OPEN ACCESS Freely available online

W. Choi, Y.-S. Jeong, J. Kim, J. Chung, S. Jung, and J.D. Reiss, "Conditioned Source Separation by Attentively Aggregating Frequency Transformations With Self-Conditioning" *J. Audio Eng. Soc.*, vol. 70, no. 9, pp. 661–673, (2022 September).
DOI: https://doi.org/10.17743/jaes.2022.0030

Conditioned Source Separation by Attentively Aggregating Frequency Transformations With Self-Conditioning

WOOSUNG CHOI,[1] YEONG-SEOK JEONG,[1] JINSUNG KIM,[1]
(ws_choi@korea.ac.kr) (dnfkdi1995@korea.ac.kr) (wlstjd2003@korea.ac.kr)
JAEHWA CHUNG,[2] SOONYOUNG JUNG,[1] AND JOSHUA D. REISS,[3]* *AES Fellow*
(jaehwachung@knou.ac.kr) (jsy@korea.ac.kr) (joshua.reiss@qmul.ac.uk)

[1] *Department of Computer Science and Engineering, Korea University, Republic of Korea*
[2] *Department of Computer Science, Korea National Open University, Republic of Korea*
[3] *Centre for Digital Music, Queen Mary University of London, London, UK*

Label-conditioned source separation extracts the target source, specified by an input symbol, from an input mixture track. A recently proposed label-conditioned source separation model called Latent Source Attentive Frequency Transformation (LaSAFT)–Gated Point-Wise Convolutional Modulation (GPoCM)–Net introduced a block for latent source analysis called LaSAFT. Employing LaSAFT blocks, it established state-of-the-art performance on several tasks of the MUSDB18 benchmark. This paper enhances the LaSAFT block by exploiting a self-conditioning method. Whereas the existing method only cares about the symbolic relationships between the target source symbol and latent sources, ignoring audio content, the new approach also considers audio content. The enhanced block computes the attention mask conditioning on the label and the input audio feature map. Here, it is shown that the conditioned U-Net employing the enhanced LaSAFT blocks outperforms the previous model. It is also shown that the present model performs the audio-query–based separation with a slight modification.

0 INTRODUCTION

Music source separation aims to extract sources from a given mixture of sources. Non-negative Matrix Factorization (NMF) [1] has been commonly used in early methods [2, 3]. This approach is based on the idea that an audio signal can be represented with fundamental components and their activation coefficients varying over time. Using NMF algorithms, a music signal's power (or magnitude) spectrogram can be decomposed into two non-negative matrices: the basis and coefficient matrices. Here, the basis matrix contains a set of basis vectors (i.e., fundamental spectral components), and the coefficient matrix contains the activation coefficients per time frame for reconstruction.

Provided the pretrained basis vectors are grouped by sources, a specific source s can be separated from a given mixture with two steps. The first step is to search for the most appropriate coefficient matrix to reconstruct the original mixture spectrogram while fixing the basis vectors. Then, the spectrogram of source s was reconstructed by using the subset of the basis matrix, only consisting of basis vectors of s and the corresponding subset of the coefficient matrix.

Inspired by NMF, some Deep Neural Networks (DNNs) [4–6] have been proposed for source separation. For example, [6] proposed a neural block called Latent Source Attentive Frequency Transformation (LaSAFT). A LaSAFT block is also based on the idea that an audio signal can be represented with a weighted sum of fundamental components. LaSAFT blocks were adopted in a label-conditioned source separation called LaSAFT–Gated Point-Wise Convolutional Modulation (GPoCM)–Net [6]. LaSAFT-GPoCM-Net separates the desired source represented by an input label from the mixture, unlike the traditional source separation, in which a model separates a predefined number of sources through a single inference process. It exploits the NMF-inspired feature extraction provided by each LaSAFT block, outperforming the existing label-conditioned models on the MUSDB18 [7] dataset.

*Corresponding author: Joshua D. Reiss, e-mail: joshua.reiss@qmul.ac.uk. Last updated: Jan 24, 2022.

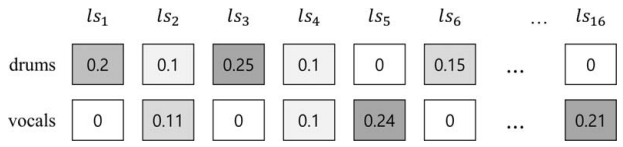

Fig. 1. Relationship modeling: sources and latent sources. ls = latent source.

Similar to NMF, it first extracts features of fundamental spectral components and outputs a weighted sum of them. However, the features of spectral components extracted by a LaSAFT block can no longer be considered basis vectors for the following reasons: they are not non-negative or trained to reconstruct the non-negative input. The differences between them are compared in more detail in SEC. 1.2.

The authors of [6] used the term *latent source* instead to refer to internal vectors. They assumed that a fundamental spectral component was from a virtual latent source. A LaSAFT block is designed to generate feature maps per latent source and outputs the weighted sum of them considering the input label. Here, the weights (or activation coefficients) are determined considering the relationship between the real source specified by the label and latent sources. To model the relationships, they adopted the attention mechanism [8].

Fig. 1 describes an example of relationship modeling between real sources and latent sources based on an attention mechanism in a LaSAFT block, in which there are two sources and 16 latent sources. For latent source ls_k, the LaSAFT block generates an individual feature map V_k. It aggregates these feature maps conditioned on the given source label s_i. It is trained to find a set of the ideal activation coefficients $W_{i,k}$ for the optimal feature aggregation to separate s_i. While preserving $\sum_k W_{i,k} = 1$, $W_{i,j}$ is trained to have a high value if the latent source ls_j is highly relevant to s_i. The LaSAFT block aggregates internal representations by taking $\sum_k W_{i,k} V_k$.

Replacing conventional blocks used in [9] with LaSAFT blocks improved the overall signal-to-distortion ratio (SDR) performance of a Conditioned-U-Net (CU-Net) [10, 6] by 0.97 dB, as reported in [6]. Still, there is room for improvement. The existing LaSAFT block does not provide time-varying activation coefficients, unlike NMF, forcing the same coefficients to be shared across all the time frames. The coefficient modeling is only globally conditioned on the label without any local conditioning. This paper shows that providing weights per time frame improves the existing LaSAFT-based separation frames. Inspired by self-conditioning methods [11], the proposed attention mask modeling is locally conditioned on an audio feature map. Query-side and key-side self-conditioning are proposed to provide separate weights per time frame.

With the modified LaSAFT blocks, the proposed label-conditioned separation model outperforms the previous LaSAFT-GPoCM-Net on several MUSDB18 tasks. It is also shown that the proposed methods with a slight modification can perform audio-query–based separation as discussed in SEC. 4, reporting that it outperforms the existing method [12].

The main contributions of this paper are summarized as follows:

- To break the rigidity of the global conditioning, an enhanced LaSAFT block was proposed by exploiting self-conditioning in the attention mask modeling.
- The enhanced label-conditioned separation model outperforms the previous LaSAFT-GPoCM-Net. An ablation study is provided to validate this approach.
- It is shown that this model can also perform audio-query–based separation with a slight modification.

The remainder of this paper is organized as follows. SEC. 1 overviews the relevant literature. SEC. 2 overviews the baseline CU-Net architecture used in this paper. SEC. 3 presents self-conditioning methods to address the limitation of the existing method. SEC. 4 summarizes the experimental results. This paper is concluded in SEC. 5.

1 BACKGROUND

1.1 NMF-Based Source Separation

Source separation has many applications in audio engineering, such as dialogue enhancement [13] and music source separation. For music source separation, early methods [2, 3] use NMF [1]. NMF algorithms factorize a non-negative matrix into two non-negative matrices. They have been widely used for source separation. The basic idea of NMF-based source separation is to represent an audio signal with a set of fundamental components and their activation coefficients varying over time. Using NMF, a signal's power (or magnitude) spectrogram, which is non-negative by definition, can be decomposed into the basis and coefficient matrices.

