Refining BSRNN: Analysis and Extension of Music Source Separation with Band-Split RNN

Sahand Akbarzadeh
Islamic Azad University
Mashhad, Iran
sahandevs@gmail.com

I. Introduction

The advancement of Music Source Separation (MSS) is pivotal in audio processing, offering significant applications in areas such as music remixing, restoration, and educational tools. Recent progress in this field has been propelled by developments in neural network architectures, yet a persistent challenge remains: the specificity of models to individual instruments, which hampers their reusability in transfer learning contexts.

This paper proposes an innovative refinement to the Band-Split Recurrent Neural Network (BSRNN) [1] architecture, which originally employed a frequency-domain approach to address the distinct characteristics of musical signals. A notable limitation of the BSRNN model lies in its instrument-specific feature extraction, constraining its adaptability across different musical contexts. Our research addresses this issue by integrating expert-derived knowledge to establish a universal set of sub-bands, which are applicable to a variety of instruments. This strategy aims to create a more versatile and broadly applicable feature extraction process within the BSRNN framework.

Central to our methodology is the implementation of a multi-task learning [2] paradigm. This paradigm facilitates the training of a foundational model wherein the feature extractor is agnostic to specific instruments. Such a model is designed to be adaptable and efficient, capable of processing a diverse array of musical inputs without necessitating complete retraining. This approach is anticipated to not only conserve training resources but also enhance the model's accuracy and efficiency in source separation.

The structure of the paper is as follows: Section II examines the original BSRNN architecture, identifying both its strengths and areas for improvement. Section III elaborates on our proposed methodology, focusing on the universal sub-band definition and multi-task learning framework. Section IV describes the experimental setup and metrics for evaluating our model's performance. Section V presents our empirical results, illustrating the improvements our modifications bring to the BSRNN model. Finally, Section VI concludes with a discussion on the broader implications of our findings for MSS research and outlines potential future directions for this line of inquiry.

II. Review of BSRNN

BSRNN is structured around a series of distinct modules designed for effective MSS. These include the Band Split module, the Band and Sequence Modeling module, and the Mask Estimation module. Additionally, the framework incorporates pre-processing and post-processing steps for optimal input and output handling.

A. Pre-processing and Post-processing Steps

In the pre-processing step, audio is loaded, normalized, and split into equal-sized chunks. Each chunk undergoes the Short-Time Fourier Transform (STFT) to convert it into the frequency domain for input into the BSRNN model. The post-processing reverses this process, converting frequency domain data back into audio signals. While the original BSRNN paper did not specify the normalization technique, we apply root mean square (RMS) and peak normalization to the entire waveform and standardization to each STFT chunk.

B. Band Split Module

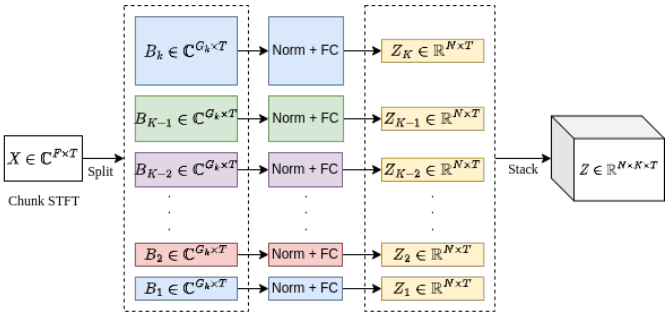

Fig1: Band Split Module

The Band Split Module divides a complex spectrogram, into K frequency subbands. We approach the conversion of complex numbers by treating their real and imaginary parts as distinct channels. Each subband is normalized and transformed through a layer normalization and a fully-connected layer, creating K separate real-valued subband features. These features are then combined into a single fullband feature tensor, pre-

serving the unique normalization and transformation applied to each subband.

In the Band Split section, it's noted that the original paper outlines several band split schemas tailored for different instruments. These schemas vary the inputs and the number of outputs in this layer, consequently altering the architecture. Among these, the v7 schema is highlighted as the closest to optimal and most performant, as per the original authors. In Section III.A, we will delve deeper into this module and propose our own schema. This proposed schema aims to develop a generic Band Split module, adaptable for any instrument.

