Monaural Source Separation in the Wild

Tianjun Ma

CMU-CS-20-109

May 2020

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Louis-Philippe Morency, Chair
Bhiksha Raj

Submitted in partial fulfillment of the requirements
for the degree of Master of Science.

Keywords: machine learning, audio signal processing, monaural source separation, source separation dataset, deep neural network, multi-headed self-attention

Abstract

Monaural source separation refers to the process of extracting individual components from a mixture, where the mixture is a single-channel audio recording of multiple sources emitting sounds simultaneously, and the individual components are the constituent sounds emitted by each source. In recent years, data-driven approaches using deep neural network-based models for monaural source separation have been shown to outperform their non-data-driven counterparts. However, these approaches are designed using specialized datasets in which the sources belong to a constrained set of categories and the mixtures are not very representative of audio mixtures in the real world. Consequently, whether existing models could generalize to more complex source separation settings is open to questions. In this work, we want study and formalize the notion of monaural source separation in *real-world scenarios* and explore model designs that adapt to such complex settings. Specifically, we present the *Wild-Mix Dataset*, a synthetic dataset in which mixtures consist of sources belonging to a variety of sound categories and are synthesized in dynamic ways. We also present *ASTNet*, the first supervised learning model to utilize multi-headed attention to tackle monaural source separation. We show that the *Wild-Mix Dataset* is a challenging benchmark for evaluating model performance in complex *real-world scenarios* and that *ASTNet* achieves the state-of-the-art performance on the *Wild-Mix Dataset*.

Acknowledgments

I would like to thank my advisor, Louis-Philippe Morency, for guiding the overall direction of my research and patiently resolving my concerns. I would like to thank my co-advisor, Bhiksha Raj, for giving me invaluable suggestions and offering me insightful critiques. I would also like to express gratitude toward my mentor, Amir Zadeh, for having countless exciting research discussions with me and helping me overcome challenging obstacles along the way. I want to thank all of them for their whole-hearted support during my graduate studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

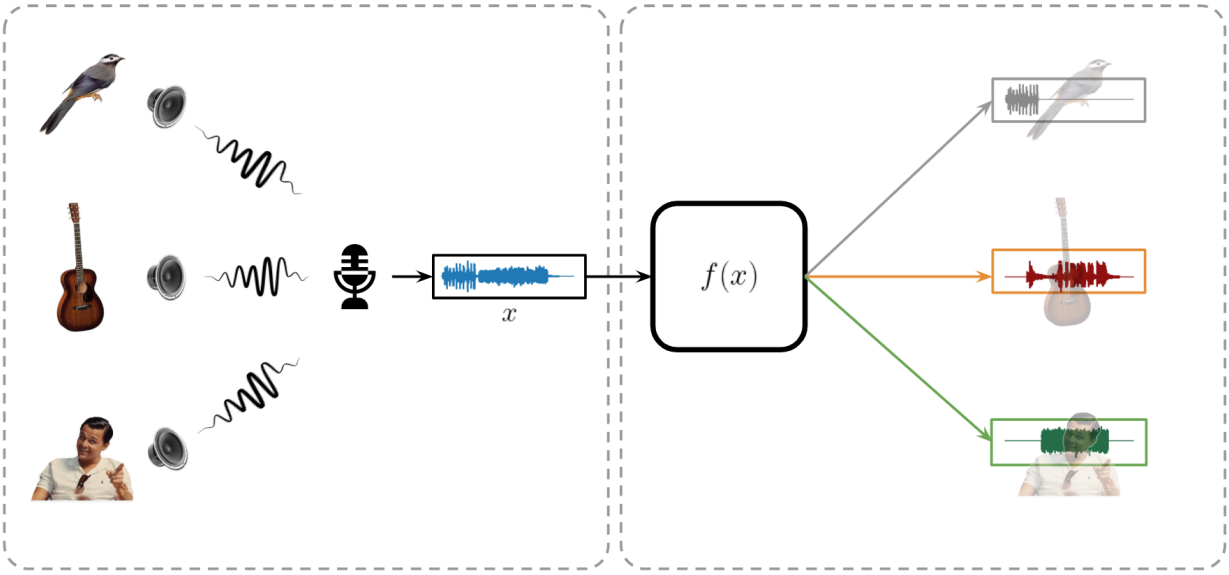


Figure 1.1: The monaural source separation process

1.1 Overview

Audio source separation has been a fundamental challenge for intelligent systems. Monaural source separation is defined where only a single-channel audio recording of multiple sources emitting audio signals simultaneously is available (Figure 1.1 (a)) and the goal is to extract constituent audio signals emitted by each individual source(Figure 1.1 (b)). Given that single-channel mixtures reveal extremely limited information about how constituent source signals are combined together, monaural audio source separation presents one of the most challenging scenarios of audio source separation.

There has been significant progress on monaural source separation approaches. In recent years, data-driven, deep neural network-based models has been shown to outperform traditional models that are not data-driven[13, 14, 31, 32]. However, to our knowledge, existing data-driven models are developed to tackle special cases of monaural source separation, since such models are trained using datasets tailored for specialized separation problems such as speech enhancement, speaker separation, and music instrument separation. These problems assume that the sources come from limited number of categories. Additionally, most existing datasets for source separation make many assumptions about mixture composition. For example, these datasets assume that source signals always appear from the beginning and last until the end of the mixture without any breaks and at most two sources of interest are present in the mixture.

In order to explore how audio source separation fundamentally works, we believe that source separation models should not only focus on these special cases, but also explore source separation in *real-world scenarios*, where audio mixtures encompass a more comprehensive set of sources beyond speech and music, and could be formed in more diverse ways. To this end, we want to formalize the notion of such *real-world scenarios* by defining a set of standards and crafting a comprehensive dataset featuring the *real-world scenarios*. Since existing models are designed for the aforementioned special cases of source separation, we want to evaluate their generalizability to the *real-world scenarios* by training them using the dataset we propose. Furthermore, we want to develop novel models that could better adapt to the increased complexity than existing models.

1.2 Contributions

Our research is therefore two-fold, we have made contributions to both dataset creation and novel model design. Specifically, our contributions are:

- Formal definition of *real-world scenarios* for monaural source separation. We establish a set of standards for emulating real world auditory scenes.
- *Wild-Mix Datasets*, which is a collection of synthetically created datasets featuring *real-world scenarios* with diverse configurations. We use them to train and evaluate supervised models.
- The library for synthesizing the *Wild-Mix Datasets*. We walk through its API and show the example usage. We also introduce how its versatility could help researchers to create arbitrarily many datasets with different configurations that could be utilized by existing or potential future research topics on monaural source separation.
- *Attentive Spatio-Temporal Network (ASTNet)*, a neural network-based model that combines recurrent units, dilated convolution, and multi-headed self-attention. It achieves the state-of-the-art performance on the *Wild-Mix Datasets*.
- Thorough evaluation of competitive monaural source separation models and *ASTNet* on both the *Wild-Mix Datasets* and TIMIT[9] speech separation datasets (speech enhancement and speaker separation). We offer an ablation study of how each components of our

model contribute to its overall performance and present our observations on the robustness of different source separation models against variations in dataset complexity.

- We make the datasets and related implementations public[1][2].

1.3 Thesis Organization

The thesis document is organized as follows: In Section 2, we introduce related work inclduing important datasets and models that have been proposed for monaural source separation. In Section 3, we define *real-world scenarios* in more details, introduce the data collection process and the data synthesis library for creating the *Wild-Mix Datasets*. In Section 4, we present *ASTNet* and the intuition behind its design, presents the experiments setup, and analyze the experiment result. Finally, in Section 5, we summarize the contributions of this thesis and discuss future directions.

[1]Wild-Mix Datasets and synthesis library: `https://github.com/A2Zadeh/WildMixDataset`
[2]ASTNet+experiment pipeline: `https://github.com/tianjunm/monaural-source-separation`

Chapter 2

Related Works

2.1 Datasets

2.1.1 Source Separation Datasets

There are many existing highly-quality datasets designed for training monaural audio source separation models. Existing source separation datasets are specialized for certain tasks including speech and music instrument separation. Examples of the most widely used datasets are:

CHiME-2 WSJ0 Dataset

The dataset is published for the 2nd 'CHiME' Speech Separation and Recognition Challenge [30]. It contains around 166 hours of English speech in a noisy room environment. The clean speeches are based on CSR-I (WSJ0) Complete[15], where sentences (5,000 word in total) come from Wall Street Journal texts. This dataset creates a suitable setting for speech separation tasks as clean speeches are accompanied by background noises from room environments, and it also provides noises outsided of the training set for evaluation purposes. The noisy speeches are synthesized using clean speeches and background noises and all audio data are recorded as 16-bit, 16kHz waveforms [30].

TIMIT Acoustic-Phonetic Continuous Speech Corpus

TIMIT Speech Corpus provides speech data originally designed for speech recognition systems, nevertheless, researchers find it a high-quality dataset for training speech separation models as well, as mixed speeches could be synthesized using the clean speechs within the TIMIT corpus[13, 14]. It contains 630 speakers speaking 8 different dialects of American English, and each speaker records 10 phonetically rich sentences. The speeches are recorded as 16-bit, 16kHz waveforms[9].

AVSpeech

A large-scale audio-visual speech dataset developed by Google[5]. It contains 4700 hours of speech videos spanning a variety of people, languages, and face poses. It is developed for build-

ing multi-modal speech separation algorithms, but the audio-only portion of the dataset also provides valuable resource for monaural speech separation.

MUSDB18

A high-quality dataset for music instrument separation. It is the first dataset targeted specifically for multi-source separation where more than 2 sources of interest are present in the mixtures. It contains 150 full-length music tracks of different genres, along with their corresponding drums, bass, vocals and others as ground truths[25].

2.1.2 Emulations of Source Separation in the Real World

Researchers have also devised many data augmentation techniques beyond existing datasets to introduce more diversity and complexity to the datasets. Their works have also inspired our formulation of the *real-world scenarios*. However, even though they have made existing datasets more representative of realistic settings, these data augmentation techniques are only applied to speech and music datasets. We will discuss how we extend *real-world scenarios* beyond these fixed categories in more details in Section 3.

Uhlich Data Augmentation

Uhlich et al. have proposed a data augmentation technique using music instrument separation datasets in [27]. Specifically, the authors have introduced the following manipulation of the DSD100 Dataset (subset of MUSDB18):

- Random swapping left and right channel for each instrument.
- Random manipulation of the volumes of source signals.
- Random chunking of source signals.
- Random mixing of instruments from different songs.

These manipulations break the originally rigid and static ways mixtures are formed by source signals and establish the stepping stone for more complex manipulations.

Generalized Speech Enhancement

Pascual et al. also argue that speech datasets should be more generalized in they way that mixtures are formed. Specifically, beyond what Uhlich et al. have proposed in their work, the authors also present some additional data augmentation techniques:

- Clipping of speech data to create different levels of distortion.
- Random re-sampling of speech waveforms.
- Synthetically generating whispered speeches.

These techniques further force speech separation algorithms to adapt to more challenging scenarios given the additional distortions and missing information from the input data. Even though these techniques are specialized to speech data, we believe they provide valuable insight to the design of the *real-world scenarios* as well.

2.2 Models

2.2.1 Before the Pervasion of Data-Driven Approaches

The performance of recent data-driven approaches using deep neural network-based models have substantially exceeded that of earlier non-data driven models designed for monaural source separation [13, 14, 20, 31, 32]. Nevertheless, we want to acknowledge some influential works before deep learning became more accessible and prevalent since they have provided a significant amount of insight and guidance on the design of subsequent deep neural network-based approaches.