The basis matrix contains a set of basis vectors representing elementary components in the spectral domain. With pretrained basis vectors, the frequency spectrum at any time frame can be approximated as a weighted sum of them. From the source separation point of view, basis vectors are fundamental spectral components that can reconstruct the original spectrogram. The coefficient matrix contains the appropriate weights per time frame for reconstruction.

Assuming all the sources are provided in the training dataset, the basis matrix for each source can be learned. Using such a set of pretrained basis matrices, sources can be separated from a given mixture. The separation process consists of analysis and reconstruction. The analysis process creates a matrix containing all the basis vectors by concatenating pretrained basis matrices. Fixing the large basis matrix, the most appropriate coefficient matrix that minimizes the loss between the estimated and original mixture spectrogram can be iteratively found. The reconstruction process estimates the source spectrograms with the coefficient matrix. To separate source s, only the basis vectors learned from s are used for reconstruction instead of the whole vectors in the basis matrix. The unwanted sounds

can be muted by eliminating the other basis vectors during reconstruction.

1.2 NMF-Inspired DNNs

Because DNNs have shown impressive performance in various domains, several DNN models inspired by NMF have been proposed for source separation. For example, DeepNMF [4] is a DNN resulting from unfolding the NMF iterations, in which each iteration has separate parameters. DeepNMF outperformed the traditional NMF-based systems [14–16] on the DSD100 music source separation benchmark [17] as reported in [18].

The set of basis vectors learned by NMF is often called a spectral dictionary. [5] employed the autoencoder [19] mechanism for spectral dictionary training instead of NMF. They reported that the music source separation performance of the autoencoder-based method was superior to that of the NMF. Once spectral dictionaries are learned, [4] and [5] follow the remaining NMF separation scheme. They iteratively search for the appropriate coefficient matrix to reconstruct the mixture.

On the other hand, [6] employs the attention mechanism [20] to model the coefficients directly instead of the iterative search. They proposed a neural block called LaSAFT, for label-conditioned source separation (see SEC. 1.3). Similar to the conventional neural building blocks, such as convolution, it is a sub-component of a deep network. It aims to capture spectral characteristics of the given label (e.g., bass) from the input audio feature map.

A LaSAFT block does not inherit NMF's fundamental properties, in which the input and output are non-negative matrices. It has four main differences from NMF: (1) As a sub-network of a deep network, it takes as input a real-numbered intermediate feature map and the source label. (2) Neural networks directly model the features of spectral components and the appropriate activation coefficients within a single step. (3) Activation coefficients are modeled, conditioned on the given label. (4) Activation coefficients are shared across time frames, whereas NMF's coefficient matrix contains separate activation coefficients per time frame.

A LaSAFT block inherits the idea used in NMF of representing an audio signal with a weighted sum of fundamental components. It first extracts latent source-dependent features and outputs the weighted sum of them. Instead of being searched by iterative search, the weights (or coefficients) are modeled by the attention mechanism. A detailed explanation of LaSAFT is given in SEC. 2.2.2.

1.3 Conditioned Source Separation

Recently, many methods have been proposed for music source separation based on deep learning approaches. A widely used approach trains a neural network that takes an input mixture and estimates the target sources. Whereas some architectures [9, 21–24] estimate only a single source, some methods [25–27] estimate multiple sources simultaneously.

Unlike this approach, a conditioned separation approach trains a network that separates the desired source, characterized by additional input. For example, some studies [6, 10, 28–31] conditioned their networks on source labels, represented as a one-hot vector or word token. If the given label is *bass*, for instance, they separate the bass from the mixture. They usually obtain a latent vector z_E using an embedding layer to represent bass in a dense embedding space. Then conditioning mechanisms, such as the Feature-wise Linear Modulation (FiLM) [32] or Adaptive Instance Normalization [33], modulate internal features with z_E to guide networks to separate bass.

The current state-of-the-art label-conditioned separation model on the MUSDB18 [7] benchmark is LaSAFT-GPoCM-Net [6], which is a variant of CU-Net [10]. Whereas CU-Net uses a fully convolutional layer for each encoding and decoding block, LaSAFT-GPoCM-Net employs convolutional layers followed by a LaSAFT block. It also adopted the Gated Point-wise Convolutional Modulation (GPoCM) for the feature conditioning mechanism, whereas CU-Net uses FiLM [32]. LaSAFT-GPoCM-Net is explained in SEC. 2.

On the other hand, some models are conditioned on time-varying information (e.g., audio or lyrics). For example, audio-query–based separation [12] aims to extract a sound, similar to given sample audio, from a mixture. Query-net used in [12] encodes a query signal, an example of the desired source to be separated, into the latent vector. The separation network is conditioned on the latent vector with Adaptive Instance Normalization [33] and concatenation-based conditioning. This idea was also adopted for hierarchical musical instrument separation [34].

LaSAFT-GPoCM-Net were initially proposed for label-conditioned music source separation. This paper shows that the proposed architecture can also perform audio-query–based source separation [12] with a slight modification. In SEC. 4.8, the performances of the modified network and previous method [12] are compared on the MUSDB18 [7] benchmark.

1.4 Global and Local Conditioning

Whereas models proposed in [12, 34] use only global conditioning, in which features are conditioned on a single compressed latent vector, some models [35, 36] use local conditioning, in which a time-dependent context vector is available for each time frame. [35] conditioned their singing voice separation model on manually aligned lyrics. [30] conditioned their model on a binary instrument activity vector. [36] proposed a unified framework for zero-shot source separation, transcription, and syntheses. Inspired by FiLM [32], they perform information fusion between pitch and timbre representation, which is similar to local conditioning, to separate the desired source.

The attention mask modeling used in the existing LaSAFT [6] block is only globally conditioned on an input label. Inspired by self-conditioning methods [11], the proposed new attention mask modeling is locally conditioned on an input audio feature map. The proposed methods can

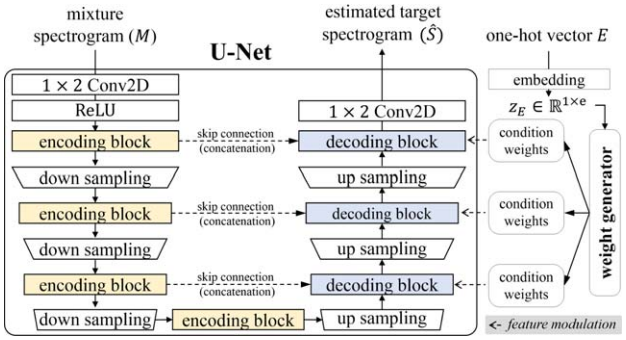

Fig. 2. Baseline architecture (some arrows omitted for clarity). Conv2d = 2D Convolution; ReLU = Rectified Linear Unit.

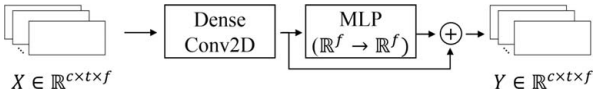

Fig. 3. Encoding block. DenseConv2D = densely connected 2D convolutional block; MLP = Multi-Layer Perceptron.

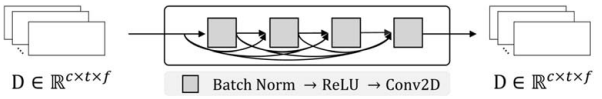

Fig. 4. Densely connected 2D convolutional block (DenseConv2D). Conv2d = 2D Convolution; Batch Norm = Batch Normalization; ReLU = Rectified Linear Unit.

provide separate activation coefficients per time frame by exploiting local conditioning, whereas the existing LaSAFT block cannot.

2 BASELINE ARCHITECTURE

This section describes the baseline architecture used in this paper. The main difference between the baseline and proposed model is the attention mechanism used in each decoding block.

CU-Net of [6, 29] is used for the backbone architecture. As shown in Fig. 2, it receives a mixture spectrogram M and one-hot encoding vector E that specifies the target source. It outputs the estimated spectrogram $\hat{S}$ of the target source. It has a generic U-Net [37], as shown on the left side of Fig. 2.

Here, M and $\hat{S}$ are complex-valued spectrograms of stereo audio signals adopting the Complex-as-Channel (CaC) separation method [9]. Real and imaginary numbers are viewed as separate channels in CaC. Thus, M and $\hat{S}$ have four channels: left-real, left-imaginary, right-real, and right-imaginary. The operation of the U-Net is controlled by the right side of Fig. 2 to separate the target source E.

