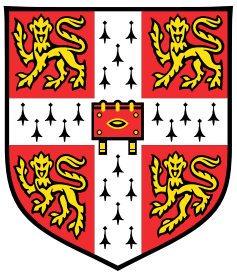
Effectiveness of SSL Representations for Source Separation



Yuang Li

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Clare College August 2022

I would like to dedicate this thesis to my loving parents.

Declaration

I, Yuang Li of Clare College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

All experiments were carried out in Python, utilizing Pytorch. The open sourced Conformer was used as the downstream model [1]. The pre-trained SSL models including WavLM [2], UniSpeech-SAT [3], Wav2Vec2 [4] and TERA [5] on Hugging Face and S3prl were used as base models. These models were modified so they fit into my project. The training and evaluation procedures were implemented from scratch. In section 5.3, pre-trained ECAPA-TDNN [6] provided by Speechbrain was used to extract speaker embeddings in source selection. In section 5.4, the ASR model fine-tuned on AMI, the VAD and diarisation results were provided by Xianrui Zheng. The transcription system was extended to include source separation. Other ASR models came from Hugging Face [7][8].

Word Count: 14982 (including tables, footnotes, captions, and appendices)

Yuang Li
August 2022

[1] https://github.com/Sanyuan-Chen/CSS_with_Conformer
[2] https://huggingface.co/microsoft/wavlm-base-plus
[3] https://huggingface.co/microsoft/unispeech-sat-base-plus
[4] https://huggingface.co/facebook/wav2vec2-base
[5] https://github.com/s3prl/s3prl
[6] https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb
[7] https://huggingface.co/facebook/wav2vec2-large-960h-lv60-self
[8] https://huggingface.co/facebook/wav2vec2-large-robust-ft-swbd-300h

Acknowledgements

I would like to thank Prof. Phil Woodland and Xianrui Zheng for their insightful opinions and continual support. Throughout the project, I received patient guidance in terms of both research methods and technical details. I am grateful that they introduced me to the interesting field of speech signal processing and self-supervised models. This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service.

Abstract

In everyday conversation, speech overlap occurs frequently, leading to the degradation of speech intelligibility and a significant challenge for automatic speech recognition (ASR). This thesis focuses on applying large self-supervised learning (SSL) models for source separation that estimates single-speaker audio streams from overlapped speech so that the robustness of ASR system can be improved.

As an initial investigation of the effectiveness of SSL representations in source separation, four SSL models were compared on simulated datasets. In contrast to existing works where SSL models were frozen, the potential of SSL models was fully exploited through a two-phase fine-tuning schedule where the lightweight downstream model was first trained and then the SSL model was fine-tuned. As the result, all SSL models provided better representations than a spectrogram, and WavLM displayed the best separation performance as it was pre-trained on the largest dataset which includes overlapped speech. By further combining two SSL models including TERA and WavLM, an absolute word error rate (WER) reduction of 0.4% was observed on the LibriCSS dataset.

Experiments were then extended to the real-world speech corpus in which case ground-truth signals are not accessible. Hence, the time-frequency domain unsupervised mixture invariant training, modified from the original time domain method was introduced to fine-tune the model with real overlapped data which enhanced the in-domain performance. In order to insert the separation model between diarisation and ASR inside an automatic transcription system, a novel iterative source selection method was proposed which automatically chooses the desired output source according to the speaker information provided by the diarization system. Absolute reductions of 1.5% and 1.9% in concatenated minimum-permutation WER for an unknown number of speakers (cpWER-us) were observed on the AMI test and development sets respectively when the separation system was used to remove overlaps and the ASR model was fine-tuned on separated speech to handle system noise. By combining the hypotheses from multiple systems through ROVER, the absolute cpWER-us reductions

reached 2.1% and 2.4% on AMI test and development sets.

Audio samples are available at *https://sites.google.com/view/sourceseparation/home*.

Table of contents

List of figures

List of tables

Nomenclature

AM Amplitude Mask

ASR Automatic Speech Recognition

BLSTM Bidirectional Long Short-Term Memory

BN Batch Normalisation

CNN Convolutional Neural Network

cpWER-us Concatenated Minimum-Permutation Word Error Rate for an Unknown Number of Speakers

CTC Connectionist Temporal Classification

DER Diarisation Error Rate

FFN Feed-Forward Network

FFT Fast Fourier Transform

GLU Gated Linear Unit

IAM Ideal Amplitude Mask

IBM Ideal Binary Mask

IPSM Ideal Phase-Sensitive Mask

IRM Ideal Ratio Mask

ISTFT Inverse Short-Time Fourier Transform

LSTM Long Short-Term Memory

MFCC Mel-Frequency Cepstral Coefficient

MHSA Multi-Head Self-Attention

MixIT Mixture Invariant Training

MoM Mixture of Mixture

MSE Mean Square Error

NLP Natural Language Processing

PIT Permutation Invariant Training

PSM Phase-Sensitive Mask

RIR Room Impulse Response

RNN Recurrent Neural Network

SI-SNR Scale-Invariant Signal-to-Noise Ratio

SNR Signal-to-Noise Ratio

SSL Self-Supervised Learning

STFT Short-Time Fourier Transform

T-F Time-Frequency

VAD Voice Activity Detection

WER Word Error Rate

Chapter 1

Introduction

Humans' remarkable capability of concentrating on and understanding a single audio source in noisy environments with the interference of multiple speech signals is known as the cocktail party problem (Haykin and Chen, 2005). Such inherent ability of humans is a desired property for ASR systems since most real-world speech signals are contaminated by noise and some overlaps. Recently, various deep learning-based methods have been proposed to improve the robustness of ASR systems against ambient noise. These methods either directly recognise noisy speech or firstly recover clean signals (Weninger et al., 2015; Zhang et al., 2018). However, interfering speakers, or overlaps, can still lead to severe degradation of recognition performance because most ASR systems assume a single active speaker.

Source separation aims to extract individual speaker signals from overlapped speech signals. It is an essential front-end preprocessing step for ASR so that the single-speaker assumption is satisfied. To develop a deep learning model for source separation, the clean single-speaker speech signals which correspond to overlapped speech are required as references for supervised learning. However, such ground-truth signals are hard to obtain in real-world scenarios. Therefore, one common approach is to create synthetic datasets (Cosentino et al., 2020; Hershey et al., 2016) by combining multiple single-speaker segments. Models that predict time-frequency (T-F) masks (Chen et al., 2021b; Kolbæk et al., 2017) or directly separate time-domain waveforms (Luo et al., 2020; Luo and Mesgarani, 2018; Subakan et al., 2021), were developed based on these synthetic datasets. The trend is to use the time-domain model with fine-grained features and optimise signal-based criteria.

Although separation performance evaluated by signal-based metrics like scale-invariant signal-to-noise ratio (SI-SNR) has kept improving in recent years (Luo et al., 2020; Luo and Mesgarani, 2019; Subakan et al., 2021), it may not reflect real situations. Therefore,

researchers have focused on creating more realistic datasets by adding ambient noises, simulating reverberation (Maciejewski et al., 2020), and using non-fully overlapped speech signals (Cosentino et al., 2020). However, the additive mixing process is inherently not real. To utilise real overlapped data without ground-truth references, researchers have proposed unsupervised training methods (Han and Long, 2022; Sivaraman et al., 2022; Wisdom et al., 2020) which use pseudo-labels from well-trained separation models or exploit the consistency between separated sources and the mixture. Due to the lack of references, most evaluations were still based on synthetic mixtures which require identifying and combining non-overlapped signals in the real corpus. Moreover, these methods concentrated on time-domain instead of T-F domain models.

With the emergence of BERT (Devlin et al., 2019) and GPT (Radford et al., 2019), self-supervised learning (SSL) has achieved impressive success in natural language processing (NLP). SSL allows models to learn general representations from a large amount of unlabelled data. These SSL models can be adapted to various downstream tasks by adding a task-specified layer and fine-tuning with a limited amount of labelled data. SSL has also shown promising results in speech-related tasks including ASR, speaker verification and source separation etc. (Yang et al., 2021). Chen et al. (2022a) set the state-of-the-art separation performance on the LibriCSS dataset (Chen et al., 2020) with WavLM. Huang et al. (2022) further compared 13 SSL models on the LibriMix dataset (Cosentino et al., 2020) to demonstrate their effectiveness. However, these comparisons were based on simulated datasets. Furthermore, SSL models were only used as feature extractors (i.e. the parameters were frozen) and large downstream models were trained for a considerable amount of time.

In this thesis, a source separation system for real-world overlapped speech was developed. The SSL model was efficiently fine-tuned through both supervised and unsupervised learning to predict T-F masks for individual speakers. Automatic source selection methods were explored so the source separation model can be directly incorporated into a transcription system.

1.1 Contributions

- An investigation of how to efficiently fine-tune SSL models for source separation rather than existing work where SSL models were frozen.
- A comprehensive comparison between four SSL models including WavLM, Unispeech-SAT, Wav2Vec, and TERA. Besides, using a single model, the combination of WavLM and TERA was investigated and a tri-stage fine-tuning schedule was proposed which enhanced the performance on the LibriCSS dataset.
- An adaptation of time-domain unsupervised mixture invariant training (MixIT) (Wisdom et al., 2020) to T-F domain. MixIT was adopted to fine-tune separation models with real overlapped signals and improved the performance on the real-world AMI dataset.
- An investigation of extending source separation models to a speaker extraction scenario where the desired output source should be chosen automatically. Based on different assumptions, two novel methods were proposed which exploit input signal embedding and average speaker embedding respectively.
- Integration of a source separation model into a fully automatic transcription system where separation follows speaker diarisation and happens before ASR. With the ASR model fine-tuned on separated data, cpWER-us was notably reduced.

1.2 Thesis Outline

The structure of this thesis is as follows:

- Chapter 2 reviews source separation models and introduces training methods, evaluation metrics, and existing datasets.
- Chapter 3 reviews SSL models for speech signals and explains how to use these models in source separation.
- Chapter 4 provides experimental details and results on conventional source separation tasks without target speakers.
- Chapter 5 presents the source selection methods and examines the separation performance in an automatic transcription system.
- Chapter 6 summarizes the thesis and discusses possible future directions.

Chapter 2

Source Separation

The task of single-channel speaker-independent source separation is defined in Equation 2.1 where N sources are combined into the mixture $y(t)$. Equation 2.2 shows a more realistic scenario where each source $x_s(t)$ is convolved with room impulse response (RIR) $h_s(t)$ and ambient noise $n(t)$ is added. In general, source separation can be regarded as a sequence-to-sequence problem with multiple targets.

$$y(t) = \sum_{s=1}^{N} x_s(t) \tag{2.1}$$

$$y(t) = \sum_{s=1}^{N} x_s(t) * h_s(t) + n(t) \tag{2.2}$$

In this chapter, basic building blocks of sequence-to-sequence models and how these blocks can be used in source separation are first introduced. Then the training method that solves the permutation problem is introduced and the T-F domain mixture invariant training (MixIT), a variant of the original time-domain MixIT is presented. Finally, the evaluation metrics and existing datasets are briefly explained.

2.1 Building Blocks

2.1.1 LSTM

Before the Transformer (Vaswani et al., 2017) appeared, recurrent neural networks (RNN) dominated sequence modelling tasks for a number of years because of the ability to handle variable input length and encode history information. The vanilla recurrent layer is shown in

Figure 2.1(a). Given the input feature sequence $\{\mathbf{x}_1,...,\mathbf{x}_T\}$, the model computes the hidden states sequence $\{\mathbf{h}_1,...,\mathbf{h}_T\}$ by taking the current frame and last hidden state as input and iterating following Equation 2.3. RNNs suffer from vanishing gradients especially when the hidden activation f is a Sigmoid function and it is difficult to maintain history information over many timesteps.

$$\mathbf{h}_t = f(\mathbf{W}^f\mathbf{x}_t + \mathbf{W}^r\mathbf{h}_{t-1} + \mathbf{b}) \tag{2.3}$$

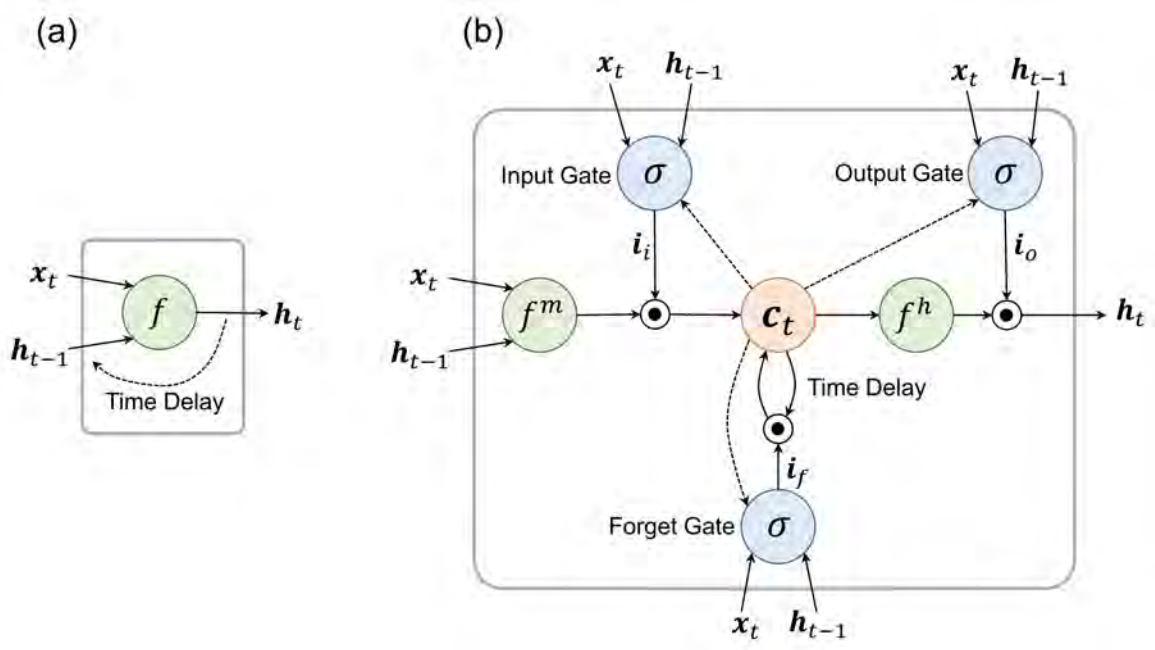


Fig. 2.1 (a) Recurrent Unit; (b) LSTM Unit.