Gaussian Mixture Model

Gaussian scaled mixture model (GSMM) is a statistical model proposed by Benaroya et al. for monaural audio source separation. They present the idea of using adaptive Wiener filtering during the derivation of maximum a posteriori and posterior mean estimates of the sources[1], and their work has influenced both feature engineering techniques for audio mixtures [18] and the design of supervised learning models utilizing adaptive filtering[16].

Non-negative Matrix Factorization

An influential unsupervised technique using non-negative matrix factorization (NMF) is proposed by Tuomas Virtanen along with a cost function that favors temporal continuity and sparseness of source signals and has demonstrated the efficacy of finding the low-rank representations of reference sources with predefined constraints. The application of NMF to monaural source separation not only provides mathematical interpretability to the separation process but has also been incorporated in many subsequent works using deep neural networks [8, 17, 20].

2.2.2 Multilayer Perceptron (MLP)

With the availability of large-scale datasets and increasing computational power, researchers are able to extensively study application of deep learning on monaural audio source separation. In this section, we introduce some representative works that incorporate deep multilayer perceptron, the most basic and essential type of deep neural network, into their model design.

DNN

Wang et al. proposed DNN-CRF, one of the first works that uses deep learning and shows the superior performance of deep neural networks in monaural source separation compared to statistical or non-data-driven models[32]. Their work argue that acoustic features are intrinsically not linearly separable, so DNNs, which models non-linearity, are ideal candidates for audio signal processing.

Autoencoder Based Source Separation (AESS)

[20] Osako et al. take inspiration from NMF models and incorporate deep multilayer perceptrons in the design of AESS, an autoencoder which models a dictionary that encodes target sources with higher expressiveness than NMF's low-rank representations. By effectively combining the idea of source representation and deep neural networks that model non-linearity, their autoencoder takes the best from both worlds and is shown to substantially outperform NMF [20].

2.2.3 Convolutional Neural Networks (CNN)

The time-frequency representation of audio data prompts researchers to exploit the rich visual information within spectrograms by adopting convolutional neural networks.

Redundant Convolutional Encoder Decoder (R-CED)

Park et al. propose R-CED, a fully convolutional autoencoder-based model for monaural source separation, and besides that it could model speech enhancement performance better than DNNs, it requires much less parameters to reach superior performance[22]. Many variations of such convolutional autoencoder-based models have been studied thereafter given their effectiveness in modeling audio data with time-frequency representations[12, 21].

2.2.4 Recurrent Neural Network (RNN)

Researchers have also noticed that DNNs are sub-optimal candidates for modeling audio data in that DNNs focus on segments of acoustic features without capturing much temporal dependency within audio streams [31]. Models with recurrent structures thus come to rescue as they are known for their ability to build connections among timesteps.

Deep Recurrent Neural Network (DRNN)

Huang et al. offers systematic study of the application of RNNs to monaural source separation. They thoroughly compare DNNs with different variations of RNNs including DRNN, a deep feed-forward neural network with one intermediate layer of recurrent connection, and stacked RNN, which is a deep neural network with stacked recurrent layers. It is shown that stacked RNNs are able to capture temporal dependency of audio streams and outperform both NMF and DNN-based models on speech enhancement and speaker separation.[14]

Long Short-time Memory (LSTM)

Even though RNNs are effective at capturing temporal dependency, they suffer from the well-known vanishing and exploding gradient problem during back propagation through time [2]. This is especially problematic for source separation as input audio could be arbitrarily long. Therefore, LSTM becomes a more suitable recurrent structure for modeling audio as there is a memory component to it that enables it to capture long-term temporal depency. The superiority

of LSTM-based models are illustrated by Chen et al. in their study of the effectiveness of stacked LSTM layers for speaker separation[3].

2.2.5 Combination of CNN and RNN

Both CNN and RNN-based models are shown to be very effective, but which one is intrinsically more suitable for monaural source separation is still an open question. More recently, researchers have also tried to blend two fundamental designs in model development.

EHNet

Zhao et al. argue that better models should capture both temporal and spectral information from the audio input. Therefore, they propose EHNet, a purely data-driven deep neural network-based model for speech enhancement. EHNet contains three components: a CNN-based component that exploits spatio-temporal information from spectrograms, a LSTM-based component that models complex long-term temporal dependency within the audio input, and a fully-connected component for generating the spectrograms of clean speech[35]. A similar model which uses dilated convolutional layers in the CNN-based component is used as the audio-only component in Google's multimodal audio source separation model [5] and is shown to achieve state-of-the-art performance in audio-only speech separation tasks as well. Such combination of RNN and CNN-based structures have also greatly inspired our model design, which we will explicate in Section 4.

Chapter 3

Dataset Construction

3.1 Introduction

Data-driven models for monaural source separtion have demonstrated substantial performance improvement in comparison to traditional statistical or unsupervised approaches. However, the scope of monaural source separation has always been limited to certain special cases such as speech enhancement, speaker separation, and music instrument separation. Given the potential of more powerful models, wouldn't a more generalized monaural source separation scenario be an interesting problem to investigate as well? We denote the generalized scenario we are referring to as *real-world scenarios*. Intuitively, such scenarios encompass all types of sound we hear in our day-to-day life (*e.g.* birds chirping, cars passing by, and keys jangling), beyond specific ones such as speech and musical instruments.

Existing datasets do not reflect the complexity of the *real-world scenarios*, as the sources are always from a fixed set of categories, namely human speech with specific types of background noise or a handful of musical instruments. The mixtures within the existing datasets are also synthesized in the same way: there are always a fixed number of sources present in the mixture, the source signals are recorded using the same microphone setting, and they have the same duration as the mixture. Even though here have been some works that design data augmentation methods to synthesize mixtures in more dynamic manners, the resulting datasets still feature the special cases mentioned previously.

Therefore, in order to investigate monaural source separation in *real-world scenarios*, we need to formalize this idea and create the corresponding datasets. To do that, we first present the notion of *real-world scenarios*, which defines a set of standards that we want a synthetic dataset to meet so that it could be be considered an emulation of the real world. Then, we carry out the dataset creation process to meet those standards. Specifically, we have adopted the AudioSet ontology and manually selected a representative subset of data within AudioSet, and have created a versatile synthesis library to help us build the *Wild-Mix Datasets*, which is a collection of synthetic datasets featuring the *real-world scenarios*.

For the rest of this chapter, we will begin by explicating *real-world scenarios* in Section 3.2. Following the definition, we introduce the steps we have taken to create datasets that meet the standards of *real-world scenarios*, including the audio data collection which will be detailed in

Section 3.3 and the dataset synthesis library which will be introduced in Section 3.4. Finally, we will showcase the result datasets in Section 3.5.

3.2 Real-World Scenarios

So, what are the *real-world scenarios* that we've been mentioning from the beginning? They are a set of desired properties that we want a monaural audio source separation dataset to have to be representative of complex auditory scenes in the real world.

Girin et al. have proposed what real-world scenarios should look like for multi-channel source separation. Specifically, they argue that datasets reflecting real-world scenarios should be created in an into-the-wild fashion, where for each audio mixture, sources and microphones are not always physically static, the number of sources could be varying, the sources could be spatially diffuse, and microphone arrays should not be standardized[11]. Even though these specifications are designed for multi-channel source separation, beside the requirement for more arbitrary settings of microphone arrays, the other three are directly applicable to the monaural case.

Drawing inspirations from both the into-the-wild formulation and the data augmentation techniques introduced in Chapter 2 [23, 27], we propose the standards that monaural source separation datasets should meet in order to feature *real-world scenarios*: diversified source signal acquisition, arbitrary mixture composition, and arbitrary mixture heterogeneity. We will discuss each of them in detail from Section 3.2.1 to Section 3.2.3.

3.2.1 Diversified Source Signal Acquisition

As we have mentioned previously, the existing datasets for special cases of monaural audio source separation impose many assumptions on how source signals are recorded. We aim to remove as many of these assumptions as possible by:

- defining a larger ontology for audio data, that is, ensuring that sources come from a much more diverse set of categories beyond speech and musical instruments,
- encouraging variegated microphone settings including the intrinsic parameters of recording device and the physical setup of recording environment
- encouraging variations in the volumes of source signals

3.2.2 Arbitrary Mixture Composition

Having mixtures formed in more complex ways is also crucial in emulating real world auditory scenes. Arbitrarity in the composition of mixtures are enforced by making the following parameters variable:

- source count: there could be more than two sources present in audio mixtures and the number of active sources within different segments of mixtures could also vary
- source appearance: in the audio mixtures, source signals should not always last from the beginning to the end but could also occur at random intervals within the mixture

- source segmentation: source signals could be cut into multiple chunks with adjustable levels of granularity and mixtures could contain any one or more chunks of the segmented source signal

3.2.3 Arbitrary Mixture Heterogeneity

Different levels of heterogeneity of source signals within mixtures is enforced by making the following parameters configurable:

- source selection: within the same dataset, different mixtures could contain sources from different categories
- category scope: the scope of all possible source categories should be adjustable
- mix method: within every audio mixture, the constituent sources can all come from distinct categories, identical categories, or a hybrid of these two cases

Given the practical difficulties of recording large-scale audio mixtures and their constituent sources manually, we decide to build synthetic datasets that meet these standards. Diversified source signal acquisition imposes requirements on the diversity of individual sources and their recording process. Arbitrary mixture composition and heterogeneity, on the other hand, emphasizes the versatility of the synthesis process. Therefore, we design and carry out an audio data collection process to tackle diversified source signal acquisition and we build a library for dynamic mixture synthesis to make sure that the resulting datasets embrace arbitrary mixture composition and heterogeneity. As a result, we are able to create datasets featuring *real-world scenarios* with the properties illustrated in Table 3.1.

	Uhlich et al.	SEGAN	Wild-Mix
Ontology	musical instruments	speech	diverse
Variegated microphone setting			✓
Variation in source volume	✓		✓
Adjustable source count			✓
Adjustable source appearance	✓		✓
Adjustable source segmentation	✓	✓	✓
Configurable source selection	✓		✓
Configurable category scope			✓
Configurable mix method			✓
Mixed sample rate		✓	
Synthetic sound		✓	
Publicly available?			✓

Table 3.1: Comparison of properties of different synthetic datasets.

3.3 Audio Data Collection

In order to make sure that the acquisition of source signals is indeed diversified, we have to define a comprehensive ontology, record audio data using different types of equipment, set

up numerous different recording environments, and make sure that audio signals are recorded with different volumes. Theses requirements makes Google's AudioSet a great choice for us to obtain the source signals. AudioSet defines a comprehensive ontology for real world audio events and offers large-scale audio data annotated from Youtube videos[10]. Therefore, AudioSet encompass the diversity requirement for both the categories of source signals and their recording process.

There is one caveat in directly using AudioSet as the basis for synthesis: even though more variations is favored during the source signal recording process, too much uncertainty would compromise our knowledge or the quality of the source signals and thus affect the synthetic dataset. To counter this, our data collection involves a verification process that filters out data that we consider problematic from the AudioSet. We will start by introducing the AudioSet in Section 3.3.1, and then describe the dataset verification process in Section 3.3.2.