2.1 Generic U-Net

Generic U-Net follows the typical workflow of the original U-Net [37]. It is an encoder-decoder network with symmetric skip connections. The encoder maps M into downsized spectrogram-like representations using encoding blocks and down-sampling layers. The decoder receives the intermediate results and estimates the spectrogram $\hat{S}$ by applying decoding blocks and up-sampling layers. Feature maps of the same scales are concatenated between the encoder and decoder. Each decoding block takes the concatenation of feature maps as input. These skip connections help the U-Net recover fine-grained details of the target [37].

Following [6], two 1×2 convolutions [6] are used to control the number of channels, as shown in Fig. 2. A 1×2 convolution with c output channels is applied, followed by Rectified Linear Unit (ReLU) [38] to the input spectrogram M. Every internal encoding or decoding block outputs a spectrogram-like tensor with c channels. The output of the last decoding block is fed to another 1×2 convolution with four output channels to obtain the estimated spectrogram $\hat{S}$. No activation function is applied to the final output.

Each down/up-sampling layer is a strided/transposed-convolution that halves/doubles the scale in both time and frequency domains of an input tensor. Unlike down/up-sampling layers, each encoding/decoding block transforms an input spectrogram-like tensor into an equally sized tensor. SECS. 2.1.1 and 2.1.2 describe encoding and decoding block architecture, respectively, in detail. Although the encoding phase is similar in nature to the existing U-Nets [37, 21, 9], the decoding phase is different. During decoding, the intermediate features are modulated by the conditioning mechanism for the target source separation. SEC. 2.2 describes the conditioning mechanisms.

2.1.1 Encoding Block

Fig. 3 illustrates the architecture of an encoding block. An encoding block receives spectrogram-like tensor X and outputs the same-sized tensor Y.[1] It first applies a densely connected 2D convolutional block (DenseConv2D) [39], which is widely used in deep learning–based source separation models [22, 9, 29]. As shown in Fig. 4, a DenseConv2D consists of densely connected composite layers, in which each layer is a stack of 2D convolution, Batch Normalization [40], and ReLU [38]. The ith composite layers take an input with $i \times c$ channels and generate the output feature map with c channels.

Then, a Multi-Layer Perceptron (MLP) takes the output of the dense block and captures the overall frequency patterns observed in the mixture spectrogram. It is a stack of two affine transformation layers with Batch Normalizations [40] and ReLU [38] activations. It maps an input spectral vector in $\mathbb{R}^f$ to an output vector in $\mathbb{R}^f$ with a hidden layer, which has $\lceil \mathbb{R}^f/16 \rceil$ units.

Finally, the outputs of two blocks are summed as the outcome. This architecture is the same as the building block called Time-Frequency Convolutions with Time Distributed Fully-connected network (TFC-TDF) proposed in [9], which showed the best performance among the five building blocks.

[1] This paper uses X and Y to denote an input and output tensor of a random block. X and Y do not have a global meaning; that is, they have different meanings in each section.

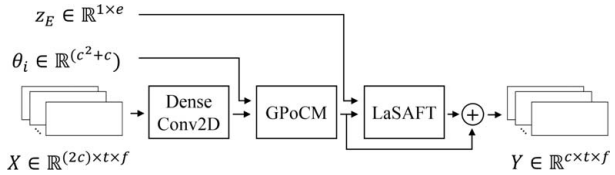

Fig. 5. Decoding block. DenseConv2D = densely connected 2D convolutional block; GPoCM = Gated Point-Wise Convolutional Modulation; LaSAFT = Latent Source Attentive Frequency Transformation.

2.1.2 Decoding Block

The same decoding block architecture used in [6] is utilized. As shown in Fig. 5, the ith decoding block receives a dense embedding vector z_E, the set θ_i of parameters for feature modulation, and a spectrogram-like tensor $X \in \mathbb{R}^{(2c)\times t\times f}$. It outputs $Y \in \mathbb{R}^{c\times t\times f}$. Note that X has twice as many channels as Y (i.e., $2c$ channels) because each decoder takes a concatenation of intermediate feature maps (i.e., c channels from the previous layer and c channels from the corresponding encoding layer). As mentioned in SEC. 2.2, θ_i is generated by the condition weight generator.

The ith decoding block first applies a DenseConv2D to X, similar to the first phase of an encoding block. The only difference is that the first layer of the DenseConv2D receives an input with $2c$, not c, because of the skip connection. The other intermediate feature maps in the decoding block are c-channeled, as in the encoding block.

The remaining modules, namely GPoCM and LaSAFT, modulate internal feature maps conditioned on the label. A GPoCM layer modulates the output of the dense block with θ_i. A LaSAFT block takes the modulated feature map and z_E as input and generates a feature map considering the target source E. Finally, the outputs of GPoCM and LaSAFT are summed as the outcome Y of the whole block, as shown in Fig. 5.

2.2 Conditioning Mechanisms

As in [6], GPoCM and LaSAFT is used to condition the generic U-Net on the given label.

2.2.1 Gated Point-Wise Convolutional Modulation

GPoCM [6] is used to globally condition an intermediate feature map on the dense embedding vector $z_E \in \mathbb{R}^{1\times e}$. A set of condition parameters are first generated for feature modulation with the *condition weight generator* as shown in the right side of Fig. 2. The condition weight generator is a stack of two affine transformation layers with Weight Normalizations [41] and ReLU [38] activations. The condition weight generator takes z_E and generates $\Theta = \{\theta_1, \theta_2 \ldots, \theta_L\}$, in which L denotes the number of decoding blocks and θ_i denotes a set of parameters for the ith decoding block. As illustrated in Fig. 5, θ_i is provided to the GPoCM layer of the ith decoding block for feature modulation, in which X is an intermediate feature map in the block.

Fig. 6 describes how a GPoCM layer modulates an internal feature map conditioned on θ_i. Suppose that the goal

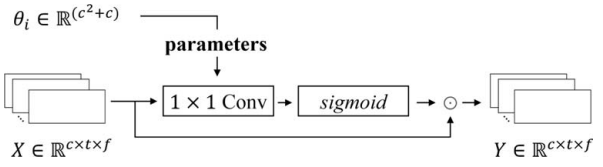

Fig. 6. Feature Modulation in the ith decoding block. conv: Convolution.

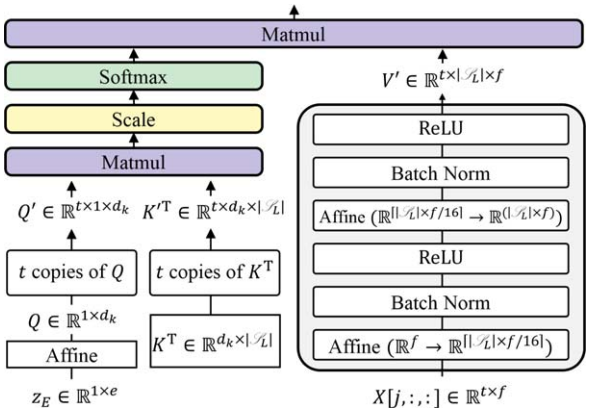

Fig. 7. Workflow of the Original LaSAFT Block. Batch Norm = Batch Normalization; ReLU = Rectified Linear Unit.

is to modulate an internal feature map $X \in \mathbb{R}^{c\times t\times f}$ of the ith decoding block, in which c, t, and f denote the number of channels, time frames, and frequency bins, respectively. As shown in Fig. 6, a GPoCM layer first applies a 1×1 convolution (also known as point-wise convolution) of which parameters are given by $\theta_i \in \mathbb{R}^{(c^2+c)}$. Then, it applies a sigmoid function and outputs a Hadamard product ($\odot$) of the sigmoid output and X. It should be noted that θ_i was generated by the *condition weight generator* for the input z_E.