C. Band and Sequence Modeling Module

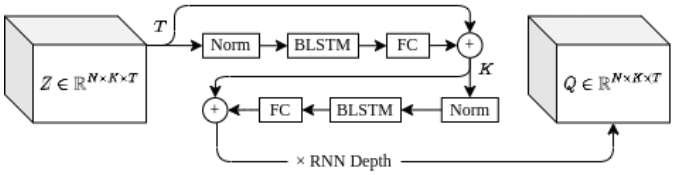

Fig2: Band and Sequence Modeling Module

The Band and Sequence Modeling Module interleaves processing at two levels: sequence and band. It uses two types of RNN layers designed to handle the temporal sequence and the frequency bands of the input. The sequence-level RNN deals with the temporal aspect, applying the same process to each frequency subband because they share the same dimensions. This design allows for simultaneous processing of all subbands, making the model more efficient.

At the band level, the RNN focuses on the relationships within each frequency band across all subbands, capturing the detailed features necessary for effective separation. The design of both RNNs includes a normalization step, followed by a BLSTM layer for capturing patterns in both forward and backward time sequences, and a fully connected layer for the actual modeling work. Residual connections help the network learn more effectively by linking the input and output of the modeling layers.

By stacking multiple RNN layers, the model's depth and complexity increase, which can lead to better performance.

D. Mask Estimation Module

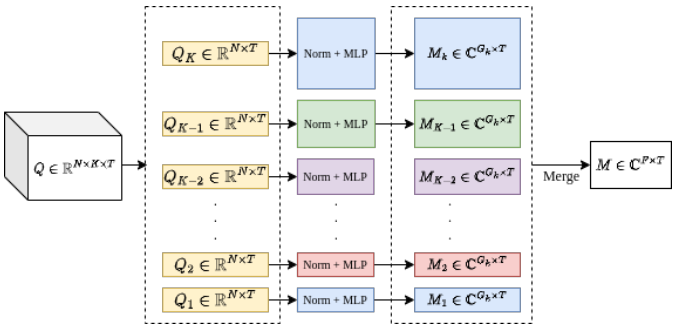

Fig3: Mask Estimation Module

The Mask Estimation Module is responsible for creating complex-valued time-frequency (T-F) masks that are used to isolate the target source from the mixture. The process starts with the (Q), which contains (K) separate features for each subband. These features have been processed by the previous modules and are now ready for mask creation.

Each of the (K) subband features undergoes normalization and then passes through a Multilayer Perceptron (MLP) with a single hidden layer. The MLP's job is to produce the real and imaginary parts of the T-F masks. These masks, denoted as (M_i), are what allow the model to differentiate and extract the desired source from the complex audio mixture.

This approach of using an MLP for mask estimation is based on findings from [3], which suggest that MLPs can more accurately estimate T-F masks compared to simpler fully connected (FC) layers.

Like in the Band Split Module, each subband feature in the Mask Estimation Module is treated with its own normalization and MLP. The resulting masks from all subbands are then combined into a fullband T-F mask (M). This fullband mask is applied to the mixed spectrogram (X) to yield the separated target spectrogram (S), the final output representing the isolated audio source.

III. Refined BSRNN

The pursuit of a more generalized and effective Music Source Separation model leads us to propose refinements to the BSRNN framework. These refinements are aimed at developing a universal feature extractor that can be applied to various musical instruments and contexts. In this section, we will detail the methodology for creating such a universal feature extractor and discuss its integration within the Band Split module. Furthermore, we will outline a multi-task learning approach that enables the training of this feature extractor to function universally.

A. Universal sub-bands

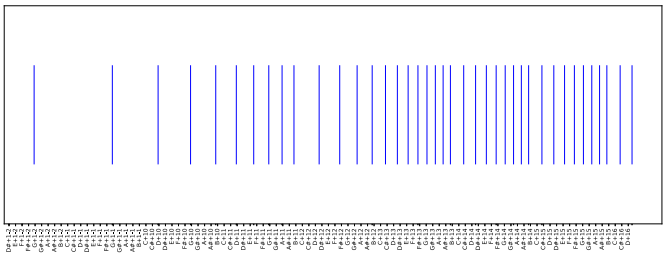

Fig4: v7 Band Split schema in log scale

In reevaluating the BSRNN architecture, the use of expert knowledge for defining frequency band ranges tailored to specific instruments is a central theme. However, we posit that the schemas provided by the original author for these bands may not adequately capture the task-specific requirements.

In 4, where we have plotted the band ranges against musical notes, it becomes evident that the distribution of these ranges is uneven—some bands encompass entire octaves, while others cover only a single note.

This visualization suggests that the original author may have prioritized certain frequency ranges based on the presumption that they hold more significance, possibly due to the presence of the fundamental frequencies of the targeted instrument. Nonetheless, a harmonically rich sound is characterized not only by its fundamental frequency but by a spectrum of overtones that contribute to its timbre. Recognizing this, our critique is that the band distribution in the original BSRNN model does not fully represent the harmonic complexity of musical sounds.