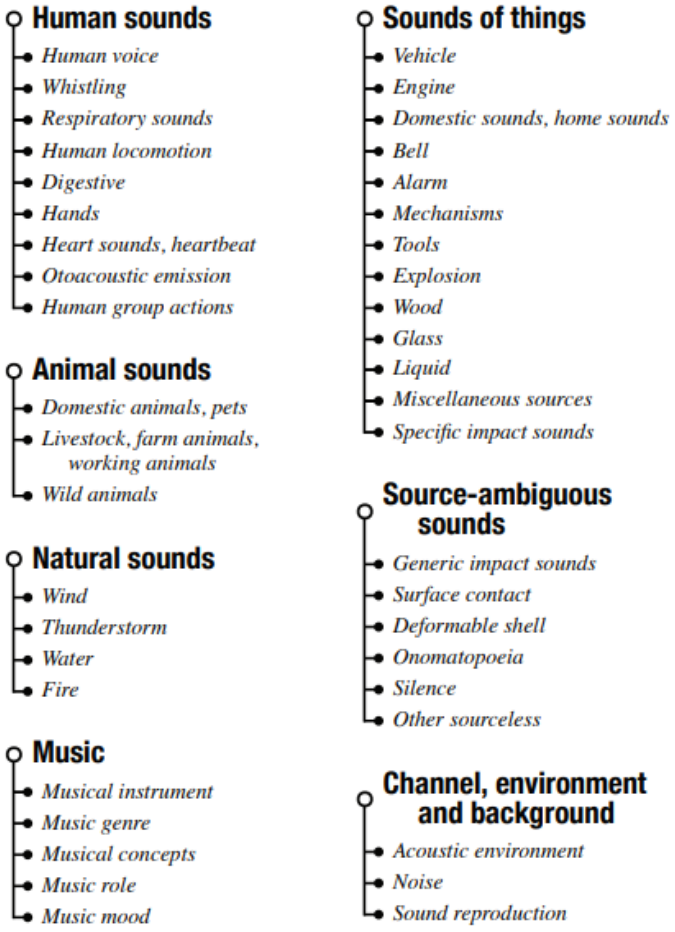


Figure 3.1: The AudioSet ontology.

3.3.1 AudioSet Ontology

We found Google's AudioSet (Figure 3.1, source: [10]) to have the largest ontology available online. AudioSet also saves us a lot of work in defining an ontology and obtaining diversified audio data.

The AudioSet establishes an audio ontology with 527 source categories, and each category contains hundreds of machine-annotated audio clips selected from YouTube videos, totaling 2.1 million videos with 5.8 thousand hours of audio [10]. The scale of AudioSet and the fact that audio data come from Youtube videos which are recorded using different microphone settings in a variety of environments ensure that the source signals are acquired in a diverse fashion and could serve as a solid basis for creating synthetic datasets featuring *real-world scenarios*.

3.3.2 Data Verification

We have to put extra care, though, on the verification process, in that the intrinsic diversity of audio data from AudioSet introduces potential ambiguity in data labeling. There are mainly two common problems that require further attention and we have to make sure the source signals we use for synthetic dataset creation are free of those problems.

First of all, AudioSet contains audio data labeled with more than one categories. Mixtures in our datasets should be synthetically created, so that we have perfect knowledge about the categories of the source signals and the ground truth information for source separation algorithms. Therefore, it is not desirable for source signals to be mixtures themselves, and it is thus necessary to avoid using such data as the basis for dataset synthesis. What's more, there are also a significant amount of data with single label but actually contain multiple sources. For example, an audio clip with label "barking" might contain background noise such as people talking. Therefore, we should not let our guard down during the verification process even when source clips have single labels.

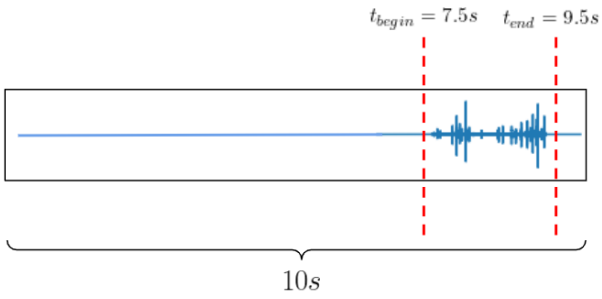


Figure 3.2: AudioSet data re-annotation

Second, data with only one source might also contain a considerable amount of silence. This type of data should be discarded or re-annotated since they are not characteristic of whatever its label suggests. For example, a 10-second audio clip labeled with bird chirping with 8 seconds of silence should be re-annotated, as illustrated in Figure 3.2. What's more, another reason for this extra step of precaution is that lengthy silence also creates complications for the adjustable

source segmentation property introduced in Section 3.2.2, in that randomized segmentation of such data will very likely capture silence instead of the sound event suggested by its label.

Therefore, we have stipulated the following rules for conducting a manual verification process of data from AudioSet, before we synthesize the final datasets:

- Discard audio data with multiple labels.
- If an audio clip has a single label, but contains unlabelled sources, either discard it or re-annotate the clip with the start and end timestep (at least 1 second) where only the labeled source is present.
- If an audio clip is predominated by silence, annotate the start and end timestep (at least 1 second) so that the annotated clip does not contain an unacceptable amount of contiguous silence, which is usually contained within half a second.

Given the amount of manual work to perform the verification on the entire AudioSet, we only select a subset of sound categories that we think are most representative of the real world and performed verification on one hundred audio clips for each category. The verification process is strictly conducted by a well-trained group of students at Carnegie Mellon University. We have around 60 categories annotated, and selected 30 categories from them for experimentation purposes (Table 3.2). Within each category of the verified AudioSet, 80% of data are reserved for training, 10% are reserved for validation, and 10% are reserved for testing.

Firetruck Siren	Violin	Male Speech
Church Bell	Flute	Child Speech
Telephone Bell Ring	Acoustic Guitar	Applause
Keys Jangling	Piano	Duck
Typing	Female Singing	Bark
Writing	Trumpet	Bird
Shaver	Sxaophone	Engine
Vacuum Cleaner	Snare Drum	Water
Chainsaw	Tambourine	Wind
Fireworks	Baby Laughter	Knock

Table 3.2: 30 categories selected for synthetic dataset creation.

3.4 Dataset Synthesis

Given that it's infeasible create datasets featuring *real-world scenarios* by manually recording all the mixtures along with their constituent sources, dataset synthesis is the most cost-effective alternative. Therefore, we build a versatile dataset synthesis library and makes it easy to create datasets with arbitrary mixture composition and heterogeneity described in Section 3.2.2 and 3.2.3. We first walk through the APIs of the library to show how it facilitates the process of creating qualified source spearation datasets featuring *real-world scenarios*. In the end, we then demonstrate the *Wild-Mix Datasets*, a collection of synthetic datasets based on the verified data from AudioSet.

3.4.1 Library APIs

We have defined a set of standards that datasets need to meet in order to be considered representative of the *real-world scenarios*. Diversified source signal acquisition is mostly taken care of by the AudioSet besides arbitrary volume adjustment, and it is the data synthesis process that ensures the arbitrarity in mixture composition and heterogeneity of the resulting synthetic dataset.

Arbitrary mixture composition specifies that there should be variations in source count, the timestep that a particular source signal appear in the mixture, and the way source signals are segmented. Arbitrary heterogeneity requires the synthesis process to allow the selection of the categories of sources for each mixture, make it possible to define what categories could be considered, and alter between different mix methods. As a result, our library allows the following parameters to be customized with different levels of granularities to synthesize datasets:

- **source volume**:
  - `"original"`, source signals will be taken unmodified
  - `"normalized"`, all the source signals will be normalized to the same loudness levels according to the EBU R 128 recommendations
  - `"random"`, all the source volumes are randomized after they are normalized to same loudness levels according to the EBU R 128 recommendations
- **source count**:
  - `count`, an integer value that specifies the number of sources each mixture contains
  - `(lo, hi)`, an integer tuple representing the minimum and maximum number of sources to appear in the mixtures
- **source appearance**:
  - `(start_low, start_high)`, an integer tuple representing the lowerbound and upperbound on the timestamp at which each source signal could appear within the mixture
  - if `"random"` is given, the source signals will appear at random positions within the mixture
- **source segmentation**:
  - `(start_low, start_high, duration_low, duration_high)`, an integter tuple representing the lowerbound on the segment starting timestamp, the upperbound of the segment starting timestamp, the lowerbound on the segment duration, and the upperbound on the segment duration, respectively
  - if `"random"` is given, segments will be taken randomly from the source signals
- **source selection**:
  - `[id_1, id_2, ...]` a list of integers specifying the IDs of audio data within each category that the mixtures could choose from (currently, there are 100 source clip within each verified category, and a the number of clips reserved for train, validation, or test set is pre-defined).

    - if `"random"` is given, the source clip is selected randomly from all clips within the category (with the constraint of train, validation, and test set specification)
- **category scope**:
    - `filename` is the path to a JSON file listing the possible categories that a dataset could sample source clips from.
    - `num_categories`, an integer representing the desired number of categories that a dataset sample source clips from. If the number is given instead of file path, the dataset will randomly select that many categories from all possible categories available in the verified AudioSet.
- **mix method**:
    - `"interclass"`, each source come from a different category
    - `"intraclass"`, all the sources come from the same category
    - `"hybrid"`, there could be zero or more sources that come from the same category
- **mixture duration**: `duration`, an integer representing the duration of mixtures in seconds. Currently, we only support fixed
- **dataset size**: `(train_size, val_size, test_size)`, a integer tuple representing the size of training data, validation data, and test data in the generated dataset. With pre-defined train/validation/test split of the verified AudioSet data, these sizes could be arbitrarily large given that we are creating a synthetic dataset.

3.4.2 Example Usage

This is an example of a particular use case of the API. It is used to create part of our datasets for experimentation.

```
python datagen.py
    --source_volume original \
    --source_count 2 \
    --source_appearance random \
    --source_segmentation (0, 4, 2, 4) \
    --source_selection random \
    --category_scope 10category.json \
    --mix_method interclass \
    --mixture_duration 4 \
    --dataset_size 20000
```

Using the specification above, the following process (Figure 3.3) is probably happening under the hood: one 4-second mixture is created by taking two random segments from two source signals and each segment appear randomly within the mixture. The part where they overlap will be added together.

The generated datasets are stored as comma-separated values (CSV). Each row of the dataset file contains the following information:

- filename: the filename to identify a particular audio file from the verified AudioSet

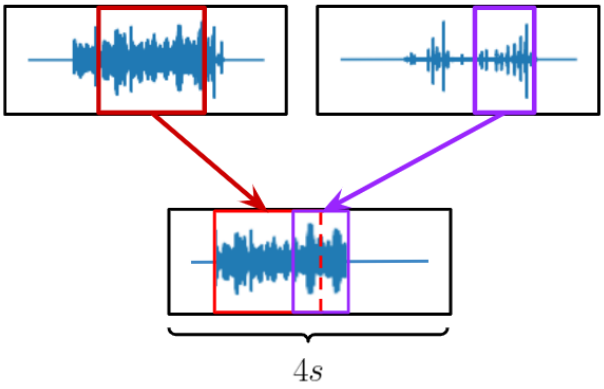


Figure 3.3: An example of the data synthesis process.

- source start: the starting timestamp of the segment within the source clip
- source end: the ending timestamp of the segment within the source clip
- source signal duration: the duration of the source clip segment
- mixture timestamp: the timestamp from which the segment starts within the mixture
- volume multiplication factor: a number between 0 and 1 that gets multiplied with the source clip segment before it is added to the mixture
- category: the category of the source, used for bookkeeping

During training, the CSV file is loaded by a PyTorch and the mixtures are created on-the-fly. This design enables fast modifications of datasets, saves tremendous amount of storage space, and is shown to be faster than loading data from the disk when the mixtures are pre-generated and are stored on disks.

3.5 Wild-Mix Datasets and Beyond

We have created a comprehensive set of datasets for the experiments. We first present the *Wild-Mix Datasets*, a collection of datasets created using the verified AudioSet data and different synthesis configurations. For additional studies, we have also used the synthesis library to create TIMIT-based speech separation datasets featuring speaker separation and speech denoising scenarios.