The Long Short-Term Memory (LSTM) (Graves et al., 2013; Hochreiter and Schmidhuber, 1997) layer alleviates the vanishing gradient problem by introducing multiple gating functions and a memory cell to control the gradient flow and preserve long-range dependencies. As shown in Figure 2.1(b), there are three gates including the input gate $\mathbf{i}_i$, the output gate $\mathbf{i}_o$ and the forget gate $\mathbf{i}_f$. These gating functions are derived from the current input feature, the memory cell and the previous hidden state (Equations 2.4 2.5 2.6). The input gate determines which information can be incorporated at the current step; the forget gate decides how much information should be forgotten from the previous memory (Equation 2.7); the output gate controls the next hidden state (Equation 2.8). To utilise future information, the RNN or LSTM can be bidirectional (Graves et al., 2005) by incorporating both forward and backward hidden

states. Bidirectional-LSTM (BLSTM) commonly improves the performance compared with unidirectional LSTM but at the cost of being non-casual.

$$\mathbf{i}_f = \sigma(\mathbf{W}_f^f \mathbf{x}_t + \mathbf{W}_f^r \mathbf{h}_{t-1} + \mathbf{W}_f^m \mathbf{c}_{t-1} + \mathbf{b}_f) \tag{2.4}$$

$$\mathbf{i}_i = \sigma(\mathbf{W}_i^f \mathbf{x}_t + \mathbf{W}_i^r \mathbf{h}_{t-1} + \mathbf{W}_i^m \mathbf{c}_{t-1} + \mathbf{b}_i) \tag{2.5}$$

$$\mathbf{i}_o = \sigma(\mathbf{W}_o^f \mathbf{x}_t + \mathbf{W}_o^r \mathbf{h}_{t-1} + \mathbf{W}_o^m \mathbf{c}_t + \mathbf{b}_o) \tag{2.6}$$

$$\mathbf{c}_t = \mathbf{i}_f \odot \mathbf{c}_{t-1} + \mathbf{i}_i \odot f^m(\mathbf{W}_c^f \mathbf{x}_t + \mathbf{W}_c^r \mathbf{h}_{t-1} + \mathbf{b}_c) \tag{2.7}$$

$$\mathbf{h}_t = \mathbf{i}_o \odot f^h(\mathbf{c}_t) \tag{2.8}$$

where $\mathbf{W}$ and $\mathbf{b}$ are learnable weight matirx and bias. $\mathbf{x}_t$, $\mathbf{h}_t$ and $\mathbf{c}_t$ are input feature, hidden state and memory cell. $\odot$ stands for element-wise multiplication and σ is the Sigmoid function.

LSTM-based models including uPIT (Kolbæk et al., 2017) and TasNet (Luo and Mesgarani, 2018) achieved initial success of applying deep learning models in source separation. Recently, the BLSTM is still a popular downstream model to process SSL representations. For instance, Huang et al. (2022) compared various SSL models for source separation by concatenating and fine-tuning a three-layer BLSTM.

2.1.2 Transformer

The LSTM only partially solves gradient vanishing/explosion problems and the iterative process can not be implemented with parallel computing. The Transformer (Vaswani et al., 2017) integrates the attention mechanism and processes the whole sequence at once instead of sequentially. Therefore, it is better at extracting long-range dependencies and supports parallel computing for efficient training.

As shown in Figure 2.2, a Transformer layer is composed of a multi-head self-attention (MHSA) module, layer normalisation (Ba et al., 2016) and a feed-forward network (FFN). There is a residual connection (He et al., 2016) before each layer normalisation which adds the inputs of MHSA/FFN with their outputs. The residual connections promote training deep networks but require identical feature sizes throughout the network.

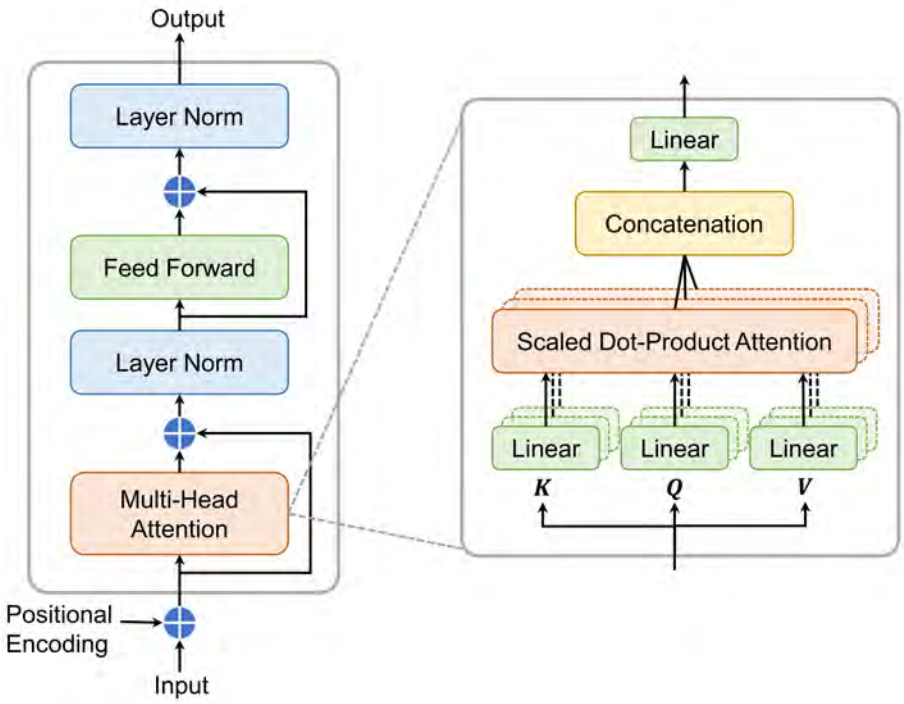


Fig. 2.2 Illustration of the Transformer layer (left) and detailed structure of multi-head self-attention (right).

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\text{head}_1,\text{head}_2,...,\text{head}_h)\mathbf{W}^o \tag{2.9}$$

$$\text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W}_i^Q,\mathbf{K}\mathbf{W}_i^K,\mathbf{V}\mathbf{W}_i^V) \tag{2.10}$$

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{2.11}$$

where d_k is the dimension of key and query. The dot product is scaled by $\sqrt{d_k}$ before passing through a Softmax function which is computed for each row.

The most critical component of the Transformer is the MHSA. It maps the hidden feature into a set of subspaces and allows attention at multiple locations. Figure 2.2 illustrates that the MHSA consists of several scaled dot-product attention units and their outputs are concatenated and then pass through a linear transformation which controls the feature size (Equation 2.9). The scaled dot-product attention (Equation 2.11) takes three matrices as input including queries $\mathbf{Q}$, keys $\mathbf{K}$ and values $\mathbf{V}$ which are analogous to an information retrieval system. The similarity between the query and the key decides which information should be

retrieved from the value. In the Transformer encoder, **Q**, **K**, **V** in Equation 2.10 are the same, so it is self-attention. With MHSA, the relation between any locations can be taken into account regardless of the distance, contributing to better modelling ability for long sequences.

Another important element is positional encoding which enables the model to utilise order information. The model can decide the absolute position or relative distance between inputs by learning the pattern of positional encoding. In practice, fixed sinusoidal embedding are added to the input (Equations 2.12 2.13).

$$\mathrm{PE}_{(pos,2i)} = \sin(pos/1000^{2i/d_{model}}) \tag{2.12}$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos(pos/1000^{2i/d_{model}}) \tag{2.13}$$

where *pos* and *i* represents position and element index respectively.

Because of its high efficiency and the capacity to model global dependencies, the Transformer has become one of the most popular architectures in NLP and speech processing. For source separation, Sepformer (Subakan et al., 2021) achieves state-of-the-art performance on WSJ0-2mix and WSJ0-3mix datasets by using Transformer blocks. Furthermore, the Transformer encoder is the basic structure in nearly all large-scale SSL models for speech signals including Wav2Vec2 (Baevski et al., 2020), Hubert (Hsu et al., 2021a) and WavLM (Chen et al., 2022a) etc.

2.1.3 Conformer

The Transformer can exploit long-range interactions, but for speech modelling, local context is equally important. A convolutional neural network (CNN) can extract local relationships and preserve translation equivariance by using shifting filters. Therefore, the convolutional layer and the Transformer are complementary, making their combination a natural choice. Wu et al. (2020) introduced the Lite-Transformer block that has two parallel branches: convolution and self-attention. Under a resource constrained setup, it significantly outperformed the standard Transformer on machine translation tasks. Speech-related tasks also enjoy the benefits of local and global contexts. The Conformer (Gulati et al., 2020) merges convolutions and self-attention without a multi-branch structure and achieved superior performance on ASR. As demonstrated in Figure 2.3, the Conformer block is similar to a Transformer layer except that 1) it has two FFN with half-step residual connections; 2) there is a convolution

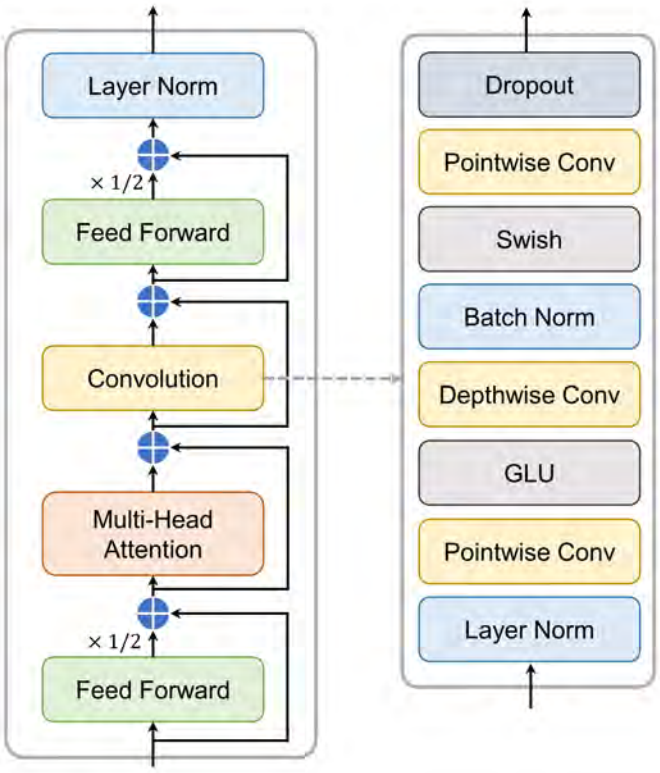


Fig. 2.3 Illustration of the Conformer module (left) and detailed structure of convolution block (right).

module after the MHSA.

As shown on the right of Figure 2.3, the main components of a convolution module include convolutional layers, normalisation, and activation functions. In detail, the input feature passes through a layer normalisation followed by a pointwise convolution (i.e. kernel size is 1×1) which effectively controls the number of channels. Then a gated linear unit (GLU) (Dauphin et al., 2017) activation controls the information flowing to the next layer. After the GLU, a depthwise convolution (Howard et al., 2017) is applied to capture fine-grained local features. It uses a single filter for each channel separately. A batch normalisation and a Swish activation function (Ramachandran et al., 2017) follow the depthwise convolution to promote the convergence speech. Finally, another pointwise convolution recovers the original number of channels and a Dropout layer (Srivastava et al., 2014) helps regularise the model. Since the convolution module stacks pointwise and depthwise convolution instead of using regular convolution, it is parameter-efficient.