To effectively address the limitations of the band range definitions in the original BSRNN architecture, it's essential to consider the harmonic structure of musical timbre. An instrument's timbre is composed of its fundamental frequency alongside a series of overtones. These overtones are integral to the sound's character and can be expressed mathematically as a series of frequencies, which are multiples of the fundamental frequency. The formula representing this overtone series is:

$$f_n = nc \cdot f_0 \tag{1}$$

where f_n is the frequency of the nth overtone and f_0 is the fundamental frequency.

Furthermore, it is crucial to recognize that the human perception of musical notes is logarithmic rather than linear. Thus, the linear band splits proposed by the original author may not effectively capture the perceived harmony within music. To devise a set of generic band splits that account for human auditory perception, we propose leveraging the twelve-tone equal temperament (12-TET) system, which is the foundation of most contemporary music. This system divides an octave into 12 equal parts, where each semitone interval is the twelfth root of two apart in frequency. [4]

Building upon the 12-TET system, we can define band splits that delve into the microtonal realm, specifically the 24-TET system, which further splits each semitone into two. Placing the center of a fundamental frequency within a microtonal band allows us to align with the perceptual characteristics of human hearing more closely. This microtonal approach is particularly relevant as trained musicians and listeners can discern pitch variations as small as 10 cents on average. By employing microtonal band splits, our refined BSRNN can better accommodate the harmonic nuances of music, facilitating a more accurate and harmonically-aware source separation, even within the 12-TET framework.

To implement the evenly distributed microtonal band-splitting method, we employ an algorithm that calculates the frequencies for microtonal notes within the range of human hearing. This algorithm is based on a division of each octave into an equal number of microtones, as defined by the `divisions` parameter. Here's the implementation in python:

```
def microtonal_notes(divisions=24):
    A4_freq = 440.0
    min_freq = 20.0
    max_freq = 20000.0
    freqs = []
    current_freq = min_freq
    while current_freq <= max_freq:
       rel_sem = 12 * np.log2(current_freq / A4_freq)
        near_micro = round(rel_sem * divisions / 12)
        nearest_freq = A4_freq \
            * (2 ** (near_micro / divisions))

        if near_micro % (divisions // 12) != 0:
            freqs.append(nearest_freq)

        current_freq = A4_freq \
            * (2 ** ((near_micro + 1) / divisions))
    return freqs
```

Using this code, one can create a sequence of microtonal notes that are evenly distributed within the audible range. Applying this algorithm, 5 illustrates that the splits are more uniformly distributed across the musical note spectrum.

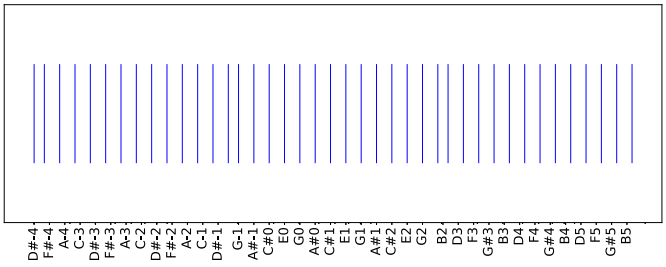

Fig5: Gneric Band Split schema in log scale

B. *Multi-task learning*

In the context of a universal architecture for Music Source Separation, simply having a versatile structure is insufficient. It's equally important to have an effective training methodology that can fully exploit this architecture's capabilities. Multi-task learning offers a powerful approach to achieve this goal. By training the model on multiple source separation tasks concurrently, the model learns to generate several masks for different sources simultaneously.

The strength of MTL lies in its ability to integrate learning from various tasks, each contributing to a more comprehensive understanding within the model. For our refined BSRNN, we implement MTL by defining a cumulative loss function that aggregates the individual losses from each source separation task. These losses are calculated using the original loss function as defined in the BSRNN paper, which is applied to each task separately.