2.2.2 LaSAFT

A LaSAFT [6] block aims to extract features for conditioned separation based on latent source analysis. Assuming there are $|\mathcal{S}_L|$ latent sources, it aims to generate feature maps per latent source and output the weighted sum of them conditioning on z_E. $|\mathcal{S}_L|$ is usually larger than the number of real sources in the training dataset. Fig. 7 illustrates the workflow of a LaSAFT block to generate the jth output channel, in which the input is given by the jth channel of $X \in \mathbb{R}^{c\times t\times f}$ (i.e., $X[j,:,:]$).

As shown on the right side of the figure, a LaSAFT block has an MLP to extract $|\mathcal{S}_L|$ spectral feature maps. Similar to an encoding block's MLP, it is a stack of two affine transformation layers with Batch Normalizations [40] and ReLU [38] activations. Whereas an encoding block's MLP maps an input spectral vector in $\mathbb{R}^f$ to an output vector in $\mathbb{R}^f$ with a hidden layer of $\lceil \mathbb{R}^f/16 \rceil$ neurons, it maps an input vector in $\mathbb{R}^f$ to an output vector in $\mathbb{R}^{(|\mathcal{S}_L|\times f)}$ with a hidden layer of $\lceil |\mathcal{S}_L| \times \mathbb{R}^f/16 \rceil$ neurons. By applying it to the jth channel of X, $V' \in \mathbb{R}^{t\times|\mathcal{S}_L|\times f}$ in which there are $|\mathcal{S}_L|$ spectral feature maps for each frame, is obtained. From the perspective of the attention mechanism [8], V' is considered

a value representation. The remaining task is aggregating $|\mathcal{S}_L|$ feature maps to obtain the output representation by taking a weighted average of them considering the given label E.

As shown on the left side of Fig. 7, it first applies an affine transformation that maps z_E onto the query space to obtain $Q \in \mathbb{R}^{1\times d_k}$, in which d_k is the dimensionality of the query and key vectors. Meanwhile, it has a learnable weight matrix $K \in \mathbb{R}^{|\mathcal{S}_L|\times d_k}$, in which $K[n] \in \mathbb{R}^{d_k}$ is a representation of the nth latent source. An unnormalized relevance score between the target and the nth latent source can be obtained by taking the dot product of Q and each $K[n]$.

To match the shape of V', Q and K^T are duplicated to obtain $Q' \in \mathbb{R}^{t\times 1\times d_k}$ and $K'^{\mathrm{T}} \in \mathbb{R}^{t\times d_k\times|\mathcal{S}_L|}$, respectively. The score is normalized by a softmax after being scaled by $\sqrt{d_k}$, which prevents the product from growing significantly large [8]. It finally outputs a new representation as follows:

$$\text{Attention}\,(Q', K', V') = \text{softmax}\left(\frac{Q'K'^{T}}{\sqrt{d_k}}\right)V'. \qquad (1)$$

Note that $\text{softmax}(Q'K'^{T}/\sqrt{d_k})$ does not have to be computed repeatedly for every channel. This term is computed only once and shared by all the channels.

3 THE PROPOSED METHODS

This section first exposes the limitation of the existing method. Also, this section presents self-conditioning methods to provide time frame–dependent weights for latent feature aggregation. The proposed methods enable the attention mask to be conditioned not only on the label but also on the input audio feature map. They extract audio content–aware query/key vectors for attention by explicitly considering the input audio features.

3.1 Limitation of the Existing Method

The existing method described in SEC. 2.2.2 only captures symbolic relations between the target source symbol and latent sources while ignoring the audio contents. This approach is sufficient only if the following assumption holds: a symbolically labeled source can be represented with a certain weighted average of latent sources' features *regardless of time and audio contents*. However, the assumption is too rigid, making the existing method less expressive.

The coefficient modeling used in the existing method is only globally conditioned on the label without any local conditioning. *Listening mechanisms* are proposed to condition the attention mask modeling not only on the label globally but also on the input audio feature map locally. The proposed methods enable the block to compute different attention coefficients per time frame, whereas the existing method forces it to use the identical score for every frame.

3.2 Query-Side Listening

The query-side listening mechanism generates the query vectors considering the audio contents and input condition.

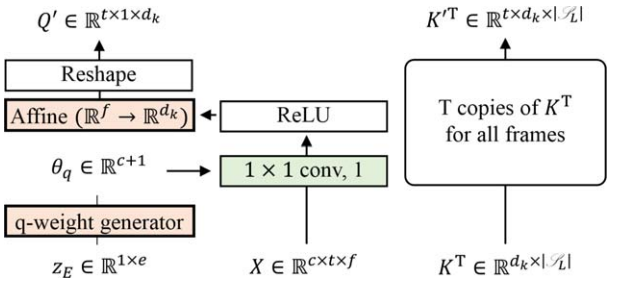

Fig. 8. Query-side Listening Mechanism. conv = Convolution; q-weight = query weight; ReLU = Rectified Linear Unit.

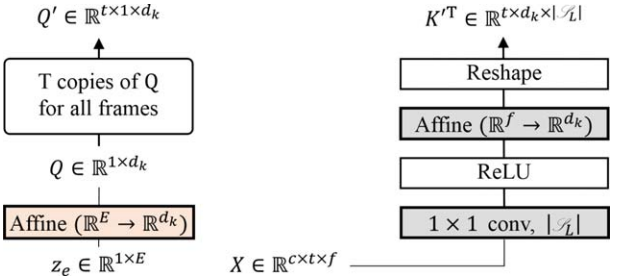

Fig. 9. Key-side Listening Mechanism. conv = Convolution; ReLU = Rectified Linear Unit.

For given audio contents $X \in \mathbb{R}^{c\times t\times f}$, a 1 × 1 convolution is applied to X. It receives the parameters θ_q from the query weight generator (q-weight generator in Fig. 8), as GPoCM does. The query weight generator takes z_E and generates θ_q, enabling the convolution to extract meaningful features from X considering the target source.

Then a stack of a ReLU, affine transformation, and reshape layer are applied to obtain query vectors $Q' \in \mathbb{R}^{t\times 1\times d_k}$, as shown in Fig. 8. By this listening mechanism, the model can take specific context-dependent query vectors considering both the target symbol and audio contents. However, it does not conduct any local conditioning on the key-side. To match the shape of the query, it simply duplicates K^T (see SEC. 2.2.2) to obtain $K'^{\mathrm{T}} \in \mathbb{R}^{t\times d_k\times|\mathcal{S}_L|}$. It forces the same key vectors to all the frames, ignoring the audio contents.

3.3 Key-Side Listening

The key-side listening mechanism first applies a 1 × 1 convolution with $|\mathcal{S}_L|$ channels into X to extract latent source-dependent features. After the normalization and activation, an affine transformation maps the latent source-dependent features to the key-side spectral space $\mathbb{R}^{d_k}$.

Finally, a reshape layer is applied to generate key vectors $K'^{T} \in \mathbb{R}^{t\times d_k\times|\mathcal{S}_L|}$. This approach can generate more flexible key vectors than the previous approach. As shown in Fig. 9, a simple affine transformation is applied to obtain the symbolic query vectors from z_E. The symbolic query vectors are duplicated to match the size of key vectors.

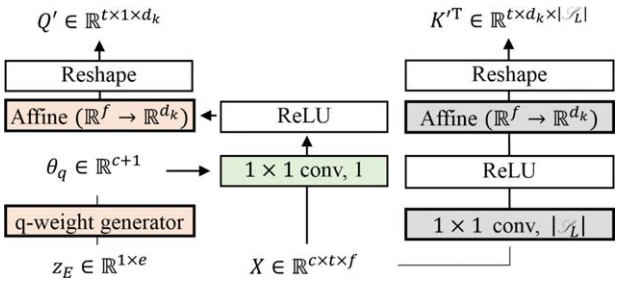

Fig. 10. Query-key listening mechanism. conv = Convolution; q-weight = query-weight; ReLU = Rectified Linear Unit.

3.4 Query-Key Listening

The query-key listening combines both query and key-side listening as shown in Fig. 10.

