-scale weakly supervised audio classification using gated convolutional neural network. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017.
25. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical image computing and computer-assisted intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
26. Jansson, A.; Humphrey, E.; Montecchio, N.; Bittner, R.; Kumar, A.; Weyde, T. Singing voice separation with deep u-net convolutional networks. In Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR, Suzhou, China, 23–27 October 2017; pp. 23–27.
27. Stoller, D.; Ewert, S.; Dixon, S. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv* **2018**, arXiv:1806.03185

28. Zhou, Z.; Siddiquee, M.M.R.; Tajbakhsh, N.; Liang, J. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–11.
29. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
30. Oord, A.V.D.; Kalchbrenner, N.; Kavukcuoglu, K. Pixel Recurrent Neural Networks. *arXiv* **2016**, arXiv:1601.06759.
31. Chan, T.S.; Yeh, T.C.; Fan, Z.C.; Chen, H.W.; Su, L.; Yang, Y.H.; Jang, R. Vocal activity informed singing voice separation with the ikala dataset. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, Australia, 19–24 April 2015; pp. 718–722.
32. Vincent, E.; Gribonval, R.; Févotte, C. Performance measurement in blind audio source separation. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1462–1469. [CrossRef]
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint* **2014**, arXiv:1412.6980.
34. Isik, Y.; Roux, J.L.; Chen, Z.; Watanabe, S.; Hershey, J.R. Single-channel multi-speaker separation using deep clustering. *arXiv* **2016**, arXiv:1607.02173.
35. Luo, Y.; Chen, Z.; Mesgarani, N. Speaker-independent speech separation with deep attractor network. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 787–796. [CrossRef]
36. Luo, Y.; Chen, Z.; Ellis, D.P. Deep clustering for singing voice separation. *MIREX, Task of Singing Voice Separation*. 2016; pp. 1–2. Avaliable online: https://www.music-ir.org/mirex/abstracts/2016/LCP1.pdf (accessed on 10 May 2020).