Originally proposed for ASR, the Conformer has also shown promising performance in source separation. Chen et al. (2021b) showed that the Conformer was significantly superior to the BLSTM and slightly outperformed the Transformer in both single-channel and multi-channel source separation tasks. Moreover, the Conformer is an effective downstream model when appended to the WavLM (Chen et al., 2022a). Therefore, in this thesis, a single Conformer block was chosen as a task-specific layer for source separation (details in section 3.2 and 4.2).

2.2 Time-Frequency Domain Source Separation

Single-channel source separation is traditionally treated in the T-F domain. If the influence of phase is ignored, the task becomes the estimation of single-speaker magnitudes. However, instead of directly estimating magnitudes, it is commonly more effective to estimate masks (Erdogan et al., 2017). T-F domain source separation is illustrated in Figure 2.4 which assumes two output sources. The time-domain signal $\boldsymbol{y}$ is converted to the spectrogram $\boldsymbol{Y}$ by the Short Time Fourier Transformation (STFT) (Griffin and Lim, 1984). Then, the separation model takes the magnitude $|\boldsymbol{Y}|$ as input and predicts masks ($\boldsymbol{M}_1$ and $\boldsymbol{M}_2$) for each source. The estimated magnitudes ($|\boldsymbol{X}_1|$ and $|\boldsymbol{X}_2|$) are the element-wise products between the masks and the magnitude of the mixture $|\boldsymbol{Y}|$. Finally, time-domain signals ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) are recovered by the Inverse STFT (ISTFT). Since the model does not predict phase, the original phase θ_Y from the mixture is used for reconstruction.

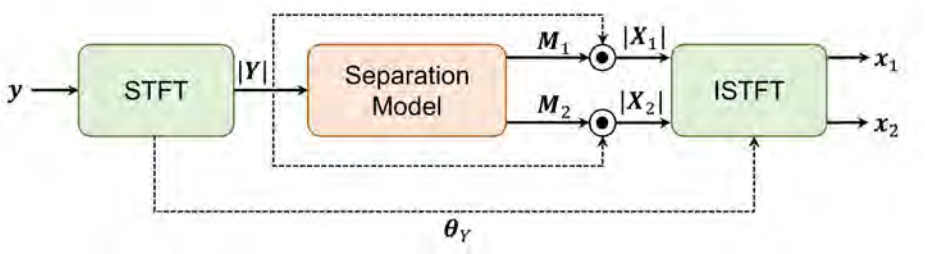


Fig. 2.4 Illustration of time-frequency domain source separation.

Being an intermediate target, the mask has several popular forms. The ideal binary mask (IBM) was first introduced as the goal of computational auditory scene analysis (Wang, 2005) which dramatically improved speech intelligibility. As the name implies, it restricts the mask value to $\{0, 1\}$ (Equation 2.14). The idea is to keep the T-F unit if the power of the target source is higher than the interference (i.e background noise or other speakers). In general, estimating the IBM is a binary classification problem.

$$\mathrm{M}_1^{IBM}(t,f) = \begin{cases} 1 & , |X_1(t,f)| > |X_2(t,f)| \\ 0 & , \text{otherwise} \end{cases} \tag{2.14}$$

The ideal Ratio mask (IRM) (Wang et al., 2014) is a soft version of IBM that is normally superior owing to more flexibility. As shown in Equation 2.15, IRM is the ratio between the target magnitude and the sum of magnitudes, so the mask values are constrained to $[0,1]$, and the sum of masks is 1 for each T-F unit. However, in practice, the sum of magnitudes is unavailable because $|Y(t,f)| \neq |X_1(t,f)| + |X_2(t,f)|$. Therefore, the ideal amplitude mask (IAM) (Kolbæk et al., 2017), also called the FFT-mask, is more feasible (Equation 2.16). With the IAM, the target magnitude can be accurately reconstructed from the input magnitude $|\boldsymbol{Y}|$.

$$\mathrm{M}_1^{IRM}(t,f) = \frac{|X_1(t,f)|}{|X_1(t,f)| + |X_2(t,f)|} \tag{2.15}$$

$$\mathrm{M}_1^{IAM}(t,f) = \frac{|X_1(t,f)|}{|Y(t,f)|} \tag{2.16}$$

The IAM and the IRM achieve the highest SNR when the phase of each source equals the phase of the mixture. However, in most cases, it is not a valid assumption, making them sub-optimal. A straightforward solution is complex masking, but it requires phase estimation which is empirically difficult. The ideal phase-sensitive mask (IPSM) (Erdogan et al., 2015) circumvents this challenge by using a real-valued mask and keeping the noisy phases. It can be seen from Equation 2.17 that the only difference between IAM and IPSM is a phase-correcting term (i.e. $cos(\theta_Y - \theta_{X_1})$) which has smaller value when the phase difference is large. Although the IPSM is unbounded, in practice most values fall into the range of 0 to 1. Therefore, it is feasible to generate masks with Softmax or Sigmoid functions.

$$\mathrm{M}_1^{IPSM}(t,f) = \mathrm{Re}(\frac{X_1(t,f)}{Y(t,f)}) = \frac{|X_1(t,f)|}{|Y(t,f)|}\mathrm{Re}(e^{i(\theta_Y - \theta_{X_1})}) = \frac{|X_1(t,f)|}{|Y(t,f)|}cos(\theta_Y - \theta_{X_1}) \tag{2.17}$$

where θ_Y and θ_{X_1} are the phases of the mixture $Y(t,f)$ and the target source $X_1(t,f)$ respectively.

Masks have no definition in the silence region where $X_1(t,f)$ and $Y(t,f)$ are zero. Therefore, the loss function is computed between the target magnitude $|\boldsymbol{X}_1|$ and the predicted magnitude $|\boldsymbol{Y}| \odot \boldsymbol{M}_1$ instead of directly between masks. For IPSM, the target simply becomes

$|\boldsymbol{X}_1|cos(\theta_Y - \theta_{X_1})$.

The configuration of STFT is another major factor of T-F domain source separation. To generate spectrograms, STFT crops the time-domain waveform into overlapped segments and computes the Fast Fourier Transform (FFT) separately. The ISTFT is implemented by the overlap-add method. The FFT window length (i.e. segment size) and overlap between adjacent frames determine the T-F resolution of spectrograms. In this thesis, the main consideration is to match the shape of the spectrogram with the size of SSL representations (details in section 3.2 and 4.2).

2.3 End-to-End Time-Domain Source Separation

Due to the decoupling of phase and magnitude, IPSM is not optimal. Moreover, spectrograms may not be the best feature representation. To overcome these setbacks, Luo and Mesgarani (2018) proposed to model the signal and perform separation in the time domain.

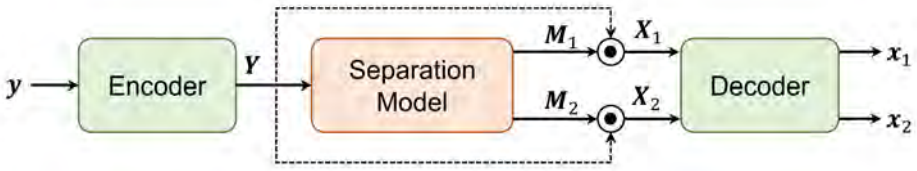


Fig. 2.5 Illustration of time-domain source separation.

Figure 2.5 illustrates the framework of time-domain source separation. Instead of directly predicting separated waveforms, it follows a similar flow as the masking-based method. An encoder converts the waveform $\boldsymbol{y}$ into a non-negative representation $\boldsymbol{Y}$ (similar to spectrogram) and after separation, a decoder resynthesises time-domain signals. The encoder is commonly a 1D convolutional layer followed by a ReLU activation function, and the decoder performs the inversion with a 1D transpose convolution. During training, the parameters of the encoder, the decoder, and the separation model are updated jointly. In general, time-domain source separation replaces the STFT with a learnable transformation and allows joint estimation of phase and magnitude.

Since the emergence of TasNet (Luo and Mesgarani, 2018), the time-domain framework has become the mainstream approach, and the majority of research has been focused on optimizing the architecture of the separation model. Initially, TasNet used a deep LSTM

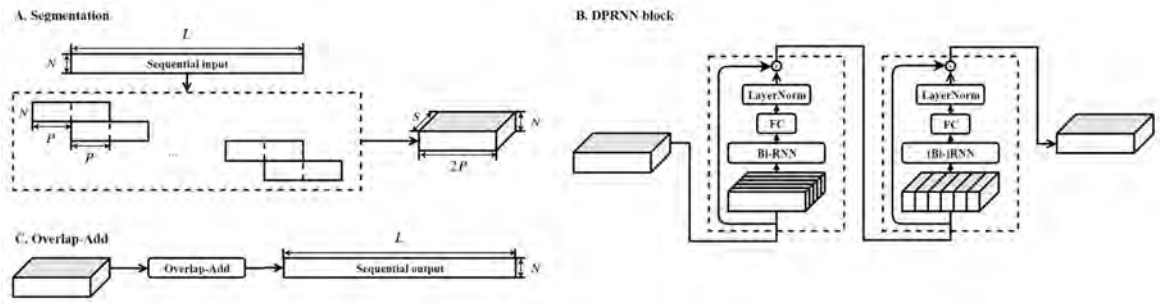


Fig. 2.6 Architecture and system flow of DPRNN, where L is the input sequence length; N is the feature size; $2P$ is the segment length; S is the number of segments. Image source: (Luo et al., 2020)

.

network. ConvTasNet (Luo and Mesgarani, 2019) improved its efficiency drastically by using a fully convolutional architecture that is mainly composed of dilated depthwise separable convolutions. To deal with the challenge of modelling extremely long sequences, Dual-path RNN (DPRNN) (Luo et al., 2020) introduced intra- and inter-chunk operations to utilise local and global information respectively. As shown in Figure 2.6 A, the sequential input of length L and feature size N is cropped into segments with half overlap. Then, these segments are re-organised into a 3D tensor. In a DPRNN block (Figure 2.6 B), the 3D tensor is first processed by a intra-chunk BLSTM layer followed by a linear layer. The intra-chunk operations are applied to each segment separately (i.e. along the dimension with length $2P$) so the output feature contains local information within individual segments. Then, the tensor is transposed so that the BLSTM and linear layers operate across all segments (i.e. inter-chunk operation), allowing feature extraction at the whole sequence level. Finally, the 3D tensor is converted back to sequence by the overlap-add method. The state-of-the-art model Sepformer (Subakan et al., 2021) uses the same dual-path structure except that BLSTM layers are replaced by Transformer layers.

Time-domain methods surpass T-F domain methods significantly when evaluated by SI-SNR. The main reason is that most time-domain models use negative SI-SNR as the loss function, whereas T-F domain models normally use mean square error (MSE) between spectrograms, so time-domain models have a natural advantage in signal-based evaluation that involves sample-wise comparisons. However, it is inadequate to conclude that time-domain methods are superior, because SI-SNR may not align with perceptual quality and the recognition accuracy of an ASR system (Chen et al., 2021b). Additionally, the filter size of 1D convolution in the encoder is much smaller than the window of STFT, resulting in very long feature sequences. As the result, time-domain models are not only computationally de-

manding but incompatible with SSL representations. Therefore, only T-F domain approaches were investigated in this thesis.

2.4 Training

2.4.1 Supervised Permutation Invariant Training (PIT)

Source separation is sometimes referred to as blind source separation, indicating that there is no target speaker. Hence, it is formulated as a multi-target regression problem. For the T-F domain methods, the model needs to predict a mask for each source. However, when calculating the loss function, it is unknown which mask corresponds to which target. This is defined as the label permutation problem.

The simple solution is to use a fixed permutation. For example, if the input signals are male-female mixtures, the first and second output sources can be specified to be corresponding to male and female speakers respectively. If the signal is not fully overlapped, the first output source can be paired with the main speaker and the second is left for the interfering speaker. As can be seen, fixed permutation fails if the assumption is not satisfied (e.g. the mixture contains speakers of same genders or the mixture is fully overlapped).