4 EXPERIMENT

4.1 Dataset

The MUSDB18 dataset [7] was used. It includes 86 training, 14 validation, and 50 test multitracks. Every multitrack is recorded as stereo waveform sampled at 44,100 Hz and has the mixture and four sources (vocals, drums, bass, and other). SDR [42] was reported for the evaluation metric. The official MUSDB18 evaluation tool[2] was used. The median SDR over all the tracks in the test set was taken following the benchmark rule, and the mean SDR over three runs was reported.

4.2 Training

Each model was trained to minimize the mean squared error between the complex-valued spectrograms on top of the CaC framework [9]. An input spectrogram M is in $\mathbb{R}^{4\times t\times f}$, in which f is determined by the Short-Time Fourier Transform (STFT) parameters. Then, t was set to 128, which means that spectrograms with 128 time frames were used as input for every experiment. It has four channels by the CaC framework (see SEC. 2.1). Following [6], c was set to 24, which means that every internal encoding or decoding block outputs a spectrogram-like tensor with 24 channels.

For validation, l_1 loss of the ground-truth waveform and estimated waveform, which is restored from the estimated spectrogram, was used. For data augmentation, mixture was generated by mixing sources from different tracks [18]. Models were trained using the Adam [43] optimizer. Depending on model sizes and listening mechanisms, a learning rate between 10^{-4} and 10^{-3} was used. More details are available online.[3]

4.3 Ablation Study

An ablation study was performed to validate the superiority of the proposed listening mechanism compared with the original LaSAFT listening mechanism. The overall experimental setup of their ablation study was followed. As in [6], different STFT parameters were used for their ablation study and the state-of-the-art model development. Because it takes a lot of resources to train the proposed architecture with a Fast Fourier Transform (FFT) window size of 4,096 (e.g., 2 weeks with four *GeForce RTX 2080ti*s), an FFT window size of 2,048 and hop-size of 1,024 are used. The existing LaSAFT-GPoCM-Net[4] [6] with the same STFT parameters as the reference model is used in the ablation study.

The baseline is similar but not identical to the reference model. Like the reference model, the baseline has four encoding layers and three decoding layers (i.e., $L = 3$, see SEC. 2.2). The same configuration setup of the reference model [6] was also followed for each DenseConv2D (SEC. 2.1.1), embedding layer (SEC. 1.3), and weight generator (SEC. 2.2.1). For each DenseConv2D, five convolution layers with kernel size 3×3 and a growth rate [39] of 24 were used. For the embedding layer, the dimensionality e was set to 32. For the weight generator, the hidden dimension is set to $eL = 32 \times 3$, in which L is the number of decoding blocks. As mentioned in SEC. 2.1.1, the bottleneck factor [6, 9] is set to 16. Two 1×2 convolutions [6] were also used to control the number of channels. A 1×2 convolution with 24 output channels is applied, followed by ReLU [38] to the input spectrogram M. Every internal encoding or decoding block outputs a spectrogram-like tensor with 24 channels. The output of the last decoding block is fed to another 1×2 convolution with four output channels to obtain the estimated spectrogram $\hat{S}$. Like the reference model, every model has four encoding blocks and three decoding blocks.

The differences between the baseline and reference model are as follows: (1) The baseline uses LaSAFT blocks only in the decoder, whereas [6] uses LaSAFT blocks in the encoder as well, and (2) a light version [29] of latent source-aware frequency transformation method with $|\mathcal{S}_L| = 16$, whereas [6] used the original heavy analyzer with $|\mathcal{S}_L| = 6$. By using a shared affine transformation layer ($\mathbb{R}^{\lceil f/16\rceil} \to \mathbb{R}^f$) for all the latent source feature maps instead of the last affine layer of Fig. 7, the number of parameters can be significantly reduced, as discussed in [29].

CU-Nets with different attention mechanisms were implemented based on the baseline architecture described in SEC. 2. The authors compare the performance of five different CU-Nets, namely the small version of the existing LaSAFT-GPoCM-Net [6] (i.e., the reference model), baseline, baseline with query-side listening, baseline with key-side listening, and baseline with query-key listening. Each model was trained for 400 epochs (approximately 1.3 million steps), and the configuration with the lowest validation loss was evaluated.

The results in Table 1 are summarized. The first row is the SDR performance of the reference model. As shown in the table, the baseline is slightly superior to the LaSAFT-GPoCM-Net structure. It is observable that adding the

[2] https://github.com/sigsep/sigsep-mus-eval.
[3] https://github.com/ws-choi/LaSAFT-Net-v2.
[4] The last row of Table 1 in [6].

Table 1. Results of the ablation study [metric: signal-to-distortion ratio (SDR), higher is better].

Model	Query listen	Key listen	Vocals	Drums	Bass	Other	Average
LaSAFT-GPoCM-Net (small) [6]			6.96	5.84	5.24	4.54	5.64
Baseline	✗	✗	7.04 ± 0.05	6.06 ± 0.83	5.14 ± 0.06	4.55 ± 0.21	5.70 ± 0.07
+ Query Listening	√	✗	7.10 ± 0.15	6.06 ± 0.10	5.29 ± 0.04	**4.71** ± 0.04	5.79 ± 0.05
+ Key Listening	✗	√	7.05 ± 0.01	6.03 ± 0.03	5.32 ± 0.06	4.68 ± 0.12	5.77± 0.03
Proposed architecture (small)	√	√	**7.20** ± 0.03	**6.11** ± 0.04	**5.48** ± 0.04	4.65 ± 0.12	**5.86** ± 0.03

Bold indicates the highest SDR in each source. GPoCM = Gated Point-Wise Convolutional Modulation; LaSAFT = Latent Source Attentive Frequency Transformation.

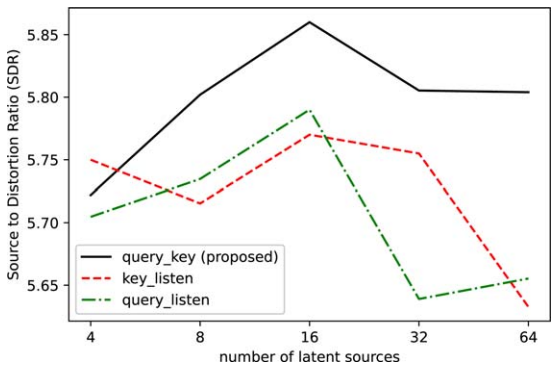

Fig. 11. Effect of the number of latent sources [x axis: number of the latent sources; y axis: signal-to-distortion ratio (SDR), higher is better].

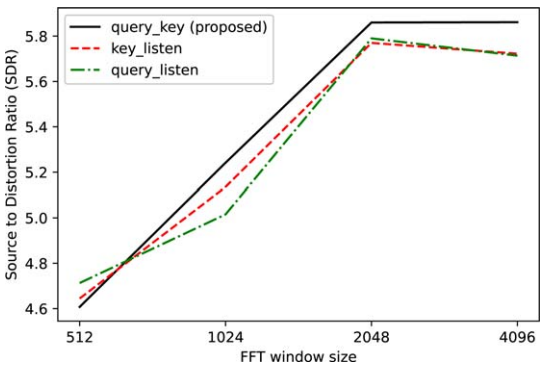

Fig. 12. Effect of FFT window size [x axis: FFT window size; y axis: signal-to-distortion ratio (SDR), higher is better]. FFT = Fast Fourier Transform.

query-side listening mechanism to each decoding block improves performance in general. Making each decoding block listen to the audio contents for key vector generation also enhances performance in general. The proposed model, which employs both listening mechanisms, outperforms the other models on every task except for the "other" stem.

4.4 Effect of the Number of Latent Sources

This section investigates the effect of $|\mathcal{S}_L|$, the number of latent sources on SDR performance. The three architectures, namely the baseline with query-side listening (*query-listen* in Fig. 11), baseline with key-side listening (*key-listen*), and baseline with query-key listening [*query-key (proposed)*], were trained with varying $|\mathcal{S}_L| \in \{4, 8, 16, 32, 64\}$. The same configurations used in the previous section except for $|\mathcal{S}_L|$.