3.5.1 Wild-Mix Datasets

Wild-Mix Datasets are created using different synthesis configurations. These datasets collectively form a comprehensive benchmark for evaluating models under the *real-world scenarios*.

Specifically, we decided to first fix our scope of source categories to a pre-defined 10-category subset of the 30 categories we selected from the verified AudioSet for experiment purposes. Even though creating datasets using larger category scopes is feasible, we selected 10 to preserve the complexity while maintaining the quality of separation result across models. We also

fixed the source volume parameter to be `original`, as variance of source volumes is already high in the verified AudioSet. We create datasets with 2, 3, 5 sources, segmented and placed within mixtures randomly. For each different source count (2, 3, or 5), we create three datasets that use inter-class, hybrid, and intra-class as mix methods. For additional experiments, we only created one additonal dataset using 5 and 30 as the category scope, respectively. For these two additional datasets, we fixed the source count to be 2 and mix method to be inter-class. All of the datasets contain a training set with size 20k and a validation set with size 2k. In total, *Wild-Mix Datasets* encompass 11 configurations in total. *Wild-Mix Datasets* meet the standards of *real-world scenarios* given the variety offered by these 11 configurations.

3.5.2 Speech Datasets

We believe incorporating classic scenarios into our experiments will offer more fairness to the experiments as we are not only using our own datasets for evaluating model designs. We chose speaker separation and speech denoising since they are two of the most frequently studied problems in the source separation community.

The speech data come from the TIMIT Acoustic-Phonetic Continuous Speech Corpus [9]. Since the TIMIT corpus only contains clean speech, we need to generate speech separation and denoising scenarios by creating synthetic datasets.

For both datasets, we respect the original train/validation split of the TIMIT speech dataset. The speaker separation dataset is created by randomly combining segments of speech from two different source speeches, and the speech denoising dataset is created by overlapping source speeches with noises randomly sampled from the verified AudioSet. Both datasets have a training set with size 23.1k and a validation set with size 8.3k since the original TIMIT speech dataset contains 4620 source clips for training and 1680 source clips for validation, and each speech is mixed with 5 other speeches or noises for the synthetic speaker separation dataset and speech denoising dataset. respectively.

Chapter 4

Modeling

4.1 Introduction

In recent years, many deep neural network-based models have been proposed to tackle monaural audio source separation and they have been shown to perform substantially better than traditional non-data-driven approaches[13, 14, 32]. However, they are designed for specialized source separation tasks such as speech enhancement, speaker separation, and music instrument separation, and how they would perform given more complex *real-world scenarios* still requires further investigation.

Nevertheless, the success of deep neural network-based models also motivates us to investigate novel model designs. Many aspects are involved in designing supervised learning algorithms, including but not limited to feature engineering, objective function design, and neural network modeling. In the scope of our work, we will choose appropriate feature engineering and objective functions, while primarily focusing on innovating neural network modeling. We present the *Attentive Spatio-Temporal Network (ASTNet)*, which is the first model to utilize multi-headed self-attention for monaural audio source separation.

We evaluate existing best-performing models on the *Wild-Mix Datasets* featuring *real-world scenarios* and investigate their robustness against increased complexity in the source separation tasks. We also show that *ASTNet* achieves the state-of-the-art performance among all of its competitors on the *Wild-Mix Datasets*. We will conduct additional studies regarding *ASTNet*'s performance on specialized source separation tasks using TIMIT-bsaed datasets featuring the speech enhancement and speaker separation scenarios.

We start by introducing the audio data descriptor (feature engineering) we use in section 4.2. We discuss the training target and objective function settings in section 4.3. We then present the design of o*ASTNet* in section 4.4. Finally, we will explicate our experimental setup in section 4.5 and analyze the experimental results in section 4.6.

4.2 Audio Data Descriptor

Appropriate feature engineering is essential to successful data-driven models. PCM, usually encoded in the WAV format, is the most common digital representation of audio. However, in

our work, we will use spectrograms, which can be obtained by transforming PCM data, as our audio descriptor. We will introduce both representations in the following sections and explain the reason behind our choice.

4.2.1 Pulse Code Modulation (PCM)

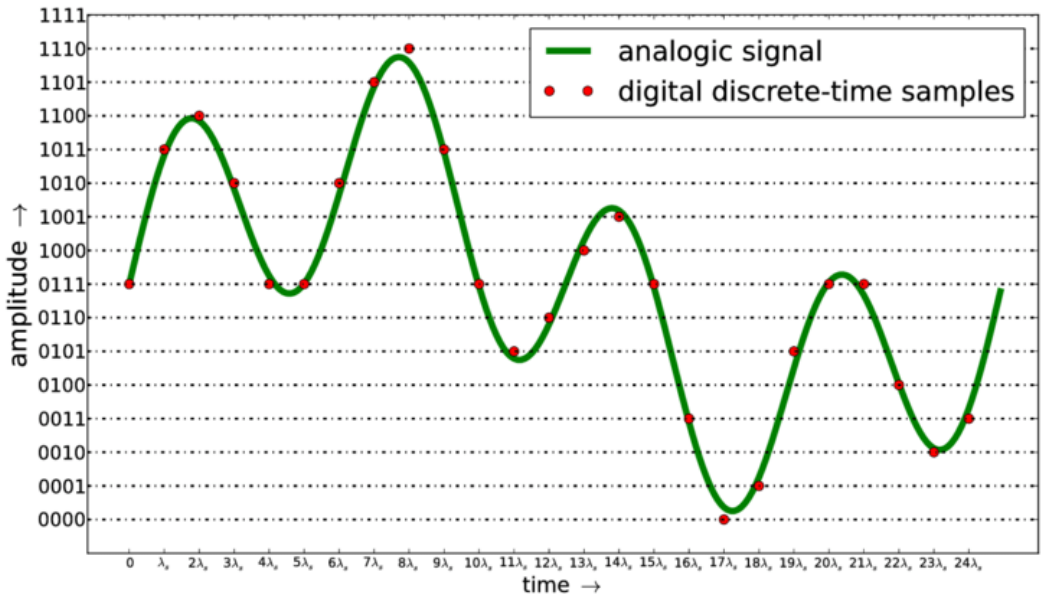


Figure 4.1: Pulse Code Modulation (PCM) audio: an analogical signal is represented by 25 samples with 4 bits each, adopted from Fabbri et al. [7]

Audio signals are most commonly represented as in the form of PCMs, also referred to as the audio waveform. Digital systems encode PCMs using discrete samples along the analogical signal, as illustrated in Figure 4.1. There are two parameters for the digital encoding: sample rate for the resolution of the encoded PCM and number of bits for encoding the amplitude range of the signal. All the datasets we use in this work are digitally encoded PCMs in the WAV format with sample rate 16kHz and 16-bit samples.

4.2.2 Spectrogram

Spectrogram is one of the most frequently adopted time-frequency representation of audio data for signal analysis. Spectrograms have both real and imaginary components, encoding the magnitude and the phase information of the original signal, respectively. It is also more expressive than audio waveforms since besides the time dimension (x axis), it also has a frequency dimension (y axis), and each time-frequency bin contains both the corresponding magnitude and phase information. A visualization of spectrogram is given in Figure 4.2. Spectrograms make

distinctive graphical patterns of different timbres within an audio mixture explicit and are thus especially suitable for source separation tasks.

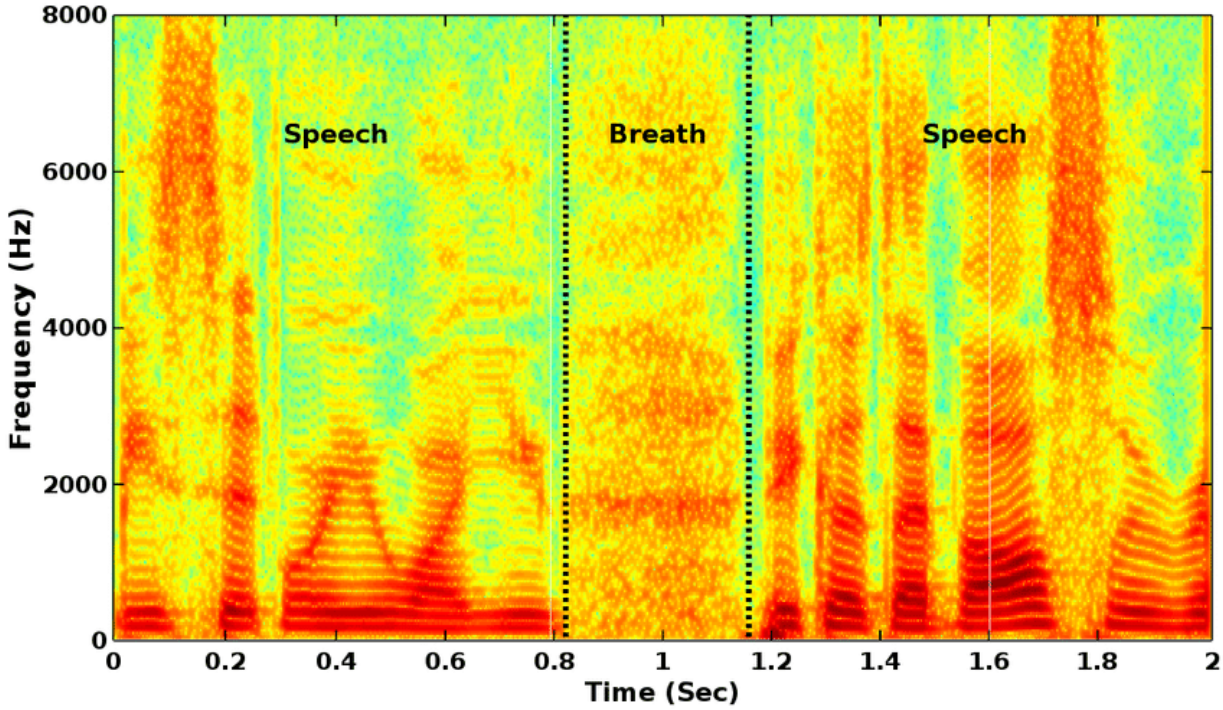


Figure 4.2: Spectrogram of a speech signal with breath sound, adopted from Dumpala et al.[4]

We obtain spectrograms by applying Short-time Fourier Transform (STFT) to 16-bit, 16kHz audio waveforms. We used the librosa [19] library for this conversion with window size 256 and hop length 196. This process is visualized in Figure 4.3, where we convert a 4-second speech waveform to its corresponding time-frequency representation using STFT.

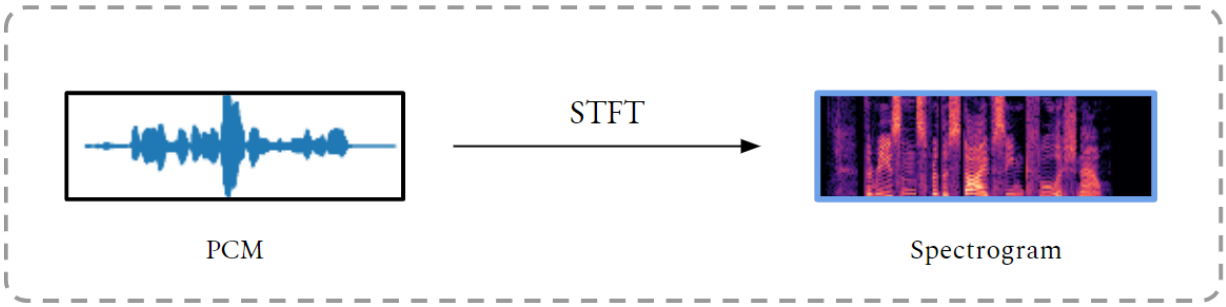


Figure 4.3: STFT conversion

There are many other time-frequency representations for audio data, including but not limited to MFCC, LOG-Mel, and MRCG [31]. Some of these representations reveal richer information than spectrograms. We chose to represent audio in the form of spectrograms for the following reasons: first, it's the most commonly adopted audio representation of existing works,

so it allows us to base our approach on many existing appraoches and compare with them more easily. Second, there's a trade-off between feature representation and learning machine [31], where the best feature puts almost no demand on model capability, while an oracle machine might excel with no feature engineering at all. We believe spectrograms obtained using STFT strike a nice balance between them by enabling learning machines to generate decent result while keeping separation challenging.