Permutation Invariant Training (PIT) (Kolbæk et al., 2017) was designed to carry out dynamic label assignments during training. It ignores the order of reference signals and output sources by treating them as sets and selects the minimum loss of all possible permutations. PIT can be inefficient if the model estimates many sources because $N!$ per-permutation losses need to be computed for N output sources, but in practice, the number of output sources is limited. In this thesis, PIT was mainly used for supervised training. As shown in Equation 2.18, the per-permutation loss is the sum of pairwise MSE between the predicted magnitude and the phase-sensitive target, and the final loss is the minimum per-permutation loss.

$$\mathcal{L}_{PIT} = \min_{\phi \in P} \sum_{(i,j) \in \phi} ||\boldsymbol{M}_i \odot |\boldsymbol{Y}| - |\boldsymbol{X}_j| cos(\theta_Y - \theta_{X_j})||_F^2 \tag{2.18}$$

where ϕ stands for a permutation; P is the set containing all possible permutations; $\boldsymbol{Y}$ is the T-F domain mixture; $\boldsymbol{M}_i$ is a output mask; $\boldsymbol{X}_j$ is a target spectrogram; θ_Y and θ_{X_j} are the phases of $\boldsymbol{Y}$ and $\boldsymbol{X}_j$; $||\cdot||_F$ is the Frobenius norm.

2.4.2 Unsupervised Mixture Invariant Traininig (MixIT)

PIT requires single-speaker ground-truth signals for supervised learning which are only provided by synthetic datasets. Consequently, the real-world performance depends on the compatibility between real and synthetic signals. However, creating realistic mixtures is challenging since it is difficult to decide the degree of reverberation, and the type of noise. Therefore, unsupervised approaches are desired to utilise real overlapped signals that lack clean references. To this end, unsupervised Mixture Invariant Training (MixIT) (Wisdom et al., 2020) was proposed.

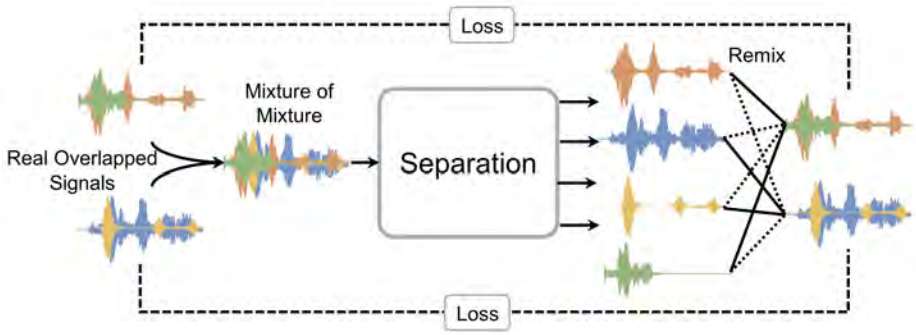


Fig. 2.7 Illustration of MixIT.

MixIT uses real mixtures as references instead of single-speaker segments and assumes the output sources can be remixed into the references. The detailed framework is shown in Figure 2.7. First, the mixture of mixture (MoM) is created by combining real overlapped signals. Then, the separation system takes the MoM as input and estimates several audio streams. To compute the loss, MixIT exhaustively searches for the best remix (i.e. the remix with the minimum error).

A problem with MixIT is over-separation. When the number of output streams is larger than the number of speakers, a single speaker's speech signal can be separated into different streams because the model never sees a clean reference during training, so it has no knowledge that an output stream should correspond to one speaker. Semi-supervised training mitigates this problem by interleaving supervised PIT with synthetic data and unsupervised MixIT with real data.

The original MixIT was used for time-domain source separation, in which case outputs can be remixed by adding samples of waveforms, but in the T-F domain, it is inappropriate to directly add magnitude because of the phase difference. Therefore, MixIT was adapted to T-F domain with PSM because the time domain remix is equivalent to the remix of

phase-sensitive targets (Equation 2.19). In this thesis, MixIT was mainly used to fine-tune the model on real meeting corpus. The loss function is shown in Equation 2.20 which has a similar form to PIT. The difference is that the prediction becomes a mixture ($\sum_{i\in\mathbf{I}} \boldsymbol{M}_i \odot |\boldsymbol{Y}|$) instead of a single spectra ($\boldsymbol{M}_i \odot |\boldsymbol{Y}|$).

$$\begin{cases} \boldsymbol{y} = \boldsymbol{x}_2 + \boldsymbol{x}_1 \\ |\boldsymbol{Y}| = |\boldsymbol{X}_1| cos(\theta_Y - \theta_{X_1}) + |\boldsymbol{X}_2| cos(\theta_Y - \theta_{X_2}) \end{cases} \tag{2.19}$$

$$\mathcal{L}_{MixIT} = \min_{\phi\in\mathcal{M}} \sum_{(\mathbf{I},j)\in\phi} ||(\sum_{i\in\mathbf{I}} \boldsymbol{M}_i \odot |\boldsymbol{Y}|) - |\boldsymbol{X}_j| cos(\theta_Y - \theta_{X_j})||_F^2 \tag{2.20}$$

where ϕ is a remix; $\mathcal{M}$ stands for all possible remix; $\mathbf{I}$ is a set of output streams that should be combined; $\boldsymbol{Y}$ is the T-F domain mixture; $\boldsymbol{M}_i$ is a output mask; $\boldsymbol{X}_j$ is a target spectrogram; θ_Y and θ_{X_j} are the phases of $\boldsymbol{Y}$ and $\boldsymbol{X}_j$; $\boldsymbol{y}$, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ stand for time-domain signals.

2.5 Evaluation Metrics

2.5.1 Scale-Invariant Signal-to-Noise Ratio

Scale-Invariant Signal-to-Noise Ratio (SI-SNR) (Luo and Mesgarani, 2018) defined by Equations 2.21, 2.22 and 2.23, is a widely used evaluation metric to compare the desired signal with the interference. For source separation, SI-SNR is based on the optimal label assignment (i.e. the permutation with the highest SI-SNR).

$$\mathbf{s}_{target} = \frac{\langle \hat{\mathbf{s}}, \mathbf{s} \rangle \mathbf{s}}{||\mathbf{s}||^2} \tag{2.21}$$

$$\mathbf{e}_{noise} = \hat{\mathbf{s}} - \mathbf{s}_{target} \tag{2.22}$$

$$\text{SI-SNR} = 10\, log_{10} \frac{||\mathbf{s}_{target}||^2}{||\mathbf{e}_{noise}||^2} \tag{2.23}$$

where $\hat{\mathbf{s}}$ and $\mathbf{s}$ is the predicted and ground-truth waveforms respectively. $\hat{\mathbf{s}}$ and $\mathbf{s}$ are normalised to zero-mean which guarantees scale-invariance.

SI-SNR has several drawbacks: 1) it performs sample-wise comparisons between the estimation and the target in the time domain (Equation 2.22) which may not align with human perception; 2) for sparsely overlapped speech, $\mathbf{s}_{target}$ is sometimes zero. In this case, SI-SNR is not well defined because it is expressed in decibels (Equation 2.23). To tackle the problem,

a small value is commonly added inside the logarithm. 3) SI-SNR can not be computed without clean targets. Therefore, in this thesis, SI-SNR was only considered for synthetic datasets.

2.5.2 Word Error Rate

Word Error Rate (WER), which is based on the Levenshtein distance between a hypothesis and a reference sequence, is a common measurement of ASR performance. It can be computed by Equation 2.24.

$$\text{WER} = \frac{I+S+D}{N} \times 100\% \tag{2.24}$$

where N is the total number of words; I, D, and S are the number of insertions, deletions, and substitutions respectively.

The WER is a feasible evaluation metric for source separation because an ASR model tends to perform better on speech with fewer overlaps which indirectly reflects the separation performance, and most real overlapped corpora provide ground-truth transcriptions. Moreover, improving the performance of an ASR system is an important application of source separation. In practice, an ASR system was adopted to transcribe the output streams of a source separation model and the minimum WER of all possible permutations was computed if there was no information about the target speaker. (details in section 4.1.2)

Besides comparing WER, researchers have also carried out listening tests to compare the perceptual quality (Sivaraman et al., 2022). However, such tests are time-consuming and hard to replicate, so WER was mainly used to compare the separation performance on real overlapped data.

2.6 Benchmark Datasets

The development of source separation is reflected in not only advanced model architectures but more challenging datasets. Initially, the only benchmark dataset was WSJ0-Mix (Hershey et al., 2016) which randomly selects utterances from Wall Street Journal corpus and creates mixtures with SNR between 0dB to 5dB. To improve realism, the WHAM! (Wichern et al., 2019) dataset includes ambient noises collected in restaurants, bars, coffee shops, etc., so the model should remove noise and separate speech signals simultaneously. The WHAMR! (Maciejewski et al., 2020) dataset extended WHAM! by introducing artificial

reverberation. LibriMix (Cosentino et al., 2020) is an expanded version of WHAM! which combined utterances from the LibriSpeech (Panayotov et al., 2015) corpus. It has significantly more unique speakers and distinct words than the WHAM! dataset, contributing to better generalisation ability of separation models. These datasets have 4 subsets which correspond to two modes (*max* and *min*) and two sample rates (8kHz and 16kHz). In the *min* mode, before mixing, the longer utterance is cropped to the length of the shorter utterance. In contrast, in the *max* mode, the shorter utterance is padded to the length of the longer utterance. The 16kHz and *max* mode subset of LibriMix was chosen for supervised training since SSL models were pre-trained on 16kHz signals and *max* mode allows non-overlapped speech to be incorporated. Furthermore, LibriMix was modified by using sensor noise instead of ambient noise and adding reverberations (details in section 4.1).

Real overlapped speech corpora have several notable differences with synthetic datasets: 1) the recordings are real conversations instead of read texts which involve frequent transitions between speakers and non-speech components like laughing and coughing; 2) the overlap ratio is commonly less than 20% (Çetin and Shriberg, 2006); 3) the speech is corrupted by real-world reverberation and noises. LibriCSS (Chen et al., 2020) is an evaluation-only dataset that simulates conversations by recording audio replays from loudspeakers placed in a meeting room. It provides sparsely overlapped utterances with overlap ratios ranging from 0% to 40% and their transcriptions. LibriCSS includes real sensor noise and room acoustics but the speech content is not actual conversation. The Augmented Multi-party Interaction (AMI) (Kraaij et al., 2005) corpus is a real-world meeting dataset created in several instrumented rooms. Speech signals were captured by a close-talk microphone and a distant microphone array synchronously. The recordings from the close-talk microphone contain clear foreground speech signals which can be used for alignment and identification of overlaps. The audio from the distant microphones are severely degraded by noise and overlaps which should be enhanced by the separation model. In this thesis, separation models were optimised both on the LibriCSS and the AMI corpus.

Chapter 3

Self-Supervised Learning for Speech Signals

SSL is a general approach to learn from unlabelled data through auxiliary tasks. It produces informative representations for downstream tasks and provides a good initialisation for further fine-tuning. Therefore, SSL significantly accelerates model development as fine-tuning requires low resources. Recently, researchers have been actively exploring SSL representations for speech signals intending to replicate its success in NLP. It is natural to adapt SSL from NLP to Speech because they are both sequences with temporal order. However, they use different representations. For language processing, words are represented by a finite number of tokens whereas, for speech, the number of features is close to infinite. This discrepancy leads to two types of auxiliary tasks: the generative task which requires regression of masked features and the discriminative task in which speech is discretised into a sequence of tokens by clustering or quantisation. In this chapter, recent SSL models are reviewed, including their architectures and the way they are pre-trained, and then how SSL representations can be applied to source separation is explained.

3.1 SSL Models

3.1.1 TERA

TERA (Liu et al., 2021) is a self-supervised Transformer encoder model trained on a generative auxiliary task. It introduced three alterations to the input spectrogram including time alteration, frequency alteration, and magnitude alteration. The training objective is to recover the unaltered features after these alterations. For time alternation, a random number of frames are masked by zero or replaced by segments of other frames. The key idea of time

alteration is that the reconstruction of missing segments encourages the model to extract contextual information from past and future frames. Frequency alteration randomly masks a single block of frequency bins for all time steps in one utterance, so the model learns to use information along the frequency axis. It was found that frequency alteration provides better speaker representations that benefit speaker recognition tasks. Magnitude alteration can be viewed as a data augmentation technique that adds random Gaussian noise to spectrograms. These alterations were dynamically combined through a stochastic policy during training to form the corrupted input features.

TERA used 80-dimensional log mel-spectrograms as the acoustic features. The mel-spectrogram was first altered and then fed into a 3-layer Transformer encoder followed by a 2-layer FFN which predicts the original mel-spectrogram. The network was updated by minimizing the L1 distance between the prediction and the unaltered mel-spectrogram. For downstream tasks, only the Transformer encoder was used for feature extraction and fine-tuning.