Fig. 11 illustrates the effect of $|\mathcal{S}_L|$ on the average SDR scores over four sources of each architecture. Increasing $|\mathcal{S}_L|$ from 4 to 16 tends to improve the performance of every architecture. However, having too many latent sources degrades SDR performance, similarly to the existing latent component analysis–based solutions in different domains [44, 45]. Especially, models with $|\mathcal{S}_L| > 16$ tend to generate more artifacts. The average Signal-to-Artifacts Ratio [42] of the three architectures is 6.24, 6.22, and 6.11 dB when $|\mathcal{S}_L|$ is 16, 32, and 64, respectively.

4.5 Effect of FFT Window Size

This section investigates the effect of the FFT window size on SDR performance. Similarly to the process described in the previous section, the three architectures were trained with varying FFT window size $\in \{512, 1{,}024, 2{,}048, 4{,}096\}$. The hop-size was set to be half of the FFT window size. The same configurations used in SEC. 4.3, including the number of frames in an input spectrogram, except for the STFT parameters, were used.

Fig. 12 illustrates the effect of the FFT window size on the average SDR scores over four sources. A model with a larger FFT window size usually performs better if every model has the same capacity. However, it does not hold if a model takes too high–resolution spectrogram. In Fig. 12, models with an FFT window size of 4,096 were slightly inferior to their counterpart models with an FFT window size of 2,048. More parameters need to be stacked to improve the SDR performance of the proposed architecture with an FFT window size of 4,096. The following section shows such an example.

4.6 Comparison With State-of-the-Art Models

To compare with the existing methods, a *large version of the proposed architecture* or *proposed architecture (large)* was trained in short. Following *the large version of LaSAFT-GPoCM-Net*,[5] or *LaSAFT-GPoCM-Net (large)*, the FFT window size was set to 4,096 and hop-size to 1,024. Many

[5] The last row of Table 2 in [6].

Table 2. Comparison with other models [metric: signal-to-distortion ratio (SDR), higher is better].

Model	Conditioned?	No. of params	Vocals	Drums	Bass	Other	Average
X-UMX [27]	✗	35.6 M	6.61	6.47	5.43	4.64	5.79
D3Net [24]	✗	N/A*	7.24	7.01	5.25	4.53	6.01
Demucs [26]	✗	265.7 M	6.84	6.86	7.01	4.42	6.28
ResUNetDecouple+ [46]	✗	>400 M	8.98	6.62	6.04	5.29	6.73
Meta-TasNet [28]	√	45.5 M	6.40	5.91	5.58	4.19	5.52
AMSS-Net [47]	√	6.6 M	6.78	5.92	5.10	4.51	5.58
LaSAFT-GPoCM-Net (large) [6]	√	31.5 M	7.33	5.68	5.63	4.87	5.88
Proposed architecture (large)	√	7.9 M	**7.78** ± 0.16	**6.25** ± 0.24	**5.64** ± 0.20	**5.09** ± 0.03	**6.19** ± 0.02

*The official repository is opened but written in NNABLA [48].
AMSS = Audio Manipulation on a Specific Source; D3Net = densely connected multi-dilated DenseNet; GPoCM = Gated Point-Wise Convolutional Modulation; LaSAFT = Latent Source Attentive Frequency Transformation; M = million; Meta-TasNet = meta-learning–inspired model for music source separation; NNABLA = Neural Network Libraries; X-UMX = CrossNet-Open-Unmix.

existing methods [22, 24, 6] also used this configuration. Also, the *proposed architecture (large)* has more blocks than the models in SEC. 4.3: five encoding blocks and four decoding blocks. For the other hyper-parameters, the same setup as the proposed model in SEC. 4.3 was used.

LaSAFT-GPoCM-Net (large) and the proposed architecture (large) follow the same differences between LaSAFT-GPoCM-Net (small) and the baseline, mentioned in SEC. 4.3. Also, each LaSAFT block in the proposed method employs the Query-key listening mechanism, whereas each LaSAFT block in the existing architecture does not. Each model was trained for 2 million steps, and the configuration with the lowest validation loss was evaluated.

Table 2 compares the performance of the proposed architecture (large) and existing models. The first row shows the performance of X-UMX [27], an enhanced Open-Unmix (UMX) [23]. The second row shows the performance of densely connected multi-dilated DenseNet (D3Net) [24]. The third row shows the performance of Demux [26], a waveform-to-waveform model with a U-Net [37] and bidirectional long short-term memory [49]. The fourth shows the performance of ResUNetDecouple+ [36]. Unlike the models above, the fifth, sixth, and seventh rows show the performance of label-conditioned models: meta-learning–inspired model for music source separation (Meta-TasNet) [28], Audio Manipulation on a Specific Source (AMSS)–Net [47], and LaSAFT-GPoCM-Net (large) [6], respectively. Meta-TasNet is a waveform-to-waveform model of which the *parameter generator* predicts the parameters. AMSS-Net was initially proposed for audio manipulation (e.g., low-pass filter) on specific sources conditioned on textual queries. The sixth row shows its performance reported in [47], in which the authors trained it only for source separation. The last row shows the performance of LaSAFT-GPoCM-Net (large), the current state-of-the-art label-conditioned model.

As shown in Table 2, the large version of the proposed model outperforms the existing conditioned models. It is worthy to note that the proposed model outperforms LaSAFT-GPoCM-Net (large) with a smaller number of parameters. Whereas the LaSAFT-GPoCM-Net (large) has 31.5 million parameters, the proposed architecture has only 7.9 million parameters because of the parameter sharing method [29] mentioned in SEC. 4.3. Also, its performance is comparable to Demucs [26], which is not a label-conditioned mode.

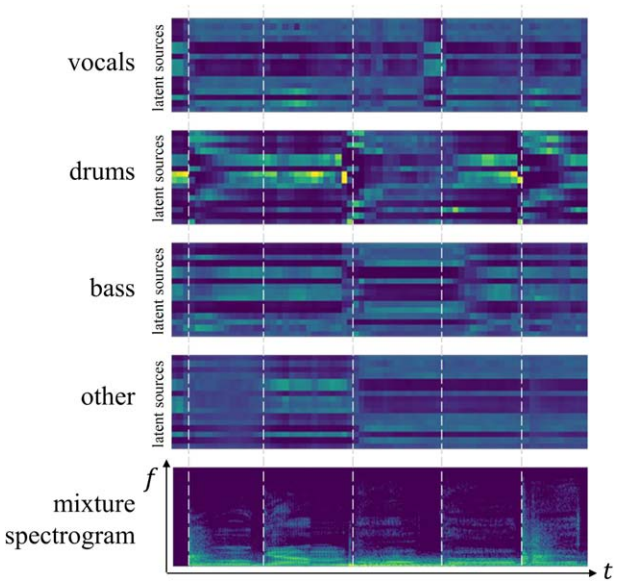

Fig. 13. Visualization of attention coefficients (input: mixture).

4.7 Qualitative Analysis

The proposed method computes individual attention coefficients per frame considering input audio features. To verify this ability, attention coefficients generated by a LaSAFT block in a trained model are visualized. The last block of a pretrained model, which generates interesting and easy-to-interpret attention maps, is chosen in this section.

Fig. 13 visualizes attention coefficients of four stems and the magnitude spectrogram of the input mixture. The input mixture was a piece of rap music. Each plotted matrix of a source is the attention coefficients (i.e., transposed softmax($Q'K'^T/\sqrt{d_k}$) $\in \mathbb{R}^{T\times|\mathcal{S}_L|}$). A white dashed line is plotted when the spectrum dramatically changes over time.

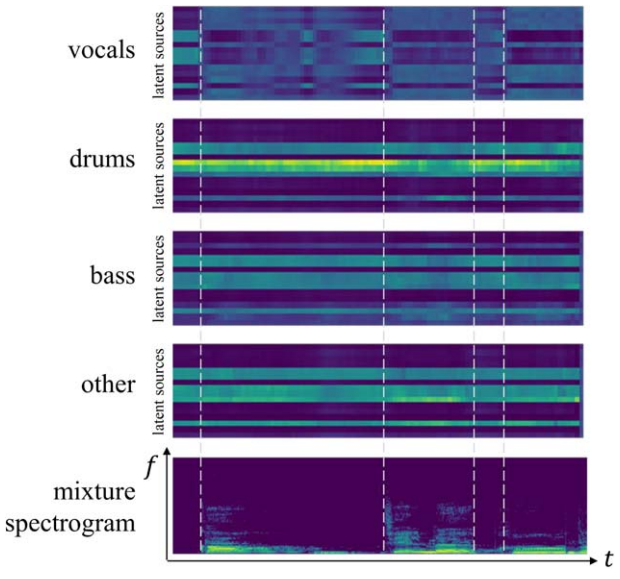

Fig. 14. Visualization of attention coefficients (input: vocals).