The cumulative loss function for the MTL framework, L_{MTL}, which is used to train the universal feature extractor, is defined as the sum of individual task losses. For each source

separation task j, we calculate a loss function L^j_{obj}. The loss function is the as the original paper. The MTL loss is then:

$$L_{\text{MTL}} = \sum_j \left(L^j_{\text{obj}}\right) \tag{2}$$

Fig6: Training pipeline without MTL

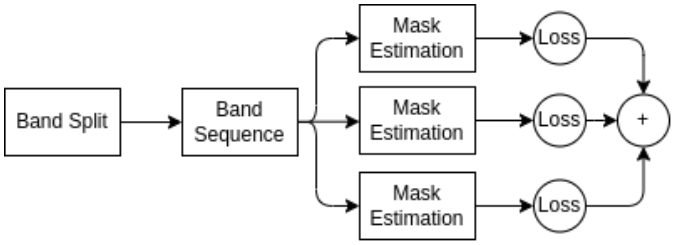

Fig7: MTL train pipeline

Upon completing the training process, the Mask Estimations Layer can be detached and replaced by a singular Mask Estimation layer. The Band Split and Band Sequence modules are then set to a non-trainable state, and transfer learning is applied to the model for a new source separation task.

IV. Experiment Configuration

Because of the hardware limitations and time constraints associated with this research, a strategic reduction in the model's complexity was necessary. We scaled down the number of trainable parameters from approximately 500 million to a more manageable 4 million. This significant decrease was implemented to facilitate a proof of concept and to expedite the initial drafting process.

We acknowledge that such a drastic reduction in parameters may potentially impact the model's performance. As part of our future work, we intend to conduct experiments using the hyperparameters recommended in the original paper to fully realize the research. The precise parameters utilized for these initial experiments, including the reduced model configuration, are documented in the version history of the accompanying Git repository.

V. Results And Analysis

This part of research is incomplete. especially the MTL part

Our initial experiment focused on evaluating the effectiveness of the generic frequency band splits proposed in Section 3. We trained a baseline model on the task of separating drum sounds from a source mixture. To facilitate a comparative analysis, we utilized both the generic splits and the v7 schema outlined by the original authors, conducting training for approximately 150 epochs for each.

Following the baseline training, we implemented transfer learning to adapt these models from drum separation to bass separation tasks. The dataset employed for both training and transfer learning was Musdb18hq, consistent with the dataset used in the original paper.

The outcome of this experiment is visually represented in 8, which displays the Universal Source-to-Distortion Ratio (USDR) scores. The results indicate a noticeable improvement in USDR for the model utilizing the generic split configuration as opposed to the v7 schema. This enhancement in performance underscores the potential of the proposed generic splits in achieving more effective music source separation.

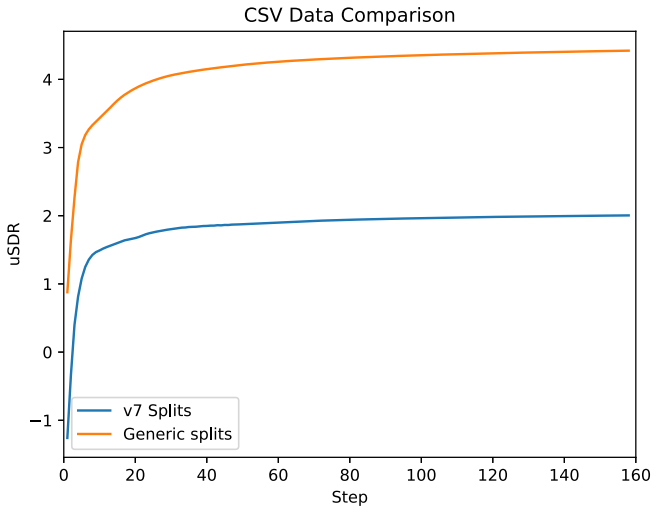

Fig8: Comparison of uSDR transfer learning from drums to bass

VI. Conclusion And Future Directions

In our study, we've seen that smartly applying knowledge from music experts can make a big difference in how well our models work. As we move forward, we should look into two key questions:

1. How big does the model need to be to handle music separation tasks, and is there a difference in size requirements when we use the generic band split approach?
2. Which set of instruments should we focus on separating in our training to develop a feature extractor that works well for any music?

For anyone interested in diving deeper or contributing, all the code and details of our work are available on GitHub at: https://github.com/sahandevs/BandSplit-RNN.

References

[1] Y. Luo and J. Yu, "Music Source Separation With Band-Split RNN", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, no., pp. 1893–1901, 2023, doi: 10.1109/TASLP.2023.3271145.

[2] R. Caruana, "Multitask Learning", *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul. 1997, doi: 10.1023/A:1007379606734.

[3] K. Li, X. Hu, and Y. Luo, "On the Use of Deep Mask Estimation Module for Neural Source Separation Systems", *arXiv.org*, Jun. 2022, [Online]. Available: https://arxiv.org/abs/2206.07347v1

[4] M. Müller, in *Fundamentals of Music Processing: Using Python and Jupyter Notebooks*, 2nd ed., Springer, p. 21.