3.1.2 Wav2Vec2

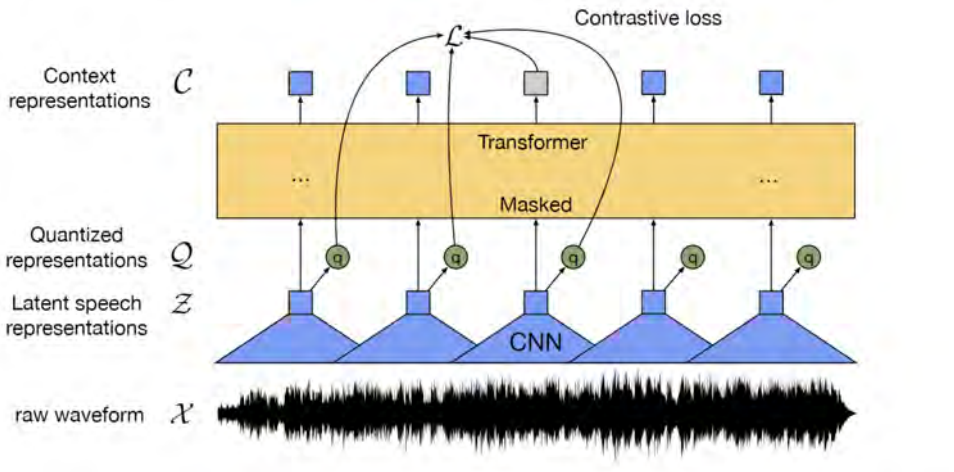


Fig. 3.1 Architecture of Wav2Vec2. Image source: (Baevski et al., 2020)

Wav2Vec2 (Baevski et al., 2020) has achieved state-of-the-art performance in ASR and only requires a limited amount of transcribed speech to fine-tune. As shown in Figure 3.1, Wav2Vec2 replaces traditional front-end transformations with a CNN that maps the raw waveforms **X** to latent representations **Z**. Then a Transformer encoder further extracts context representations **C** from latent representations. Similar to TERA, the latent features are

partially masked, but instead of directly reconstructing these features, Wav2Vec2 introduced a quantisation module that discretises latent features into more compact representations $\mathbf{Q}$ as training targets.

To make the quantisation module fully differentiable, Wav2Vev2 adopts product quantisation and Gumbel Softmax (Jang et al., 2017) to select discretised representations from multiple codebooks. The chosen representations from each codebook are concatenated and passed through a linear layer to obtain the final quantised feature.

The CNN, Transformer encoder, and quantisation module were trained jointly through a contrastive learning task. The model should discriminate the true quantised features at the masked time steps from a set of distractors. The contrastive loss is defined in Equation 3.1. $\mathbf{q}_t$ is the positive sample centered at the mask region that the model needs to identify. $\mathbf{Q}_t$ is a set that includes $\mathbf{q}_t$ and K negative samples randomly selected from other masked time steps. By minimizing the contrastive loss, the similarity between $\mathbf{q}_t$ and the output representation $\mathbf{c}_t$ was maximized.

$$\mathcal{L}_c = -log\frac{exp(sim(\mathbf{c}_t, \mathbf{q}_t)/k)}{\sum_{\tilde{\mathbf{q}}\in\mathbf{Q}_t} exp(sim(\mathbf{c}_t, \tilde{\mathbf{q}})/k)} \tag{3.1}$$

where *sim* is the cosine similarity and k is non-negative temperature.

To fully utilise the representations in the codebook, a diversity loss was incorporated which functions as regularisation. As shown in Equation 3.2, the diversity loss is the negative entropy of the distribution over the codebook entries, so it encourages equal probability for each entry and penalises the bias toward a limited number of representations. The final loss was the weighted sum of contrastive and diversity loss.

$$\mathcal{L}_d = \frac{1}{GV}\sum_{g=1}^{G}\sum_{v=1}^{V}\overline{p}_{g,v}log\overline{p}_{g,v} \tag{3.2}$$

where G is the number of codebooks and V is the number of entries in each codebook.

Wav2Vec2 has a larger Transformer encoder than TERA with 12 layers. The CNN feature extractor includes 7 convolutional layers with strides of (5, 2, 2, 2, 2, 2, 2) and filter sizes of (10, 3, 3, 3, 2, 2), so it downsamples the 16kHz waveform to a 50Hz feature sequence. For each frame, the receptive field is around 25ms with a stride of 20ms.

3.1.3 HuBert and Its Variants

HuBert (Hsu et al., 2021a) has the same model architecture as Wav2Vec2 and also masked the latent features during training (i.e. the output of the CNN), but rather than quantisation, HuBert adopted an offline clustering approach to generate discrete labels for representation learning.

Let $\mathbf{Z}$ denote the latent representation sequence and $\tilde{\mathbf{Z}}$ is the corrupted representation where frames with indices $t \in M$ are masked. The Transformer encoder takes $\tilde{\mathbf{Z}}$ as input and estimates distributions of pseudo-labels at each time step (i.e. $p(\boldsymbol{y}_t|\tilde{\mathbf{Z}},t)$). The cross-entropy loss was only computed over the masked region (Equation 3.3), guiding the model to learn both local acoustic content and global correlations. As shown in Equation 3.4, the distribution over target label is generated via the Softmax function where $\boldsymbol{y}_t^{(i)}$ is the embedding of the target label (i.e. the corresponding cluster centre); $\boldsymbol{c}_t$ denotes the output representation of the Transformer; $\mathbf{A}$ is a projection matrix and τ is set to 0.1.

$$\mathcal{L}_m = -\sum_{t \in M} log\ p(\boldsymbol{y}_t|\tilde{\mathbf{Z}},t) \tag{3.3}$$

$$p(\boldsymbol{y}_t^{(i)}|\tilde{\mathbf{Z}},t) = \frac{exp(sim(\mathbf{A}\boldsymbol{c}_t,\boldsymbol{y}_t^{(i)})/\tau)}{\sum_{j=1}^{N} exp(sim(\mathbf{A}\boldsymbol{c}_t,\boldsymbol{y}_t^{(j)})/\tau)} \tag{3.4}$$

To improve the quality of target labels, HuBert adopted a bootstrap approach for label refinement. At the first iteration, the clustering step is based on the 39-dimensional Mel-frequency cepstral coefficients (MFCCs) including the differential terms. It is expected that higher-dimensional embeddings learned from data are better representations than human-designed acoustic features like MFCCs. Therefore, in future iterations, clusters are iteratively created by using intermediate embeddings from HuBert.

HuBert primarily focuses on modelling the speech content rather than the speaker information. **UniSpeech-SAT** (Chen et al., 2022b) was proposed as a variant of HuBert to facilitate the extraction of speaker identity and thus promote speaker-related downstream tasks like speaker diarisation and verification. On top of the cross-entropy loss, the utterance-wise contrastive loss was incorporated which has the same form as Equation 3.1. The difference is that distractors were sampled from masked representations in the same batch rather than in the same utterance. Since it was assumed that the utterances in a batch correspond to different speakers, the model should recognise the representations of the correct speaker, resulting in better speaker representations. Furthermore, UniSpeech-SAT enjoyed the benefits of

utterance mixing augmentation and a large-scale dataset containing 94k hours of speech from multiple sources. The utterance mixing simulated a multi-speaker scenario by combining the main utterance with a randomly selected segment from another speaker. The overlap ratio was kept under 50% to make sure the model learns to extract information from the main speaker under the interference of another speaker.

WavLM (Chen et al., 2022a) used the same training objective as the HuBert. To improve the Transformer encoder, gated relative position bias is integrated into the self-attention mechanism. The gates are computed based on the current content which dynamically adjusts the position embedding. Therefore, the gated relative position bias not only represents the offset between key and query but considers the content difference. For example, the position bias encodes different information if the current frame contains different numbers of speakers. WavLM was trained on the same dataset as UniSpeeech-SAT. Besides mixing utterances, WavLM also augmented speech signals by adding environmental noises, leading to the speech denoising task in the representation space.

3.2 SSL Representations for Source Separation

SSL models can be incorporated into an ASR framework by attaching a linear layer on top of the Transformer encoder to map the SSL representations into class labels and fine-tuning the base model with the linear layer. For source separation, there are two more steps before feeding the SSL representations into the downstream model which are multi-layer feature merging and stride matching.

SSL representations from different layers encode different information. For example, shallow layers may capture low-level acoustic information while deeper layers can learn high-level semantic information. As source separation requires the estimation of fine-grained T-F masks, features from shallow layers tend to be more effective. Huang et al. (2022) showed that for source separation and speech enhancement, representations from the first layer of HuBert are significantly better than representations from deeper layers and the weighted sum of multi-layer features leads to the best performance. Therefore, features including the output of every Transformer layer and the output of the CNN encoder are fused using weighted sum (Figure 3.2). A single static weight that is learned during training is used for each layer. All the weights are normalised by a Softmax activation to ensure that they are non-negative and sum-to-one. Note that there are more sophisticated ways to dynamically merge features by utilising self-attention or gating mechanism (Sun et al., 2021),

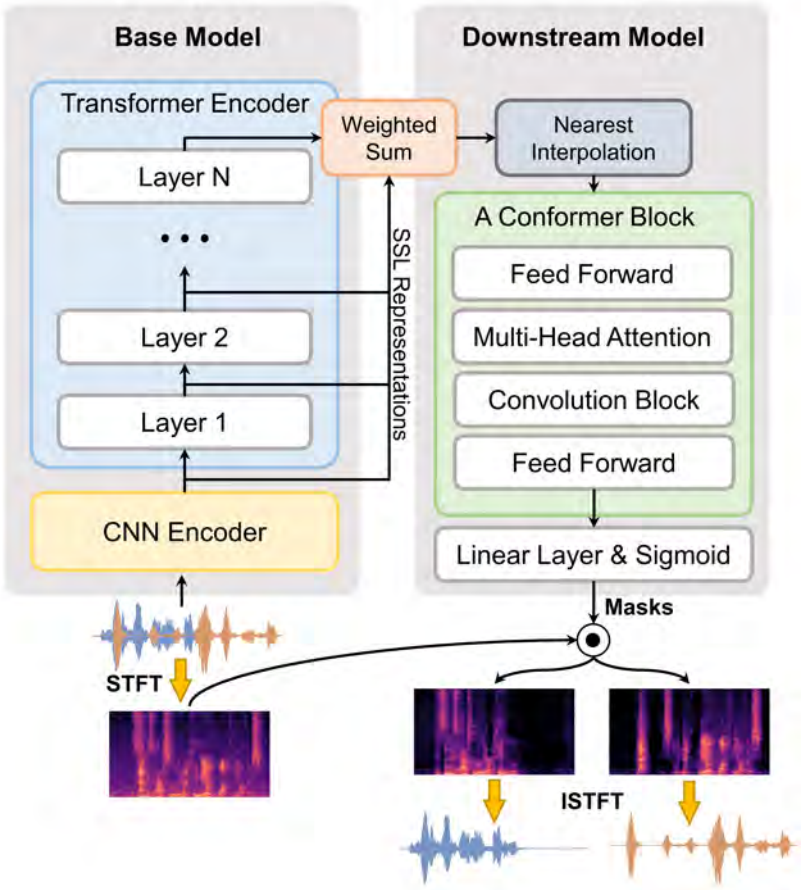


Fig. 3.2 Framework of T-F domain source separation using SSL representations.

but these methods are commonly applied for a limited number of representations and will introduce large computational overheads if used to combine more than 10 feature vectors. The naive weighted sum has negligible parameters and can be interpreted as a measurement of importance for SSL representations. Moreover, all the investigated SSL models have the same hidden size, so they can be directly combined.

The STFT typically has a window size of 25 ms and a stride of 10 ms. Latent features from the CNN encoder have the same window size but the stride is 20 ms. Therefore, spectrograms have twice the length compared with SSL representations. Therefore, before the downstream model, the nearest interpolation is applied to increase temporal resolution, which is equivalent to twice replicating SSL representations at each time step.

In general, SSL representations extracted from the based model are fed into a Conformer block after merging and stride matching. Then outputs of the Conformer are projected to

T-F masks with a linear layer followed by Sigmoid activation. Finally, separated signals are reconstructed from the masked spectrograms and the phases of the mixture. Compared to previous work (Chen et al., 2022a) that used an 18-layer Conformer on top of WavLM, in this thesis, the downstream model was made as small as possible, so training was more efficient and SSL representations played a more important role. Furthermore, unlike ASR, existing work for source separation only used SSL representations to substitute conventional acoustic features rather than further fine-tuning the SSL models. In this thesis, scenarios with the base model frozen or unfrozen were both investigated.

Chapter 4

Experiments for Blind Source Separation

In this chapter, multiple aspects of source separation models were examined using both simulated and real datasets. Section 4.1 and section 4.2 provide details about the construction of training datasets, evaluation methods and model configurations. In section 4.3, basic setups like the type of masking were investigated and various SSL models were compared. In section 4.4, the effectiveness of the separation model on real meeting corpus was verified and the performance was further optimized with unsupervised MixIT.