The attention coefficients can be observed to also have a similar evolving tendency with the audio contents.

Whereas the input used in Fig. 13 is a mixture of various sources, the input used in Fig. 14 contains only vocals. Because drums, bass, and other are irrelevant to the signal, their attention coefficients are nearly constant. It is worthy to note that the activation patterns are almost the same across sources when there are no acoustic activities. These patterns are *learned representations* of *silent activity*. Also, the reverse patterns can be observed when there are sound activities.

Finally, the results generated by LaSAFT-GPoCM-Net and the proposed model are compared with several audio samples in the demonstration site.[6]

4.8 Audio-Query–Based Separation

With a slight modification on the query-side listening, the proposed model can perform audio-query–based source separation [12]. As introduced in [12], audio-query–based separation aims to extract a sound, similar to given sample audio, from a mixture.

Whereas the proposed network is conditioned on a one-hot vector E, audio-query–based separation is conditioned on a sample audio track. The network was modified by replacing the embedding layer that maps E to a dense embedding vector $z_E \in \mathbb{R}^{1\times e}$ (see Fig. 2) with query-net [12]. Query-net extracts a dense query vector $z_Q \in \mathbb{R}^{1\times e}$ from sample audio. The modified model listens to audio contents conditioned on a query vector instead of a symbol. It (i.e., proposed audio query) was evaluated by using the same evaluation scheme of [12], and the results are shown in Table 3.

[6] https://ws-choi.github.io/LASAFT-Net-v2/demo/index.html.

Table 3. Audio-query–based source separation [metric: signal-to-distortion ratio (SDR), higher is better].

Model	Vocals	Drums	Bass	Other	Average
Lee [12]	5.48	4.59	3.45	3.26	4.20
Proposed audio query	**7.04**	**5.77**	**5.30**	**4.53**	**5.65**

5 CONCLUSION

This paper exposes the limitation of the existing LaSAFT block and proposes an enhanced LaSAFT block, introducing a combination of two proposed listening mechanisms. The proposed architecture employing the enhanced LaSAFT blocks outperforms the existing label-conditioned separation models on the MUSDB18 benchmark. With a slight modification, the proposed model can also perform audio-query–based separation. For future work, one can design a few-shot or zero-shot separation method by extending this approach.

6 ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2021R1A6A3A03046770, 2021R1A2C2011452).

7 REFERENCES

[1] D. D. Lee and H. S. Seung, "Algorithms for Non-Negative Matrix Factorization," in *Proceedings of the 13th International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 13, pp. 535–541 (Denver, CO) (2000 Jan.).

[2] A. Ozerov and C. Fevotte, "Multichannel Nonnegative Matrix Factorization in Convolutive Mixtures for Audio Source Separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 3, pp. 550–563 (2010 Mar.). http://doi.org/10.1109/TASL.2009.2031510.

[3] T. Virtanen, "Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria," *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 3, pp. 1066–1074 (2007 Mar.). http://doi.org/10.1109/TASL.2006.885253.

[4] J. Le Roux, J. R. Hershey, and F. Weninger, "Deep NMF for Speech Separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 66–70 (Brisbane, Australia) (2015 Apr.). https://doi.org/10.1109/ICASSP.2015.7177933.

[5] K. Osako, Y. Mitsufuji, R. Singh, and B. Raj, "Supervised Monaural Source Separation Based on Autoencoders," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11–15 (New Orleans, LA) (2017 Mar.). https://doi.org/10.1109/ICASSP.2017.7951788.

[6] W. Choi, M. Kim, J. Chung, and S. Jung, "LaSAFT: Latent Source Attentive Frequency Transformation for Conditioned Source Separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 171–175 (Toronto, Canada) (2021 Jun.). https://doi.org/10.1109/ICASSP39728.2021.9413896.

[7] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "MUSDB18 - A Corpus for Music Separation," *Zenodo* (2017 Dec.). http://doi.org/10.5281/zenodo.1117371.

[8] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6000–6010 (Long Beach, CA) (2017 Dec.).

[9] W. Choi, M. Kim, J. Chung, D. Lee, and S. Jung, "Investigating U-Nets With Various Intermediate Blocks for Spectrogram-Based Singing Voice Separation," in *Proceedings of the 21st Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 192–198 (Montreal, Canada) (2020 Oct.).

[10] G. Meseguer-Brocal and G. Peeters, "Conditioned-U-Net: Introducing a Control Mechanism in the U-net For Multiple Source Separations," in *Proceedings of the 20th Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 159–165 (Delft, The Netherlands) (2019 Nov.).

[11] S. Birnbaum, V. Kuleshov, S. Z. Enam, P. W. Koh, and S. Ermon, "Temporal FiLM: Capturing Long-Range Sequence Dependencies With Feature-Wise Modulations," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 10287-10298 (Vancouver, Canada) (2019 Dec.), https://dblp.org/search?q=Temporal%20FiLM%3A%20Capturing%20Long-Range%20Sequence%20Dependencies%20with%20Feature-Wise%20Modulations.

[12] J. H. Lee, H.-S. Choi, and K. Lee, "Audio Query-Based Music Source Separation," in *Proceedings of the 20th Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 878–885 (Delft, The Netherlands) (2019 Nov.).

[13] J. Paulus, M. Torcoli, C. Uhle, et al., "Source Separation for Enabling Dialogue Enhancement in Object-Based Broadcast With MPEG-H," *J. Audio Eng. Soc.*, vol. 67, no. 7/8, pp. 510–521 (2019 Jul.). https://doi.org/10.17743/jaes.2019.0032.

[14] J. Eggert and E. Korner, "Sparse Coding and NMF," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, vol. 4, pp. 2529–2533 (Budapest, Hungary) (2004 Jul.). https://doi.org/10.1109/IJCNN.2004.1381036.

[15] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative Matrix Factorization With the Itakura-Saito Divergence: With Application to Music Analysis," *Neural Comput.*, vol. 21, no. 3, pp. 793–830 (2009 Mar.). https://doi.org/10.1162/neco.2008.04-08-771.

[16] F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative NMF and Its Application to Single-Channel Source Separation," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 865–869 (Singapore) (2014 Sep.).

[17] A. Liutkus, F.-R. Stöter, Z. Rafii, et al., "The 2016 Signal Separation Evaluation Campaign," in P. Tichavský, M. Babaie-Zadeh, O. J. J. Michel, and N. Thirion-Moreau (Eds.), *Latent Variable Analysis and Signal Separation*, Lecture Notes in Computer Science, vol. 10169, pp. 323–332 (Springer, Cham, Switzerland, 2017). https://doi.org/10.1007/978-3-319-53547-0_31.

[18] S. Uhlich, M. Porcu, F. Giron, et al., "Improving Music Source Separation Based on Deep Neural Networks Through Data Augmentation and Network Blending," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 261–265 (New Orleans, LA) (2017 Mar.). https://doi.org/10.1109/ICASSP.2017.7952158.

[19] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data With Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507 (2006 Jul.). https://doi.org/10.1126/science.1127647.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (San Diego, CA) (2015 May).

[21] A. Jansson, E. Humphrey, N. Montecchio, et al., "Singing Voice Separation With Deep U-Net Convolutional Networks," in *Proceedings of the 18th Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 745–751 (Suzhou, China) (2017 Oct.).

[22] N. Takahashi, N. Goswami, and Y. Mitsufuji, "MMDenseLSTM: An Efficient Combination of Convolutional and Recurrent Neural Networks for Audio Source Separation," in *Proceedings of the 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 106–110 (Tokyo, Japan) (2018 Sep.). https://doi.org/10.1109/IWAENC.2018.8521383.