4.3 Objective Functions

There are two essential problems to address in the design of our objective function. First, we have to specify the target of optimization. Second, we need to circumvent the multi-source alignment problem during supervised training.

4.3.1 Training Targets

Since we are using spectrograms, the monaural source separation models are trained using the mixture spectrogram as the input and multiple source spectrograms as the ground truth labels. The model should simultaneously predict all of the separated sources. However, during the training process, the training targets do not necessarily need to be the source spectrograms. There are three main types of training targets that researchers have used to train spectrogram-based models: masking-based targets, mapping-based targets, and signal approximation-based targets [26]. There exist an ideal complex ratio mask (cIRM) for each source, and the spectrogram of each source can be recovered by applying each mask to the mixture spectrogram with point-wise multiplication. This ideal mask can be calculated based on the the mixture spectrogram and its corresponding source spectrograms. In the masking-based approach, the ideal mask is pre-calculated for each pair of mixture and sources and is used as the training target. In the mapping-based approach, the source spectrograms are directly used as the training target. We adopt the signal approximation-based approach, where while we use the source spectrograms as training targets, the model does not directly approximate the source spectrograms. Instead, the model first predicts the complex ratio masks to be applied to the mixture spectrogram, and the loss is then calculated between the source spectrograms and the results of applying each predicted masks to the spectrograms. This process is illustrated in Figure 4.4, where $\hat{S}$ is the approximated source, M the predicted mask, and Y the mixture.

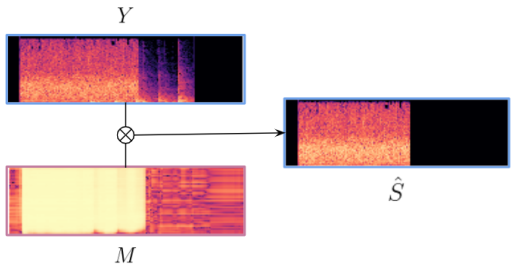


Figure 4.4: The signal approximation-based target, M

We formalize the signal approximation-based approach by first looking at the representation of spectrograms. Spectrograms obtained using STFT contain both the real and the imaginary parts, and they represent the magnitude and phase information, respectively. We utilize the complex component so that the phase information of audio data could also be modeled, which is shown to increase model performance than modeling the magnitude information alone [5, 33]. Using the signal approximation-based approach, we denote the source spectrogram (ground truth) as S, the approximated source spectrogram as $\hat{S}$, the predicted mask as M, and the input mixture spectrogram as Y. To incorporate both magnitude and phase information, we optimize the real and imaginary components separately [26]. We first decompose each component as follows:

$$\hat{S} = M * Y \tag{4.1}$$

Each spectrogram can be further decomposed into its corresponding real and imaginary components:

$$Y = Y_r + iY_i \tag{4.2}$$
$$M = M_r + iM_i \tag{4.3}$$
$$\hat{S} = \hat{S}_r + i\hat{S}_i \tag{4.4}$$

Then, we can approximate $\hat{S}$ using the predicted M as follows:

$$\hat{S} = M * Y \tag{4.5}$$
$$= (M_r + iM_i) * (Y_r + iY_i) \tag{4.6}$$
$$= (M_r Y_r - M_i Y_i) + i(M_r Y_i + M_i Y_r) \tag{4.7}$$

By equation (4.4), we have:

$$\hat{S}_r = M_r Y_r - M_i Y_i \tag{4.8}$$
$$\hat{S}_i = M_r Y_i + M_i Y_r \tag{4.9}$$

We use total Euclidean distance (L2 norm of the difference between the target and the approximated spectrogram) between the ground truths and the approximated source spectrograms, for both the real and imaginary components as the loss metric. Therefore, for each source S of an input mixture spectrogram Y with the predicted mask M, the loss is defined as

$$\mathcal{L}_S = \frac{1}{N_{seq} \cdot N_{freq}} \cdot (||\hat{S}_r - S_r||_2 + ||\hat{S}_i - S_i||_2) \tag{4.10}$$
$$= \frac{1}{N_{seq} \cdot N_{freq}} \cdot (||(M_r Y_r - M_i Y_i) - S_r||_2 + ||(M_r Y_i + M_i Y_r) - S_i||_2) \tag{4.11}$$

Note that we are normalizing the loss with the number of time-frequency bins ($N_{seq} \cdot N_{freq}$) within each spectrogram. This quantity is the same for every data point within the dataset.

Since we are training the models using mini-batches, for each mini-batch, the model is trained by optimizing against:

$$\mathcal{L}_{batch} = \frac{1}{N_{batch} \cdot N_{sources}} \sum_{Y} \sum_{S} \mathcal{L}_{S} \tag{4.12}$$

This is the averaged total loss with respect to N_{batch}, the batch size and N_s, the number of sources within the mixtures.

4.3.2 Permutation-Invariant Training

Even though the source signals within audio mixtures do not intrinsically form an ordered set, an implicit and arbitrary ordering among them is imposed by the supervised learning setting. More specifically, during the dataset creation process, since each source is randomly selected to appear in the mixture, the indexing of sources in the label(the ground truths for the separated result), is arbitrary. For example, there might be multiple mixtures containing speech and dog barking, but the corresponding ground truths could be ordered as speech followed by dog barking or the other way around. This creates confusion for the model as given a dataset that imposes an indexing of the ground truths for each misture, the model is trained to not only separate the sources but also respect their indexing. However, the randomness within the ground truth ordering makes accurate prediction impossible.

In order to circumvent this issue, we are adopting the idea of permutation-invariant training, proposed by Yu et al. [34]. The idea is to calculate the loss using every permutation of the indexing of the predicted sources and pick the permutation that results in the minimum loss. This operation ensures that the arbitrary indexing of within each label will not matter during the training process. It clearly does not scale with increasing number of sources, but is sufficiently fast for our experiments given that our datasets contain at most 5 sources. Given s sources, the Hungarian algorithm is the fastest known algorithm since it finds the best permutation within $O(n^3)$ time. It should be adopted as an alternative implementation for permutation-invariant training for future source separation settings with large s.

4.4 Model Design

Better neural network design is the core of a more successful data-driven approach for monaural source separation. Existing models have identified the importance of utilizing temporal dependency within audio data and have shown that recurrent models introduce significant performance improvement [13, 14]. Other works also explored the potential of exploiting the rich visual information offered by spectrograms using convolutional layers [12, 21]. We discover another factor that we deem essential to better performance: the ability to utilize more useful contextual information. Our proposed model design, *ASTNet*, is able to utilize dynamic contextual information using multi-headed self-attention, the main technique used in the transformer model for sequence to sequence modeling [28].

We will start by showing the overall architecture of the *ASTNet*. We will then discuss each component within the *ASTNet* and their functionalities in more details in the subsequent sections.

4.4.1 Overall Architecture

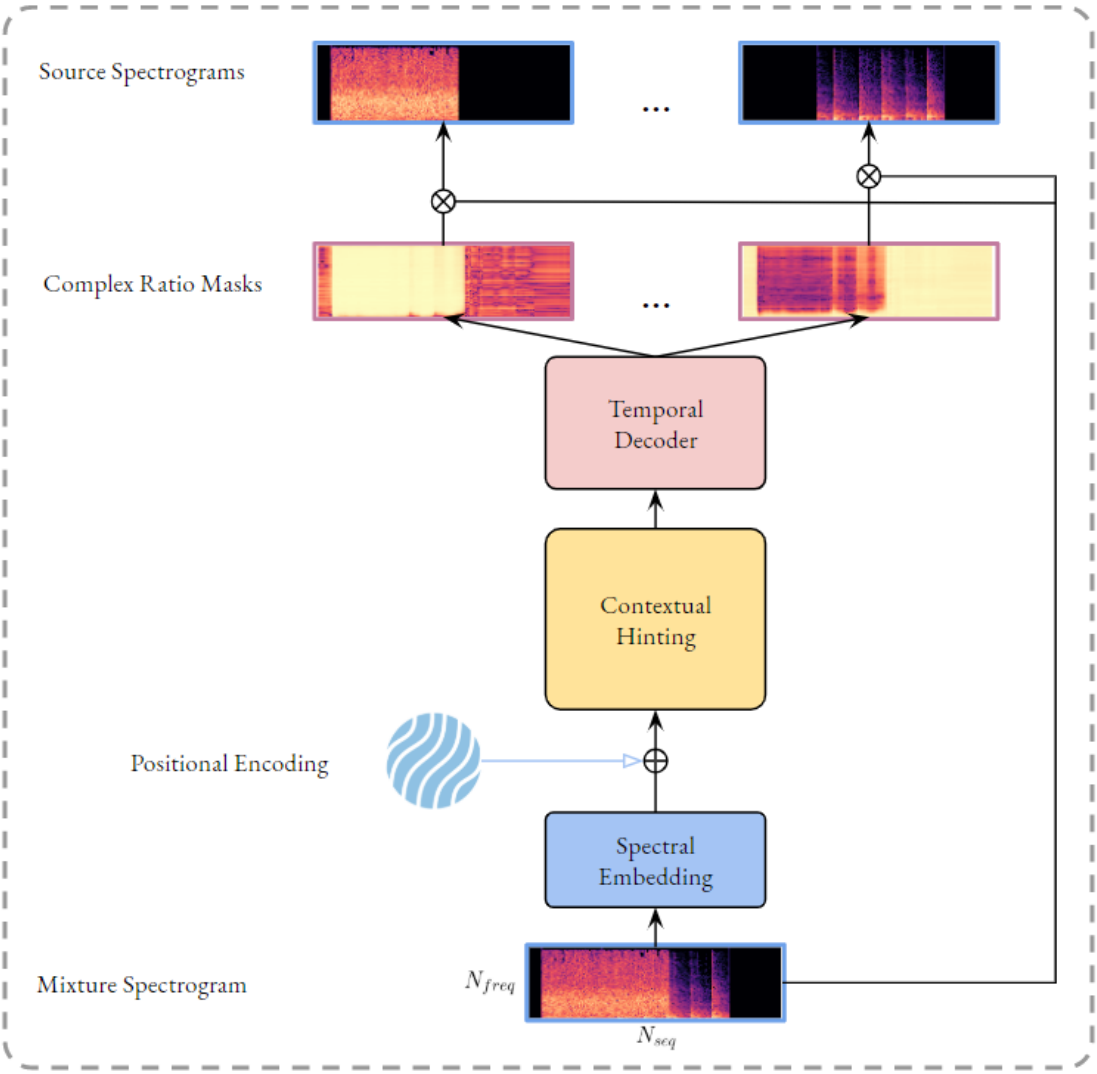


Figure 4.5: Overall architecture of the ASTNet.

We present the *Attentive Spatio-Temporal Network (ASTNet)*, and its architecture is illustrated in Figure 4.5.

Each mixture spectrogram with $N_{seq} \times N_{freq}$ time-frequency bins is first fed into *spectral embedding*, a CNN-based embedding component, then into *contextual hinting*, a component for emphasizing contextual information using positional encoding and multi-headed self-attention, and finally to *temporal decoder* that utilizes temporal dependency in the decoding process. The output of the model contains the complex ratio mask for each source, and the source spectrograms are approximated using these complex ratio masks according to the signal approximation approach mentioned in Section 4.3.1.