4.1 Data Synthesis and Evaluation

4.1.1 Reverberated LibriMix

Room		Mic. Center		Sources		T60
L	$\mathcal{U}(5,10)$	L	$0.5 \times L_{room} + \mathcal{U}(-0.5,0.5)$	H	$\mathcal{U}(0.9,1.8)$	
W	$\mathcal{U}(5,10)$	W	$0.5 \times W_{room} + \mathcal{U}(-0.5,0.5)$	distance	$\mathcal{U}(0.66,2)$	$\mathcal{U}(0.1,0.6)$
H	$\mathcal{U}(3,4)$	H	$\mathcal{U}(0.9,1.8)$	θ	$\mathcal{U}(0,2\pi)$	

Table 4.1 Hyperparameters for generating RIR. $\mathcal{U}$ stands for uniform distribution. T60 denotes reverberation time in seconds. θ is the horizontal angle of the sources to the microphone. All other parameters' units are meters. These parameters are referred from WHAMR! dataset (Maciejewski et al., 2020).

As mentioned in section 2.6, the *max* version of LibriMix (Cosentino et al., 2020) was adopted for supervised training, but instead of using the original mixtures corrupted by limited types of ambient noises, a new reverberated LibriMix was created following the approach in (Chen et al., 2021b) to simulate a room environment. As shown in Table 4.1,

parameters such as room dimensions and the microphone position were first sampled from uniform distributions. Then, the image method (Allen and Berkley, 1979) was used to generate artificial room impulse responses (RIR) for each speaker. After convolving clean signals with RIRs, these signals were recombined and isotropic noise (Habets and Gannot, 2007) was added with SNR ranging from 10dB to 20dB. The final dataset contains 209 hours of training mixtures sampled at 16kHz. Note that the reverberated single speaker signals were used as targets, so the model can focus on separation rather than performing separation and dereverberation simultaneously. Further details about all datasets used in this thesis are provided in the Appendix A.

4.1.2 Evaluation

Evaluations were based on the permutation invariant SI-SNR and WER introduced in section 2.5.1 and section 2.5.2. Evaluations should be realistic, so non-fully overlapped mixtures between two speakers were mainly considered. To enable both signal-based and ASR-based evaluations, three datasets were chosen including the synthetic SparseLibriMix dataset (Cosentino et al., 2020), the simulated LibriCSS dataset (Chen et al., 2020) and the real AMI corpus (Kraaij et al., 2005).

The SparseLibriMix provides mixtures at 6 overlap ratios including 0%, 20%, 40%, 60%, 80% and 100%. Each mixture contains multiple utterances from two speakers. As in section 4.1.1, the SparseLibriMix was modified by adding reverberation and noise. Since the SparseLibriMix provides ground-truths signals, SI-SNR was used to measure the sample-wise similarity between two predictions and two targets.

LibriCSS and AMI datasets were introduced in section 2.6. Since they do not have clean references, the utterance-wise evaluation was employed. First, continuous recordings were cropped into utterances according to ground-truth boundaries. Then these utterances were passed through the separation model and the outputs were transcribed resulting in several hypotheses. For each utterance, the hypothesis with the lowest WER was selected to compute the final WER. As AMI is much noisier than LibriCSS, different ASR models were adopted. For LibriCSS, a Wav2Vec2-Large model [1] fine-tuned on 960 hours of LibriSpeech dataset (Panayotov et al., 2015) was selected. For AMI, a Wav2Vec2-Robust [2] fine-tuned on

[1] https://huggingface.co/facebook/wav2vec2-large-960h-lv60-self

[2] https://huggingface.co/facebook/wav2vec2-large-robust-ft-swbd-300h Although Switchboard is an 8kHz dataset that is resampled to 16kHz before fine-tuning the Wav2Vec2, on the AMI corpus, it still achieves lower WER compared with the model fine-tuned on LibriSpeech dataset.

the noisy Switchboard dataset (Godfrey et al., 1992) was used. Note that in this chapter, ASR models fine-tuned on AMI were not considered because models trained on non-overlapped speech are more sensitive to interfering speakers and therefore better reflect separation performance. Moreover, it simulated a real-world situation where training transcriptions in the target domain are not available. For a more thorough evaluation, AMI test set were divided according to the percentage of overlaps, but unlike LibriCSS which has fixed overlap ratios and provides accurate utterance boundaries, overlaps in AMI are not controlled and the alignment file is not 100% precise, so 4 AMI subsets with rough overlap intervals ($0\% \sim 10\%, 10\% \sim 20\%, 20\% \sim 30\%, 30\% \sim 40\%$) were created.

4.2 Model Details

4.2.1 Architecture

For the base model, WavLM was selected for most experiments since it was pre-trained on a large-scale dataset augmented by overlaps. WavLM had the same convolutional feature extractor as Wav2Vec2 described in section 3.1.2 and its encoder consisted of 12 Transformer layers with hidden states of 768 dimensions, 8 attention heads, and 3072-dimensional FFN, yielding 90.2M parameters. The downstream model was a single Conformer block with hidden states of 256 dimensions, 4 attention heads, and 1024-dimensional FFN. The Dropout layer was deactivated and the kernel size of the convolution layer was 33. This downstream model had 1.8M parameters, smaller than all downstream models used in previous works.

The front-end features were magnitude spectrograms generated by STFT with a 400-point Hann window, a stride of 160 points, and a filter size of 512 samples. The symmetrical components were discarded, so each input frame had 257 Fourier coefficients. Masks with the same shape as input spectrograms were estimated. The numbers of output masks were 2 and 4 for supervised PIT and unsupervised MixIT respectively.

4.2.2 Training Configuration

The models were trained through a two-phase scheme. In the first phase, only the Conformer block was updated by the AdamW optimiser (Loshchilov and Hutter, 2018) with the weight decay of 1e-2. A learning rate schedule was used where the linear warm-up step was set to 5000 and the training iteration was 100,000. The peak learning rate was 2e-5 and decayed to 0 linearly after warm-up steps. Each batch contained 24 randomly cropped segments with a length of 4 seconds and gradients were accumulated every 4 iterations to simulate a

larger batch of 96 segments. Ramdom cropping ensured that training batches cover fully overlapped, partially overlapped, and non-overlapped data. In the second phase, the WavLM was unfrozen and fine-tuned jointly with the Conformer. The hyperparameters were the same as the first phase except that the model was trained with a smaller learning rate for fewer iterations (i.e. peak learning rate of 1e-5 and training iteration of 80,000). Note that for efficiency, in some experiments, only the performance after phase one was compared.

4.3 Experiments on Simulated Dataset

In this section, evaluations were performed on SparseLibriMix and LibriCSS datasets. The WavLM-based separation models were mainly used and trained through supervised PIT on reverberated LibriMix dataset. In section 4.3.1 and section 4.3.2, only the downstream model (i.e. the Conformer block) was trained. In the following sections, the fine-tuning of base models was explored and different SSL representations were compared.

4.3.1 Amplitude Mask and Phase Sensitive Mask

To develop an effective source separation model, the type of target should be decided first. IPSM is better than IAM since it takes phase difference into consideration (section 2.2). As shown in Table 4.2, IAM and IPSM represent the upper bound performance when the model perfectly estimated the targets but the phase of the mixture was used in ISTFT. It can be seen that on the SparseLibriMix dataset the SI-SNR is consistently higher for IPSM than IAM. Consequently, the separation model that predicts PSM (WavLM&PSM) has higher SI-SNR than the model with AM as targets (WavLM&AM). Furthermore, the difficulty of the separation task increases with the overlap ratio, so the SI-SNR decrease. However, zero-overlap is an exception with low SI-SNR. The possible reason is that the model rarely saw non-overlapped data from two speakers and it might be difficult to identify speaker

Method	**Overlap ratio in %** 0	20	40	60	80	100	AVG
WavLM & AM	5.56	7.91	7.77	6.92	6.76	6.18	6.85
WavLM & PSM	6.49	8.40	8.06	7.21	7.00	6.35	7.25
IAM	22.39	15.89	13.83	12.57	12.08	11.70	14.74
IPSM	23.65	18.03	16.15	14.97	14.52	14.16	16.91

Table 4.2 The comparison between PSM and AM on the SparseLibriMix dataset measured by SI-SNR ↑ (dB).

transitions without overlaps. In the later sections, it was found that fine-tuning the base model can solve this problem.

Method	Overlap ratio in %						
	0L	0S	10	20	30	40	AVG
No separation	2.9	3.1	9.2	16.4	24.3	32.3	14.7
WavLM & AM	2.8	2.9	5.8	8.6	11.6	13.9	7.6
WavLM & PSM	2.9	3.0	5.6	8.4	11.2	13.3	7.4

Table 4.3 The comparison between PSM and AM on the LibirCSS dataset measured by WER $\downarrow$ (%). 0L/0S means 0% overlap with long/short inter-utterance silence.

As shown in Table 4.3, without separation, the ASR performance was severely degraded by overlaps. The separation models significantly decreased WERs, especially for large overlap ratios. Moreover, PSM contributed to lower WERs and the performance gap between AM and PSM is more obvious for large overlaps as more phases belong to the overlapped region, making phase difference information more crucial. However, for zero overlaps (0L and 0S), AM is slightly better than PSM because in this case, the clean phase of a single speaker is available so PSM is not beneficial. Note that each segment in LibriCSS includes a full utterance of the main speaker and some interfering speech components from another speaker whereas in SparseLibriMix each segment contains multiple utterances from two speakers. Therefore, when there is no overlap, the model only needs to output the original input for LibriCSS but for SparseLibriMix the model still needs to divide the utterances of two speakers.

In general, IPSM is a more effective separation target than IAM and it does not pose any additional challenge to the training process. Furthermore, as explained in section 2.4.2, PSM is compatible with unsupervised MixIT. Therefore, in the following experiments, PSM was used by default.

4.3.2 Input Features

Traditional T-F domain source separation uses spectrograms as input features. To incorporate SSL models, the naive approach is to concatenate spectrograms with SSL representations along the feature dimension (Chen et al., 2022a). Hung et al. (2022) verified that for source separation and speech enhancement using the concatenation of spectra and SSL features as input is better than using SSL representations solely because spectrograms provided

Method	Overlap ratio in %						
	0	20	40	60	80	100	AVG
Spectrogram	5.68	4.83	4.22	3.48	3.08	2.80	4.02
WavLM	6.49	8.40	8.06	7.21	7.00	6.35	7.25
WavLM & Spectrogram	4.99	7.66	7.59	6.86	6.70	6.09	6.65

Table 4.4 The comparison between input features on the SparseLibriMix dataset measured by SI-SNR ↑ (dB). 'WavLM & Spectrogram' represents the concatenation of WavLM's features and spectrogram.

Method	Overlap ratio in %						
	0L	0S	10	20	30	40	AVG
Spectrogram	2.7	2.9	7.9	14.0	20.6	26.2	12.4
WavLM	2.9	3.0	5.6	8.4	11.2	13.3	7.4
WavLM & Spectrogram	2.8	3.0	5.8	8.9	12.2	14.3	7.8

Table 4.5 The comparison between input features on the LibriCSS dataset measured by WER ↓ (%). 'WavLM & Spectrogram' represents the concatenation of WavLM's features and spectrogram.

fine-grained information.

As shown in Table 4.4 and Table 4.5, three types of input features were compared including 257-dimensional spectrogram, 768-dimensional SSL representation (i.e. the weighted sum of multi-layer features) and their concatenation resulting in 1025-dimensional feature. In the experiments, the WavLM was frozen and the numbers of trainable parameters were similar. In contrast to previous work, it was found that spectrogram harmed the performance since 'WavLM & Spectrogram' led to lower SI-SNR on SparseLibriMix and higher WER on LibriCSS compared with 'WavLM' that only exploited SSL representations. The main reason is that the downstream model is very small and thus lacks the capability to process raw acoustic features. It is evident that the performance is poor on both datasets when only spectrograms were used as input. Furthermore, since features from WavLM were from multiple layers, low-level information can be extracted from shallow layers so the spectrogram may be redundant. Figure 4.1 illustrates the weights that correspond to each layer of WavLM. It can be seen that higher weights were assigned to shallow layers, indicating low-level acoustic information is more important than high-level semantic information since the model need to predict fine-grained masks rather than discrete labels. Without the spectrogram as input, the model paid more attention to low-level features which are easier to process than

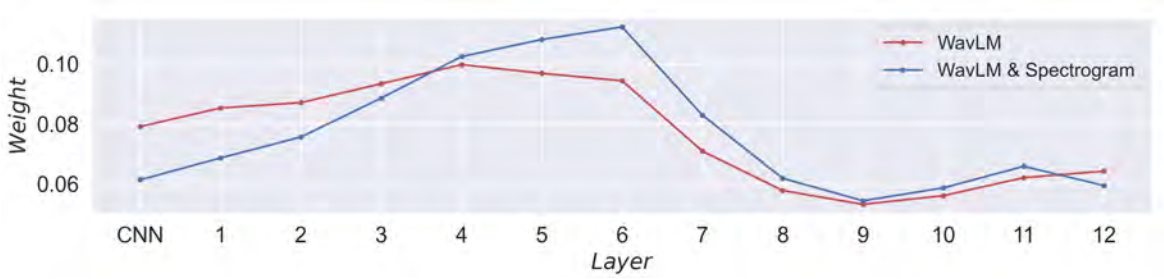


Fig. 4.1 Weight analysis for different input features.

spectrograms. In the remaining experiments, the input of the Conformer block only consisted of SSL representations because of the superior performance and more direct comparisons between SSL models.