[23] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - A Reference Implementation for Music Source Separation," *J. Open Source Softw.*, vol. 4, no. 41, paper 1667 (2019 Sep.). http://doi.org/10.21105/joss.01667.

[24] N. Takahashi and Y. Mitsufuji, "Densely Connected Multi-Dilated Convolutional Networks for Dense Prediction Tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 993–1002 (Nashville, TN) (2021 Jun.).

[25] J.-Y. Liu and Y.-H. Yang, "Dilated Convolution With Dilated GRU for Music Source Separation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4718–4724 (Macao, China) (2019 Aug.). http://doi.org/10.24963/ijcai.2019/655.

[26] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," *arXiv preprint arXiv:1911.13254* (2019 Nov.). https://doi.org/10.48550/arXiv.1911.13254.

[27] R. Sawata, S. Uhlich, S. Takahashi, and Y. Mitsufuji, "All for One and One for All: Improving Music Separation by Bridging Networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 51–55 (Toronto, Canada) (2021 Jun.). https://doi.org/10.1109/ICASSP39728.2021.9414044.

[28] D. Samuel, A. Ganeshan, and J. Naradowsky, "Meta-Learning Extractors for Music Source Separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 816–820 (Barcelona, Spain) (2020 May). https://doi.org/10.1109/ICASSP40776.2020.9053513.

[29] Y.-S. Jeong, J. Kim, W. Choi, J. Chung, and S. Jung, "LightSAFT: Lightweight Latent Source Aware Frequency Transform for Source Separation," in *Proceedings of the Music Demixing (MDX) Workshop* (Online) (2021 Nov.).

[30] O. Slizovskaia, G. Haro, and E. Gómez, "Conditioned Source Separation for Musical Instrument Performances," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 2083–2095 (2021 May). https://doi.org/10.1109/TASLP.2021.3082331.

[31] O. Slizovskaia, L. Kim, G. Haro, and E. Gomez, "End-to-End Sound Source Separation Conditioned on Instrument Labels," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 306–310 (Brighton, UK) (2019 May). https://doi.org/10.1109/ICASSP.2019.8683800.

[32] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "FiLM: Visual Reasoning With a General Conditioning Layer," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 3942-3951 (New Orleans, LA) (2018 Feb.).

[33] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization," in *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), pp. 1501–1510 (Venice, Italy) (2017 Oct.).

[34] E. Manilow, G. Wichern, and J. Le Roux, "Hierarchical Musical Instrument Separation," in *Proceedings of the 21st Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 376–383 (Montreal, Canada) (2020 Oct.).

[35] C. Jeon, H. Choi, and K. Lee, "Exploring Aligned Lyrics-Informed Singing Voice Separation," in *Proceedings of the 21st Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 685–692 (Montreal, Canada) (2020 Oct.).

[36] L. Lin, Q. Kong, J. Jiang, and G. Xia, "A Unified Model for Zero-Shot Music Source Separation, Transcription and Synthesis," in *Proceedings of the 22nd Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 381–388 (Online) (2021 Nov.).

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Eds.), *International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, vol. 9351, pp. 234–241 (Springer, Cham, Switzerland, 2015). https://doi.org/10.1007/978-3-319-24574-4_28.

[38] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 315–323 (Fort Lauderdale, FL) (2011 Apr.).

[39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269 (Honolulu, HI) (2017 Jul.).

[40] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 448–456 (Lille, France) (2015 Jul.).

[41] T. Salimans and D. P. Kingma, "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 29, pp. 901–909 (Barcelona, Spain) (2016 Dec.).

[42] E. Vincent, R. Gribonval, and C. Févotte, "Performance Measurement in Blind Audio Source Separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469 (2006 Jul.). https://doi.org/10.1109/TSA.2005.858005.

[43] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)* (San Diego, CA) (2015 May).

[44] S. Grant and J. R. Cordy, "Estimating the Optimal Number of Latent Concepts in Source Code Analysis," in *Proceedings of the 10th IEEE Working Conference on Source Code Analysis and Manipulation (SCAM)(SCAM)*, pp. 65–74 (Timişoara, Romania) (2010 Sep.). https://doi.org/10.1109/SCAM.2010.22.

[45] J.-L. Xu and C.-S. Zhu, "Personalized and Accurate QoS Prediction Approach Based on Online Learning Matrix Factorization for Web Services," in *Proceedings of the ITM 4th Annual International Conference on Information Technology and Applications (ITA)*, vol. 12, paper 03027 (Guangzhou, China) (2017 Sep.). https://doi.org/10.1051/itmconf/20171203027.

[46] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, "Decoupling Magnitude and Phase Estimation With Deep ResUNet for Music Source Separation," in *Proceedings of the 22nd Conference of the International Society for Music Information Retrieval (ISMIR)*, pp. 342–349 (Online) (2021 Nov.).

[47] W. Choi, M. Kim, M. A. Martínez Ramírez, J. Chung, and S. Jung, "AMSS-Net: Audio Manipulation on User-Specified Sources With Textual Queries," in *Proceedings of the 29th ACM International Conference on Multimedia (MM)*, pp. 1775–1783 (Online) (2021 Oct.). https://doi.org/10.1145/3474085.3475323.

[48] T. Narihira, J. Alonsogarcia, F. Cardinaux, et al., "Neural Network Libraries: A Deep Learning Framework Designed From Engineers' Perspectives," *arXiv preprint arXiv:2102.06725* (2021 Jun.). https://doi.org/10.48550/arXiv.2102.06725.

[49] A. Graves and J. Schmidhuber, "Framewise Phoneme Classification With Bidirectional LSTM and Other Neural Network Architectures," *Neural Netw.*, vol. 18, no. 5, pp. 602–610 (2005 Jul.). https://doi.org/10.1016/j.neunet.2005.06.042.

THE AUTHORS


Woosung Choi


Yeong-Seok Jeong


Jinsung Kim


Jaehwa Chung


Soonyoung Jung


Joshua D. Reiss

Woosung Choi received his Ph.D. degree from Korea University in 2021. Currently, he is a postdoctoral visiting fellow at the Centre for Digital Music, based at Queen Mary University of London. He is interested in machine learning for audio and speech. Specifically, his research interests include speech enhancement, music source separation, source-aware audio manipulation, and multimodal learning for developing easy-to-use audio editing interfaces.

•

Yeong-Seok Jeong received his Bachelor's degree from Hallym University in 2021. Currently, he is pursuing an M.S. degree at Korea University. He is interested in Machine Learning for audio and speech. Specifically, his research interests include speech enhancement, music source separation, and representation learning for music sources.

•

Jinsung Kim is currently an M.S. student at Korea University. He received his Bachelor's degree from Korea University in 2021. His research interests include Machine Learning for speech, voice, and audio signal processing. He is particularly interested in speech separation, voice conversion, and audio representation learning.

•

Jaehwa Chung is an associate professor at the Department of Computer Science at Korea National Open University. He received M.S. and Ph.D. degrees at the Department of Computer Science Education at Korea University, Korea. He has published 30 research papers and authored seven books. His research interests include spatio-temporal data management, large-scale data analysis, and audio source separation.

•

Soonyoung Jung is a professor in the Department of Computer Science and Engineering at Korea University. He received his B.S., M.S., and Ph.D. degrees in computer science from Korea University in 1990, 1992, and 1997, respectively. In 2006–2007, he was a visiting professor in the Department of Computer & Information Science & Engineering, University of Florida. His research interests include deep-learning–based audio processing, voice conversion, automatic speech recognition, and generative modeling.

•

Josh Reiss is Professor of Audio Engineering with the Centre for Digital Music at Queen Mary University of London. He has published more than 200 scientific papers (including over 50 in premier journals and six best paper awards) and co-authored two books. His research has been featured in dozens of original articles and interviews on TV, on radio, and in the press. He is a Fellow and currently President of the Audio Engineering Society and chair of their Publications Policy Committee. He co-founded the highly successful spin-out company, LandR, and recently co-founded Tonz and Nemisindo, also based on his team's research. He maintains a popular blog, YouTube channel, and Twitter feed for scientific education and dissemination of research activities.