4.4.2 Network Components

We elaborate each component of the ASTNet with more details. We will discuss them in the order they appear in the overall architecture, from input to output. Throughout the model, ReLU is used after each linear layer as the non-linear activation function.

Spectral Embedding

We have discovered that many existing models are utilizing convolutional neural network (CNN) given that it is able to extract useful visual features from the time-frequency representation of audio data.

In our case, the spectral embedding within *ASTNet* contains deep convolutional sub-layers, as illustrated in Figure 4.6. Each sub-layer contains a dilated convolution layer followed by a batch normalization layer. The last convolutional layer is followed by a linear layer before exiting the spectral embedding component.

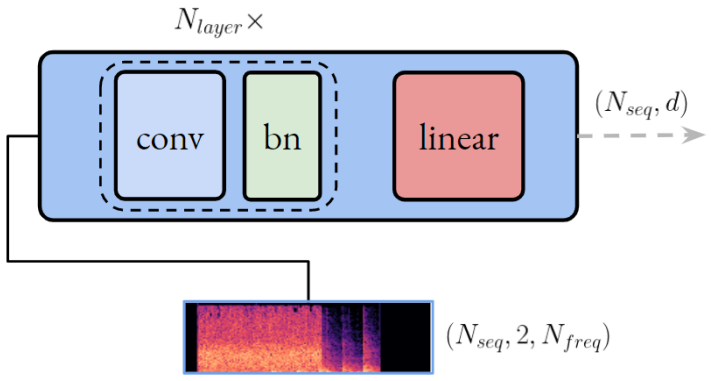


Figure 4.6: The spectral embedding.

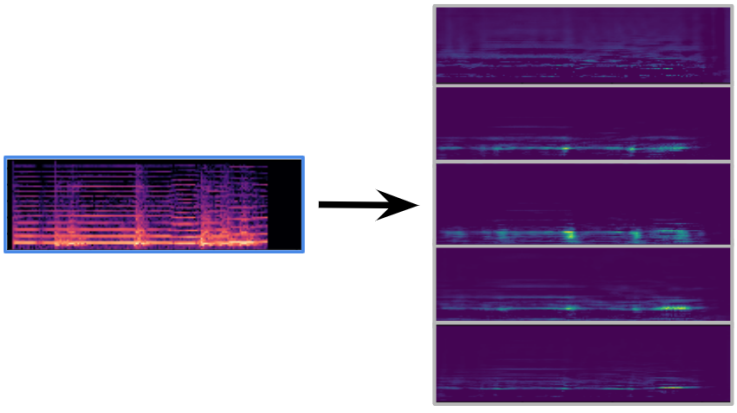


Figure 4.7: An example intermediate output before the last linear layer.

The input to the spectral embedding are both the real and imaginary components of the mixture spectrogram with $N_{seq} \times N_{freq}$ time-frequency bins, and the output of the spectral embedding is a tensor with size N_{seq} by d, where d is the encoding size for the subsequent contextual hinting component.

We argue that the outputs of the spectral embedding function similarly as a dictionary in an autoencoder. Given a mixture spectrogram, it generates a set of feature decompositions of the mixture. These decompositions offer richer information regarding the composition of the mixture for the following layers in the network than the mixture spectrograms alone. An example of the decompositions of the mixture containing violin and dog barking is obtained by visualizing the intermediate output before it goes into the final feed-forward layer of the spectral embedding, as shown in Figure 4.7.

Contextual Hinting

The contextual hinting utilizes multi-headed self-attention. The internals of the component is illustrated in Figure 4.8:

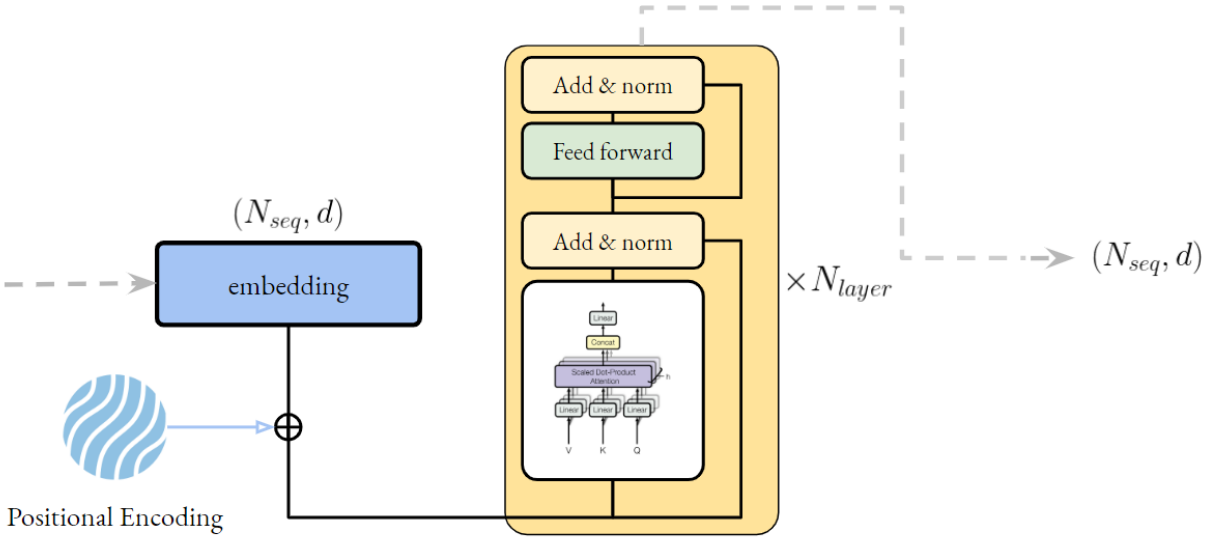


Figure 4.8: The contextual hinting.

The input to the contextual hinting is the output of the spectral embedding with dimension N_{seq} by d. The output of the contextual hinting is a condensed representation of the result of applying multi-headed self-attention. The output has the exact same dimension as the output from the spectral embedding.

Intuitively speaking, when human auditory system tries to disambiguate sources from mixtures, it tends to recognize and replay individual source signals after hearing the entire input signal, instead of solely relying on previous timestamps to make decisions. Similarly, segments within a mixture where only a single source is present provide auditory cues for the model when it tries to separate overlapping sources.

We argue that multi-headed self-attention helps the model to pay attention to these contextual cues during the source separation process. Multiple attention heads provide great flexibility in the model's ability to focus on different areas within the input, especially given that there are multiple sources present in the mxiture and there are also multiple decompositions provided by the spectral embedding.

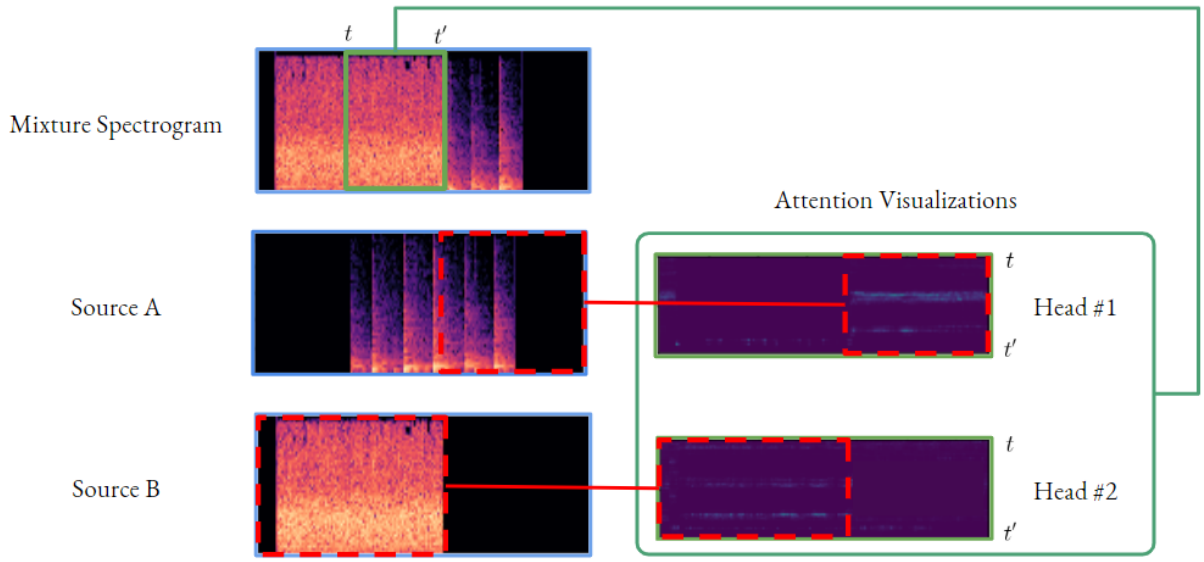


Figure 4.9: Visualization of multi-headed self-attention

The multi-headed self-attention mainly uses Q and K, the query and key vectors, to calculate attention scores for different representation subspaces. We discover that the attention vectors, obtained using

$$\text{softmax}(\frac{QK^T}{\sqrt{d}})$$

, align with our speculation that during a segment that is congested (the most overlapped segment within the mixture), the attention vectors help our model to focus on the less congested segment with respect to different sources. This is illustrated by an example in Figure 4.9, the most congested region is marked by a segment starting from timestemp t and ending at t', where both source A and source B are present. According to the visualization, one of the attention vectors is helping each frame between t and t' to focus on the segment where only source A is present, the other is helping each frame to focus on the segment where only source B is present.

Temporal Decoder

The ability to capture long-term temporal dependency has been shown to be extremely important in building better source separation models, as evidenced in [13, 14]. Therefore, the temporal decoder is based on Long Short-term Memory (LSTM), which not only models temporal

dependency but also prevents the problem of exploding or vanishing gradients during BPTT, a problem that often occurs in vanilla RNNs [2, 3]. The internals of the temporal decoder are illustrated in Figure 4.10:

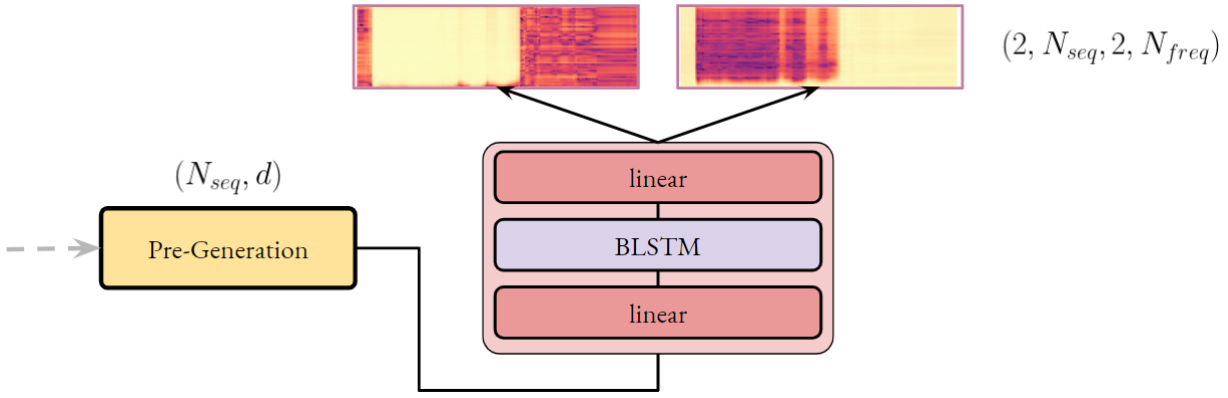


Figure 4.10: The temporal decoder

The input to the decoder is the output from the contextual hinting component with dimension N_{seq} by d. The outputs of the decoder are the estimated complex ratio masks to be applied to the mixture for approximating source spectrograms.