4.3.3 Fine-Tuning

For source separation, previous work has seldomly unfrozen the base model as they used large downstream models. However, since a tiny downstream model was employed, the performance was constrained by the limited number of trainable parameters. Hence, to further utilise the pre-trained WavLM, it was fine-tuned with the Conformer block following the first phase where only the Conformer block was trained (details in section 4.2.2). Tables 4.6 and 4.7 show the performance before and after WavLM was fine-tuned. Comparing the first and the last row, the average SI-SNR increases by 2.84dB on SparseLibriMix, and the average WER decreases by 0.7% on LibriCSS. The improvement is more obvious in terms of SI-SNR as sample-level separation is more challenging than separating speech content so a large number of tunable parameters is more critical.

Method	**Overlap ratio in %**						
	0	20	40	60	80	100	AVG
Freeze WavLM	6.49	8.40	8.06	7.21	7.00	6.35	7.25
Unfreeze WavLM	13.89	11.08	9.62	8.21	7.70	7.32	9.64
Freeze->Unfreeze WavLM	13.75	11.85	10.12	8.74	8.22	7.88	10.09

Table 4.6 Evaluation of fine-tuning methods on the SparseLibriMix dataset measured by SI-SNR ↑ (dB). 'Freeze WavLM' means only fine-tuning the downstream model (phase 1); 'Unfreeze WavLM' means directly fine-tuning the whole model; 'Freeze->Unfreeze WavLM' is the two-phase training scheme.

To verify the effectiveness of the two-phase training scheme, the performance where the whole model was unfrozen and fine-tuned from the beginning (the second rows of Tables 4.6

Method	Overlap ratio in %						
	0L	0S	10	20	30	40	AVG
Freeze WavLM	2.9	3.0	5.6	8.4	11.2	13.3	7.4
Unfreeze WavLM	2.9	3.0	6.1	9.4	12.4	15.1	8.2
Freeze->Unfreeze WavLM	2.8	2.9	5.1	7.5	9.8	12.0	6.7

Table 4.7 Evaluation of fine-tuning methods on the LibriCSS dataset measured by WER ↓ (%). 'Freeze WavLM' means only fine-tuning the downstream model (phase 1); 'Unfreeze WavLM' means directly fine-tuning the whole model; 'Freeze->Unfreeze WavLM' is the two-phase training scheme.

and 4.7) was evaluated. The training configuration was the same as phase one. Although on the SparseLibriMix, the average SI-SNR (9.64dB) is notably better than freezing the WavLM and only slightly worse than the two-phase training approach, its WERs on LibriCSS are highest. The main reasons are overfitting and the failure to exploit SSL representations. The type of noise and reverberation are the same for LibriMix and SparseLibriMix so when the whole model was trained directly, the model overfitted this specific kind of data and generalised poorly to the out-of-domain LibriCSS dataset. Additionally, the weights assigned to WavLM's layers tend to be more uniformly distributed (blue line in Figure 4.2) which is noticeably different from the experiment where WavLM was frozen (red line), indicating that the base model extracted significantly different features compared with original SSL representations.

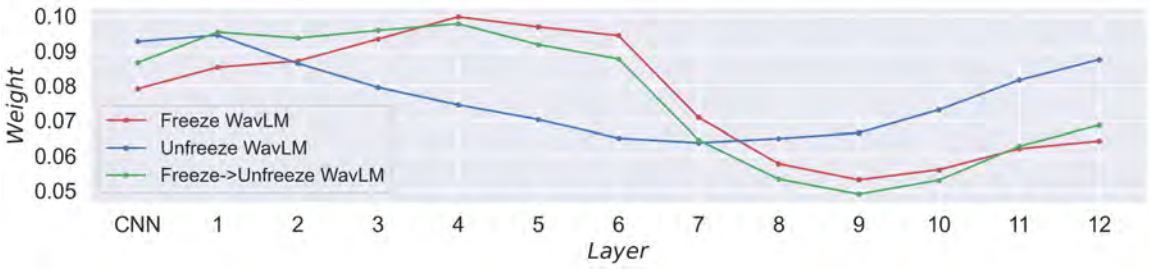


Fig. 4.2 Weight analysis for fine-tuning methods.

The two-phase training scheme improved the model's generalisation on out-of-domain datasets by better utilizing SSL representations that contain information from diverse speech corpora. Specifically, the two phases can be regarded as exploiting and adjusting SSL representations respectively. Across these two phases, the weight for each layer remained roughly the same except that shallow layers were assigned slightly higher weights after fine-tuning (green line).

4.3.4 Comparisons between SSL Representations

Model	Architecture	Parameters	Input	Dataset
TERA	3-Transformer	21.3M	Mel-spectrogram	LS 960hr
Wav2Vec2	7-CNN & 12-Transformer	90.2M	Waveform	LS 960hr
UniSpeech-SAT	7-CNN & 12-Transformer	90.2M	Waveform	Mix 94khr
WavLM	7-CNN & 12-Transformer	90.2M	Waveform	Mix 94khr

Table 4.8 Details of SSL models (Baevski et al., 2020; Chen et al., 2022a,b; Liu et al., 2021). LS 960hr: LibriSpeech dataset (Panayotov et al., 2015). Mix 94khr: LibriLight (Kahn et al., 2020), VoxPopuli (Wang et al., 2021) and GigaSpeech datasets (Chen et al., 2021a)

Four SSL models introduced in section 3.1 were compared. The model details are provided in Table 4.8. Except for TERA, the other models have similar architectures and numbers of parameters. Tables 4.9 and 4.10 show the comparisons between SSL models with the base model frozen or fine-tuned. For every subset of SparseLibriMix, all SSL representations achieved higher SI-SNR after phase one compared with the spectrogram. On the LibriCSS dataset, when overlap ratios are greater than zero, WERs are lower with SSL representations than with the spectrogram. This proves that SSL representations are more informative for source separation than raw acoustic features and easier to be processed by a small downstream model.

	Overlap ratio in %						
Method	0	20	40	60	80	100	AVG
Spectrogram	5.68	4.83	4.22	3.48	3.08	2.80	4.02
Phase 1: train the downstream model.							
TERA	8.17	7.34	6.38	5.43	4.80	4.63	6.13
Wav2Vec2	**9.81**	7.83	6.78	5.67	5.15	4.71	6.66
UniSpeech-SAT	8.18	**8.89**	7.82	7.08	6.61	6.07	**7.44**
WavLM	6.49	8.40	**8.06**	**7.21**	**7.00**	**6.35**	7.25
Phase 2: fine-tune the base model.							
TERA	10.67	8.93	7.67	6.67	6.02	5.71	7.61
Wav2Vec2	14.19	10.37	8.62	7.32	6.78	6.36	8.94
UniSpeech-SAT	**14.79**	11.80	9.88	8.59	7.94	7.55	**10.09**
WavLM	13.75	**11.85**	**10.12**	**8.74**	**8.22**	**7.88**	**10.09**

Table 4.9 The comparison between SSL models on SparseLibriMix dataset measured by SI-SNR ↑ (dB).

Method	Overlap ratio in %						
	0L	0S	10	20	30	40	AVG
Spectrogram	2.7	2.9	7.9	14.0	20.6	26.2	12.4
Phase 1: train the downstream model.							
TERA	**2.7**	**3.0**	7.4	12.8	18.3	22.7	11.2
Wav2Vec2	2.8	3.1	7.7	13.0	18.6	23.0	11.4
UniSpeech-SAT	2.9	**3.0**	5.7	9.0	12.1	14.7	7.9
WavLM	2.9	**3.0**	**5.6**	**8.4**	**11.2**	**13.3**	**7.4**
Phase 2: fine-tune the base model.							
TERA	**2.8**	3.1	7.2	12.1	16.5	21.0	10.5
Wav2Vec2	**2.8**	3.1	7.0	11.0	15.7	19.1	9.8
UniSpeech-SAT	2.9	**2.9**	**5.0**	7.6	9.9	**12.0**	**6.7**
WavLM	**2.8**	**2.9**	5.1	**7.5**	**9.8**	**12.0**	**6.7**

Table 4.10 The comparison between SSL models on LibriCSS dataset measured by WER ↓ (%).

In detail, when the downstream model was trained, TERA has the lowest average SI-SNR on SparseLibirMix owing to the small model size. WavLM and UniSpeech-SAT are significantly better than TERA and Wav2Vec2 since they were trained on larger datasets augmented by overlaps. Figure 4.3 illustrates the weight analysis for different SSL models. The model pre-trained on clean audio (i.e. Wav2Vec2) demonstrates a different pattern compared with models trained on data with overlaps (i.e. UniSpeech-SAT and WavLM), especially for the middle layers. For TERA, the weight increases as the layer gets deeper which is a common trend of all SSL models within the first few layers, but TERA failed to provide further high-level information with such a shallow network. Wav2Vec2 achieved the highest SI-SNR (9.81dB) for zero overlap, indicating the model can extract effective speaker information from clean speech signals to help identify speaker transitions.

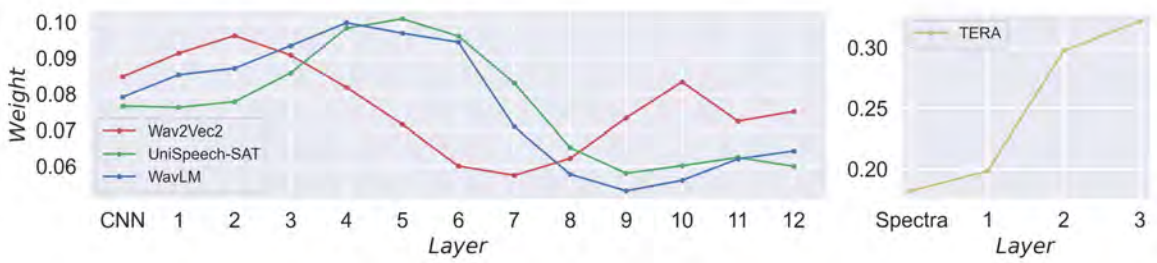


Fig. 4.3 Weight analysis for different SSL models.

As shown in Table 4.10, the results on LibriCSS are similar to those of SparseLibriMix. In phase one, UniSpeech-SAT and WavLM have lower average WERs than TERA and Wav2Vec2. However, contrary to the results on SparseLibriMix, TERA's WERs are slightly lower than Wav2Vec2. The possible reason is that spectra alternations used to pre-train TERA improve its generalisation. Furthermore, WavLM outperformed UniSpeech-SAT by 0.5% in average WER since WavLM's generalisation was promoted by noise augmented data. After being fine-tuned (phase two), all SSL models achieved better performance on both datasets. The improvement of TERA is the smallest on the SparseLibriMix due to its small size. Compared with the frozen WavLM and UniSpeech-SAT, the fine-tuned Wav2Vec2 has better SI-SNR on SparseLibriMix but the WERs on LibriCSS are still higher, indicating poor generalisation ability.

In summary, with similar architectures, the effectiveness of SSL models and their generalisations are mostly determined by training data and augmentations. WavLM's training set covers diverse speech contents and noise types, contributing to the promising performance in source separation.

4.3.5 Combination of WavLM and TERA

Since SSL models were trained differently, they may extract complementary information from speech signals. Therefore, combining SSL representations is potentially helpful for source separation. The combination of WavLM and TERA through the weighted-sum was explored because of their intrinsic differences. Firstly, TERA operates in the frequency domain whereas WavLM captures features from waveforms. Secondly, TERA was pre-trained with generative criteria, unlike WavLM which utilised discrete targets. Moreover, TERA will not introduce extensive computational overheads.

As shown in Table 4.11 and Table 4.12, when only the downstream model was trained, the combined model improved the SI-SNR on the SparseLibriMix dataset but the out-of-domain performance on LibriCSS was not enhanced by the additional low-level features from TERA. After base models were fine-tuned, the combined model is superior on both datasets because it has more parameters and two network paths that extract features from both time and T-F domains. As illustrated in Figure 4.4, after fine-tuning (phase 2), the importance of TERA's representations increased.