4.5 Experiment Setup

We thoroughly evaluate the models using both *Wild-Mix Datasets* that feature *real-world scenarios* and TIMIT-based speech separation datasets that feature speech enhancement and speaker separation scenarios. We investigate how the existing models respond to the increased complexity of the datasets.

We keep the feature engineering and objective function settings consistent for all of the models. We only use the training and validation set of each dataset since no cross-validation is conducted.

Since the idea of simultaneously separating multiple sources in the *real-world scenarios* is novel, we think there is no formal metric for evaluating the separation result for this setting yet. Therefore, we compare the performance of different models using their validation loss (EU). For the TIMIT-based speech separation datasets, we measure the widely used metric developed by Vincent et al. [29]. Specifically, we use the sound-to-distortion ratio (SDR) improvement as the performance metric for speech enhancement and speaker separation. SDR is measured in dB and is the most general score used for speech separation models [5]. We utilize the `mir_eval` library for SDR calculation [24]. The performance ditribution among models measured using validation loss is consistent with that measured using SDR improvement.

4.5.1 Candidate Models and Training Setting

We select a set of the most competitive models from existing works and use them as baselines. Specifically, we selected cSA-LSTM proposed by Sun et al. [26], which is shown to achieve better performance than most RNN-based models. Erdogan et al. have also argued that BLSTMs are better than LSTMs in speech related tasks [6]. Therefore, we introduced the bidirectional component into the cSA-LSTM model to create another baseline called cSA-BLSTM. Ephrat et al. have proposed an audio-visual model in [5], which combines dilated convolution layers with BLSTM in its audio-only baseline. The audio-only baseline which we denote as L2L-AO is claimed to be the state-of-the-art model at the time it is proposed, so it could serve as a competitive baseline for the *real-world scenarios* as well. We also include Transformer+, which is a transformer baseline that only contains the transformer encoder with a BLSTM decoder similar to the temporal decoder component of *ASTNet*.

All the hyperparameter settings for each model are optimized using extensive grid search and are subject to change based on different amount of available computational resources. All of the models are trained using the Adam optimizer.

4.6 Result Analysis

4.6.1 Overall Results

	Validation EU (10-category)								
	2S inter	2S hybrid	2S intra	3S inter	3S hybrid	3S intra	5S inter	5S hybrid	5S intra
cSA-LSTM	10.1	10.4	12.0	12.4	13.0	14.2	14.5	14.9	16.2
cSA-BLSTM	9.2	9.5	12.0	12.0	12.8	14.2	14.4	14.6	16.3
Transformer+	8.6	9.0	11.6	11.4	12.0	14.2	13.7	14.2	16.1
L2L-AO	7.9	8.5	10.9	10.5	11.4	13.9	13.0	13.6	15.9
ASTNet	**7.0**	**7.4**	**10.5**	**9.6**	**10.8**	**13.6**	**12.6**	**13.2**	**15.7**

Table 4.1: Performance over real-world datasets with category scope 10.

	Validation EU	
	5-category	30-category
L2L-AO	7.1	9.1
ASTNet	**6.7**	**8.3**

Table 4.2: Performance on 2-source, inter-class datasets with source category scope 5 and 30.

The *ASTNet* achieves the state-of-the-art results on all of the *Wild-Mix Datasets*. However, we have noticed that it performs slightly worse than L2L-AO, the most competitive baseline, on the speech enhancement dataset (Table 4.3). It should be noted that the speech separation is the only task where each mixture (noisy speech) has only correspond with one source spectrogram (clean speech). It seems that ASTNet is suitable for more complicated cases where there are at

	SDR Improvement	
	Enhancement	Separation
cSA-BLSTM	6.7	1.5
L2L-AO	**10.3**	5.0
ASTNet	10.2	**6.6**

Table 4.3: SDR improvement on TIMIT-enhance dataset and TIMIT-separation dataset.

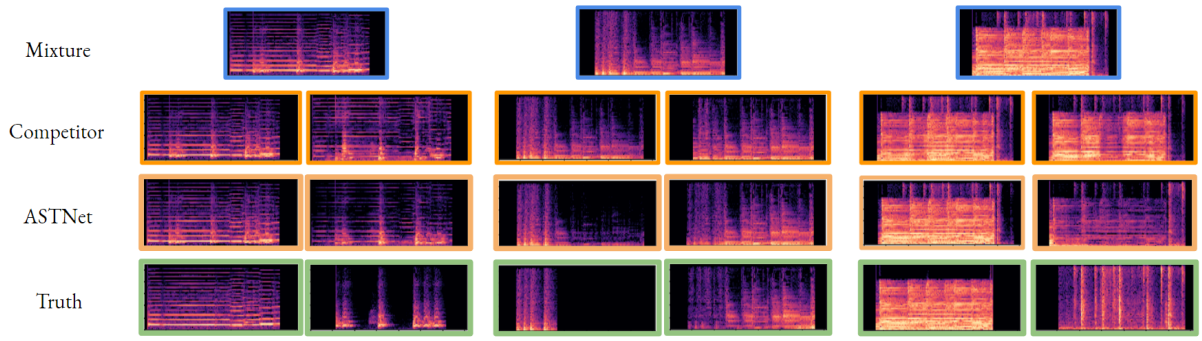


Figure 4.11: Separation results in the form of spectrograms.

least two source spectrograms to approximate. This is evidenced by its superior performance on the speaker separation task, where each mixture spectrogram (mixed speech) corresponds to two source spectrograms (separated speeches), as well as all the *Wild-Mix Datasets* in which there are always at least 2 sources present in a mixture.

4.6.2 Source Separation Visualizations

We showcase some of the separation results using *ASTNet* and compare them with L2L-AO, its strongest competitor in Figure 4.11.

4.6.3 Ablation Study

We have also conducted ablation study of *ASTNet* to learn about the contribution of individual components within the model. It is interesting to see that permutation-invariant training is an indispensable external manipulation for training supervised source separation algorithms. However, future works should study how to fundamentally solve the permutation problem as sources within are audio mixture are intrinsically not an ordered set. Multimodal approaches are able to tackle this by utilizing visual cues [5] corresponding to each sources, but it is still an open problem for the monaural case. We also notice that temporal dependency is an extremely important factor to consider for monaural source separation, as it introduces the greatest performance improvement to the final model.

	2S Inter, 10-category
Full Model	**7.0**
- no sequential generator	9.0
- no spectral embedding	8.6
- no contextual attention	7.9
- no PIT	15.2

Table 4.4: Ablation Study: we analyze the significance of different parts of our model design.

4.6.4 Observations

We also visualized some trends from the experiment results using the *Wild-Mix Datasets* to investigate how monaural source separation models respond to variations in the complexity of the datasets. We find the following research questions to be particularly interesting:

Robustness against increasing number of sources

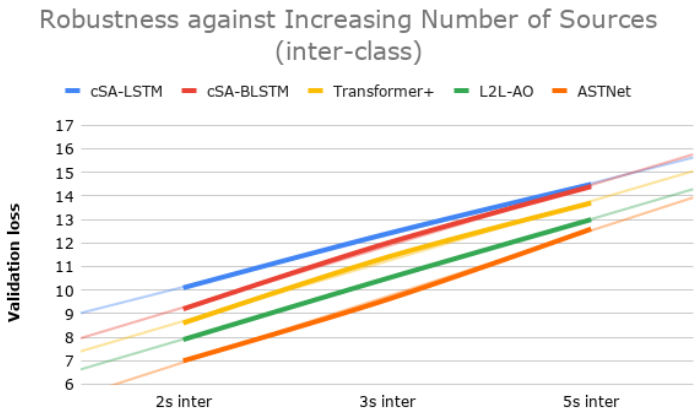
While setting everything else stationary, we only vary the number of sources within the datasets. For each mix method, we plot the performance of models under datasets with 2 sources, 3 sources, and 5 sources, respectively. Increased number of sources directly makes the source separation process more challenging. Surprisingly, all of the models do not respond to the jump agnostically as the curve is tend to be more flat as the number of sources increases.

Robustness against Source Homogeneity

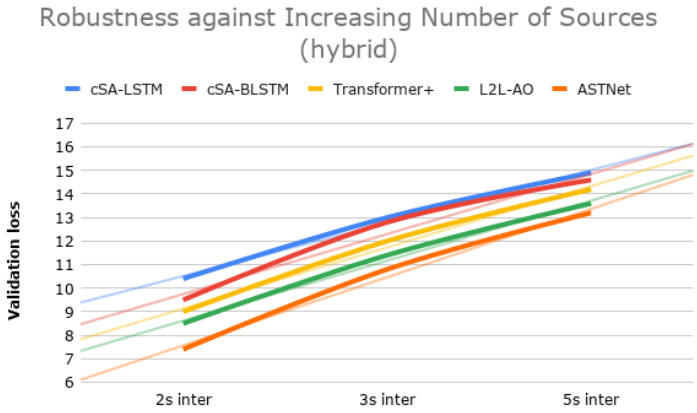
We do the same thing with the mix methods which represent different levels of hetero/homogeneity of mixtures. For each source count among 2, 3, and 5, we plot the performance of models using inter-class, hybrid, and intra-class mix methods, respectively. It seems like existing source separation models suffers more from higher homogeneity than from increasing number of sources.

Robustness against Source Heterogeneity

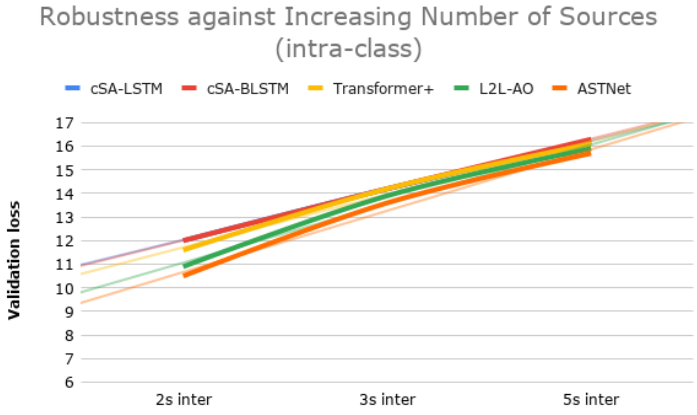
We have 9 dataset configurations with category scope set to 10. We believe it's unnecessary to repeat all 9 configurations for the case where we have 5 and 30 as the category scope and experiment with all the baselines, since the overall distribution of model performance will be very similar and too many additional experiments will be redundant. Therefore, we only generate datasets using the 2-source inter-class configuration with category scopes 5 and 30 and only evaluate *ASTNet* against L2L-AO, the most competitive baseline, just to demonstrate the effect of altering heterogeneity by having different category scopes. We observe that models do get affected by the higher variations of source categories, and it seems like *ASTNet* is less tolerant to higher heterogeneity than L2L-AO, even though its absolute performance is better. We observer similar trends for the previous two research questions as well, since the curves of *ASTNet* are the steepest even though its performance is superior. We speculate that this might be due to the fixed number of attention heads under increasing complexity.



(a) EU of datasets with inter-class mix method.

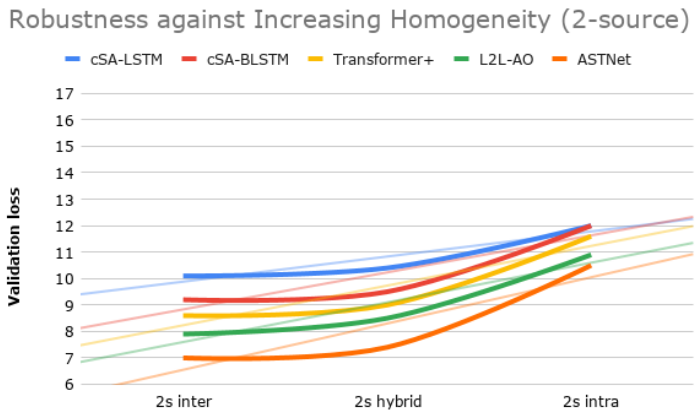


(b) EU of datasets with hybrid mix method.

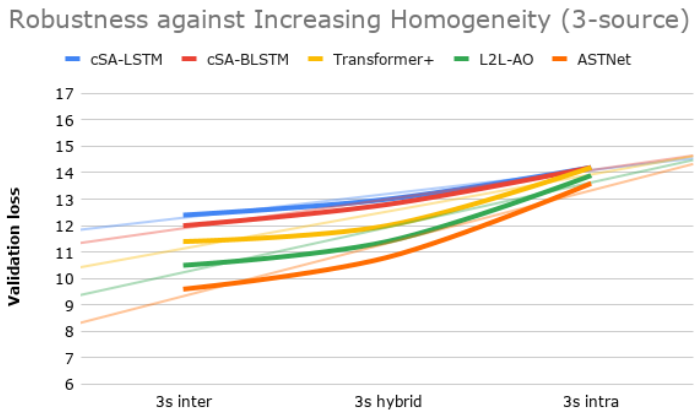


(c) EU of datasets with intra-class mix method.

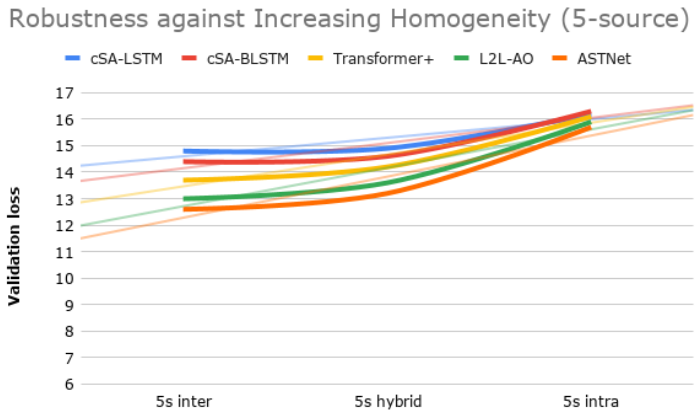
Figure 4.12: Robustness against increasing number of sources



(a) EU of 2-source datasets with different mix methods.



(b) EU of 3-source datasets with different mix methods.



(c) EU of 5-source datasets with different mix methods.

Figure 4.13: Robustness against increasing homogeneity

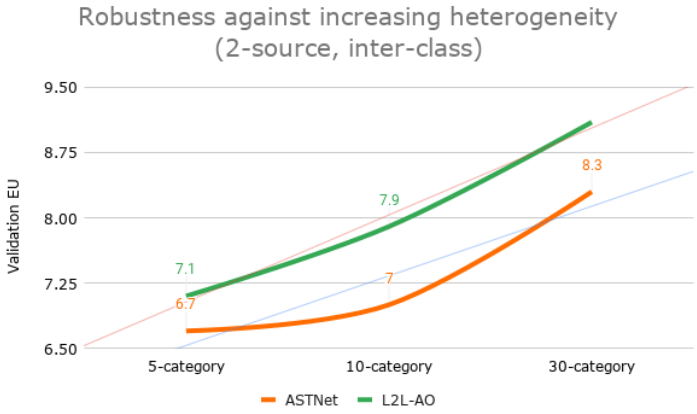


Figure 4.14: Robustness against increasing heterogeneity

Chapter 5

Conclusion

5.1 Summary

In this work, we have formalized the notion of *real-world scenarios*, which specified a set of standards for monaural source separation datasets to be representative of diverse and complex real world auditory scenes. We have also created the *Wild-Mix Datasets* as challenging benchmarks for evaluating source separation algorithms in the *real-world scenarios* along with the correpsonding data synthesis library. Beside datasets, we have also proposed *ASTNet*, a novel deep neural network-based model that utilizes multi-headed self-attention to capture dynamic contextual cues during the source separation process. In the end, we conduct extensive experiments to investigate how existing models developed for specialized source separation tasks, as well as *ASTNet*, respond to different levels of complexity in the *real-world scenarios*. We show that *ASTNet* perform consistently better on more complex *real-world scenarios* than other existing models, but we also discover that all models, including *ASTNet*, are especially vulnerable to increasing homogeneity of audio mixtures.

5.2 Future Works

Many areas can be further studied based on our observations. For model design, future research should focus on how to better utilize more information effectively from the spectrograms. Even though *ASTNet* achieves better overall performance, it seems to be less resilient to increasing complexity of the datasets than existing baselines, given that its performance worsens at a steeper rate, and the reasons behind this phenomenon could also be studied. What's more, the supervised setting introduces the inevitable source permutation problem, and developing fundamental solutions beyond external manipulations is an interesting research topic as well.

Given the versatile dataset synthesis library, it also provides a flexible infrastructure to create datasets targeting zero-shot and few-shot learning. Additionally, since AudioSet is an audio-visual dataset, datasets used for multimodal source separation can also be created with marginal amount of functionalities added to the synthesis library. *ASTNet*, then, could serve as a strong audio-only baseline or provide insights regarding model design for multimodal approaches.

Bibliography

[1] L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):191–199, 2006. 2.2.1

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. 2.2.4, 4.4.2

[3] Jitong Chen and DeLiang Wang. Long short-term memory for speaker generalization in supervised speech separation. pages 3314–3318, 09 2016. doi: 10.21437/Interspeech.2016-551. 2.2.4, 4.4.2

[4] Sri Harsha Dumpala and K N R K Alluri. An algorithm for detection of breath sounds in spontaneous speech with application to speaker recognition. pages 98–108, 08 2017. ISBN 978-3-319-66428-6. doi: 10.1007/978-3-319-66429-3_9. (document), 4.2

[5] A. Ephrat, I. Mosseri, O. Lang, T. Dekel, K Wilson, A. Hassidim, W. T. Freeman, and M. Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *arXiv preprint arXiv:1804.03619*, 2018. 2.1.1, 2.2.5, 4.3.1, 4.5, 4.5.1, 4.6.3

[6] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 708–712, 2015. 4.5.1

[7] Renato Fabbri, Vilson Vieira, Antonio Pessotti, and Débora Corrêa. Psychophysics of musical elements in the discrete-time representation of sound. 12 2014. (document), 4.1

[8] Jennifer Flenner and Blake Hunter. A deep non-negative matrix factorization neural network. 2017. 2.2.1

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren. Darpa timit acoustic phonetic continuous speech corpus cdrom, 1993. 1.2, 2.1.1, 3.5.2

[10] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017. 3.3, 3.3.1

[11] Laurent Girin, Sharon Gannot, and Xiaofei Li. Audio source separation into the wild. In *Multimodal Behavior Analysis in the Wild*, Computer Vision and Pattern Recognition, pages 53–78. Academic Press (Elsevier), November 2018. doi: 10.1016/B978-0-12-814601-9.

00022-5. URL https://hal.inria.fr/hal-01943375. 3.2

[12] E. M. Grais and M. D. Plumbley. Single channel audio source separation using convolutional denoising autoencoders. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1265–1269, 2017. 2.2.3, 4.4

[13] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1562–1566, 2014. 1.1, 2.1.1, 2.2.1, 4.1, 4.4, 4.4.2

[14] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2136–2147, 2015. 1.1, 2.1.1, 2.2.1, 2.2.4, 4.1, 4.4, 4.4.2

[15] D. Paul J. Garofalo, D. Graff and D. Pallett. Csri (wsj0) complete, 2007. 2.1.1

[16] X. Jaureguiberry, P. Leveau, S. Maller, and J. J. Burred. Adaptation of source-specific dictionaries in non-negative matrix factorization for source separation. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5–8, 2011. 2.2.1

[17] T. G. Kang, K. Kwon, J. W. Shin, and N. S. Kim. Nmf-based target source separation using deep neural network. *IEEE Signal Processing Letters*, 22(2):229–233, 2015. 2.2.1

[18] A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard. Adaptive filtering for music/voice separation exploiting the repeating musical structure. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 53–56, 2012. 2.2.1

[19] Brian McFee, Vincent Lostanlen, Matt McVicar, Alexandros Metsai, Stefan Balke, Carl Thomé, Colin Raffel, Ayoub Malek, Dana Lee, Frank Zalkow, Kyungyun Lee, Oriol Nieto, Jack Mason, Dan Ellis, Ryuichi Yamamoto, Scott Seyfarth, Eric Battenberg, , Rachel Bittner, Keunwoo Choi, Josh Moore, Ziyao Wei, Shunsuke Hidaka, nullmightybofo, Pius Friesch, Fabian-Robert Stöter, Darío Hereñú, Taewoon Kim, Matt Vollrath, and Adam Weiss. librosa/librosa: 0.7.2, January 2020. URL https://doi.org/10.5281/zenodo.3606573. 4.2.2

[20] Keiichi Osako, Yuki Mitsufuji, Rita Singh, and Bhiksha Raj. Supervised monaural source separation based on autoencoders. pages 11–15, 03 2017. doi: 10.1109/ICASSP.2017.7951788. 2.2.1, 2.2.1, 2.2.2

[21] A. Pandey and D. Wang. A new framework for cnn-based speech enhancement in the time domain. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(7):1179–1188, 2019. 2.2.3, 4.4

[22] Se Rim Park and Jinwon Lee. A fully convolutional neural network for speech enhancement. In *INTERSPEECH*, 2017. 2.2.3

[23] Santiago Pascual, Joan Serrà, and Antonio Bonafonte. Towards generalized speech enhancement with generative adversarial networks. *CoRR*, abs/1904.03418, 2019. URL http://arxiv.org/abs/1904.03418. 3.2

[24] Colin Raffel, Brian Mcfee, Eric Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and

Daniel Ellis. mir$_e$*val* : *A transparent implementation of common mir metrics*. 102014. 4.5

[25] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December 2017. URL `https://doi.org/10.5281/zenodo.1117372`. 2.1.1

[26] Y. Sun, Y. Xian, W. Wang, and S. M. Naqvi. Monaural source separation in complex domain with long short-term memory neural network. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):359–369, 2019. 4.3.1, 4.3.1, 4.5.1

[27] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 261–265, 2017. 2.1.2, 3.2

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL `http://arxiv.org/abs/1706.03762`. 4.4

[29] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006. 4.5

[30] Emmanuel Vincent, Jon Barker, Shinji Watanabe, Jonathan Le Roux, Francesco Nesta, and Marco Matassoni. The second 'chime' speech separation and recognition challenge: An overview of challenge systems and outcomes. pages 162–167, 12 2013. doi: 10.1109/ASRU.2013.6707723. 2.1.1

[31] D. Wang and J. Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018. 1.1, 2.2.1, 2.2.4, 4.2.2

[32] Yuxuan Wang and Deliang Wang. Cocktail party processing via structured prediction. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 224–232. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4838-cocktail-party-processing-via-structured-prediction.pdf`. 1.1, 2.2.1, 2.2.2, 4.1

[33] D. S. Williamson, Y. Wang, and D. Wang. Complex ratio masking for joint enhancement of magnitude and phase. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224, 2016. 4.3.1

[34] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245, 2017. 4.3.2

[35] H. Zhao, S. Zarar, I. Tashev, and C. Lee. Convolutional-recurrent neural networks for speech enhancement. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2401–2405, 2018. 2.2.5