The two-phase training scheme provided insight that the order of fine-tuning is critical to out-of-domain generalisation. It is more reasonable to firstly fine-tune the model part

Method	Overlap ratio in % 0	20	40	60	80	100	AVG
Phase 1: train the downstream model.							
WavLM	**6.49**	8.40	8.06	7.21	7.00	6.35	7.25
WavLM + TERA	6.03	**8.49**	**8.16**	**7.29**	**7.17**	**6.47**	**7.26**
Phase 2: fine-tune the base model.							
WavLM	13.75	**11.85**	10.12	8.74	8.22	7.88	10.09
WavLM + TERA	**13.90**	11.84	**10.29**	**8.93**	**8.44**	**7.94**	**10.22**
WavLM + TERA (ordered fine-tune)	12.25	10.96	9.76	8.44	8.12	7.57	9.47

Table 4.11 The evaluation of the combined model between WavLM and TERA on SparseLibriMix dataset measured by SI-SNR ↑ (dB). 'WavLM + TERA' means features from two models are combined through weighted-sum. Ordered fine-tuning means TERA is fine-tuned before WavLM.

Method	Overlap ratio in % 0L	0S	10	20	30	40	AVG
Phase 1: train the downstream model.							
WavLM	**2.9**	**3.0**	**5.6**	**8.4**	**11.2**	**13.3**	**7.4**
WavLM + TERA	**2.9**	**3.0**	**5.6**	**8.4**	11.4	13.5	7.5
Phase 2: fine-tune the base model.							
WavLM	**2.8**	**2.9**	5.1	7.5	9.8	12.0	6.7
WavLM + TERA	**2.8**	**2.9**	5.1	7.3	9.5	11.6	6.5
WavLM + TERA (ordered fine-tune)	3.0	3.0	**4.8**	**7.1**	**9.2**	**10.9**	**6.3**

Table 4.12 The evaluation of the combined model between WavLM and TERA on LibriCSS dataset measured by WER ↓ (%). 'WavLM + TERA' means features from two models are combined through weighted-sum. Ordered fine-tuning means TERA is fine-tuned before WavLM.

that is less well-trained. For example, the Conformer block was randomly initialised, so it should be trained first. Following this idea, TERA which was pre-trained on a smaller amount of data should be fine-tuned before WavLM. The second phase of training was split into two parts. In the first part, TERA and the Conformer were trained. In the second part, the whole model was unfrozen and fine-tuned. With the 'ordered fine-tuning', the lowest average WER on LibriCSS was observed and improvements are obvious for large overlaps. However, as WavLM was fine-tuned for fewer steps, the in-domain performance is slightly

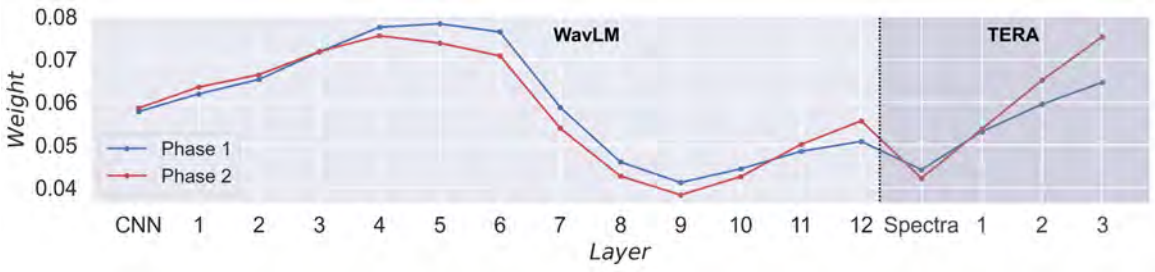


Fig. 4.4 Weight analysis for the combined model of TERA and WavLM.

poorer (i.e. lower SI-SNR on SparseLibriMix). It is similar to the situation where training the downstream model was compared with directly fine-tuning the whole model (section 4.3.3).

4.3.6 Comparisons with Baselines

ID	Model	Overlap ratio in % 0L	0S	10	20	30	40	AVG
	Baselines							
1	No separation	2.9	3.1	9.2	16.4	24.3	32.3	14.7
2	Conformer-Large (Chen et al., 2021b)	4.1	4.7	6.6	9.6	12.9	15.3	8.9
3	WavLM (frozen) (Chen et al., 2022a)	4.5	4.4	5.6	7.5	9.4	**10.9**	7.4
4	WavLM (unfrozen) (Chen et al., 2022a)	4.5	4.3	5.9	8.3	11.1	12.5	8.2
	Ours							
5	WavLM (frozen)	2.9	3.0	5.6	8.4	11.2	13.3	7.4
6	WavLM (fine-tuned)	**2.8**	**2.9**	5.1	7.5	9.8	12.0	6.7
7	WavLM + TERA (ordered fine-tuned)	3.0	3.0	**4.8**	**7.1**	**9.2**	**10.9**	**6.3**

Table 4.13 The comparison between our models and baselines on the LibriCSS dataset measured by WER ↓ (%).

As shown in Table 4.13, our models were compared with several baselines on LibriCSS. SparseLibriMix was not used as it includes the same type of noise and reverberation as the training set, making the comparison unfair. The pre-trained Conformer-Large (Chen et al., 2020) was re-evaluated using the same ASR model as other experiments whereas WavLM's results were directly taken from the paper (Chen et al., 2022a) where a different ASR model was used [3]. The training set for the baseline models was similar to the reverberated LibriMix

[3]The WavLM paper does not open source the model for source separation. When evaluating the performance on LibriCSS, they used an ASR model with 2.08%/4.95% WERs on LibriSpeech clean/other. The ASR model that was used has WERs of 1.9%/3.9% on LibriSpeech clean/other.

in terms of duration and noise. Compared with Conformer-Large, all our models achieved lower WERs. It should be emphasized that when the WavLM was frozen, the model only has 1.8M trainable parameters, but the performance is already better than Conformer-Large with 58.7M parameters.

In the existing work, unfreezing the base model led to overfitting which can be seen from the third and the fourth row of Table 4.13. The fine-tuning schedule alleviated this problem so that training the base model contributed to performance improvements (comparing the 5th and the 6th row). Moreover, after the combination of WavLM and TERA was fine-tuned, WERs are lower compared with all baselines. Although a slightly better ASR system was used, the combined model was trained for fewer steps compared with baselines (i.e. 180k steps versus 260k steps). Furthermore, unlike the baselines' implementations (the 3rd and the 4th row), ASR model and CTC loss (Watanabe et al., 2017) were not included into the training process and a much smaller downstream model was used with 1.8M parameters rather than a Conformer model with 22.1M parameters. Therefore, in general, the proposed method is both more effective and efficient.

4.4 Experiments on the AMI Corpus

In this section, the separation models were extended to AMI, a real meeting corpus that contains conversation-like recordings. The beam-formed recordings from distant microphones were used. Since AMI is significantly different from the synthetic training set in terms of ambient noise and speech content, the model should be fine-tuned on AMI corpus. PIT requires clean reference signals, so non-overlapped utterances were first selected from the AMI training set and then mixtures were created with SNR between -5dB to 5dB, resulting in the AMI-clean dataset. The non-overlapped utterances only take up around 25% of the AMI training set, so to fully utilise all utterances, AMI-full dataset was created using the complete corpus. In this case, the target may contain multiple speakers, so AMI-full was only used in unsupervised MixIT. To allow signal-based evaluation using SI-SNR, Syn-AMI was created from non-overlapped utterances in the AMI test set following the same mixing approach as AMI-clean. Note that AMI-clean, AMI-full, and Syn-AMI consist of fully overlapped mixtures and their details are included in Appendix A. It was found that the combined model of WavLM and TERA did not provide performance gain on AMI as TERA's pre-training dataset (LibriSpeech) is considerably dissimilar to AMI and the larger model tends to overfit small training sets. Besides, the TERA branch increased the training time by around 30%. Therefore, a single WavLM was applied as the base model.

4.4.1 Fine-Tuning

As shown in Table 4.14, the WavLM-based separation model trained on the LibriMix dataset (the 2nd row) can already significantly improve the ASR performance on the AMI dataset. Compared with no separation, the absolute WER reductions are 5.8% and 6.5% on the AMI test and development sets respectively. For 40% overlap, the WER decreases by more than 10%. The model was re-trained using the AMI-clean dataset (the 3rd row) which is more aligned with AMI corpus than LibriMix. However, only SI-SNR on Syn-AMI increases whereas WERs on AMI slightly decrease. This indicates that training on the in-domain AMI-clean improved the sample-wise separation accuracy but since AMI-clean is a small dataset that contains fewer distinct speakers and words than LibirMix, the model is less effective to separate speech content.

	Syn-AMI (SI-SNR ↑)	**AMI** (WER ↓) Overlap ratio in %					
Training set		10	20	30	40	Test	Dev
None (No Separation)	0.14	29.7	39.6	48.6	54.4	43.8	41.2
LibriMix	2.81	**27.5**	35.0	40.6	44.2	38.0	34.7
AMI-clean	4.35	27.8	34.9	40.4	44.1	38.3	35.1
LibriMix->AMI-clean	**4.61**	27.6	**34.5**	**40.0**	**43.0**	**37.7**	**34.5**

Table 4.14 Separation performance evaluated on AMI datasets. All models are based on WavLM with multi-phase training. 'LibriMix->AMI-clean' means that the model trained on LibriMix was further fine-tuned with the AMI-clean dataset.

To improve the in-domain performance, the model trained on LibriMix was further fine-tuned with the AMI-clean dataset for 80k steps (the last row). The model achieved both the highest SI-SNR on Syn-AMI and the lowest WERs on AMI. It can be concluded that the model pre-trained on the synthetic dataset encoded prior information of diverse mixtures and is effective on the real overlapped corpus. The in-domain fine-tuning further promoted the performance, especially the sample-wise accuracy.

4.4.2 Comparisons between PIT and MixIT

Unsupervised MixIT introduced in section 2.4.2 supports the training of separation models without clean references, so the AMI-full dataset can be used. Table 4.15 shows the comparisons between PIT and MixIT where the downstream model was mainly trained. Comparing the third row with the top two rows, MixIT consistently provides lower WERs

Method	Training set	**Syn-AMI** (SI-SNR ↑)	**AMI** (WER ↓) Overlap ratio in %					
			10	20	30	40	Test	Dev
PIT	LibriMix	1.93	28.2	35.8	41.7	46.4	38.9	36.2
PIT	AMI-clean	**3.33**	28.2	36.0	42.6	45.6	39.2	36.2
MixIT	AMI-full	2.81	27.7	35.6	**41.1**	45.4	38.3	35.7
PIT&MixIT	LibriMix&AMI-full	2.88	**27.4**	**35.3**	41.3	**44.8**	**38.0**	**34.9**
PIT&MixIT*	LibriMix&AMI-full	4.13	**27.3**	**33.8**	**39.8**	**42.1**	**37.0**	**33.5**

Table 4.15 The comparison between supervised PIT, unsupervised MixIT, and semi-supervised PIT&MixIT. The last line (PIT&MixIT*) represents the results after two-phase semi-supervised training. In other experiments, only the downstream model was trained.

on all subsets of the AMI. The main reason is that MixIT makes use of real overlapped utterances, increasing the diversity of training data. Furthermore, MixIT is not limited to speech components. With four output sources, the non-speech components like coughing, laughing, and background noises were sometimes separated from speech signals.

MixIT does not explicitly guide the model to separate a single speaker's signal into one output stream which may lead to over-separation. Semi-supervised learning combines PIT with MixIT and utilises both LibriMix and AMI-full datasets. In this case, the dataset is enlarged and the model sees clean single-speaker signals during training. For implementation, PIT was used with a probability of 20% and batches were randomly sampled from LibriMix. Otherwise, MixIT was adopted with batches sampled from AMI-full. MixIT dominated the training process since the convergence was more stable than using a higher ratio of PIT. With semi-supervised training (PIT&MixIT), the WERs are lower than unsupervised MixIT. After the base model (PIT&MixIT*) was fine-tuned, the performance is better on AMI compared with all supervised models in Table 4.14. The downside of MixIT is that it requires more output sources which increases the difficulty in a real scenario where the desired output source need to be selected without ground truth (details in the next chapter). Moreover, supervised training contributed to higher SI-SNR on Syn-AMI because the model directly learned to separate mixtures created from non-overlapped speech but it may not reflect real-world performance.

In all experiments so far, masks were generated by Sigmoid activation which did not introduce correlations between outputs. MixIT makes output sources interconnected since they need to be recombined. Therefore, Softmax can be a better masking function as it

Masking activation	**Syn-AMI** (SI-SNR ↑)	**AMI** (WER ↓) Overlap ratio in % 10	20	30	40	Test	Dev
Sigmoid	4.13	27.3	33.8	39.8	42.1	37.0	33.5
Softmax	3.90	27.1	34.0	39.5	43.1	37.0	33.6

Table 4.16 The comparison between masks generated by Sigmoid and Softmax functions.

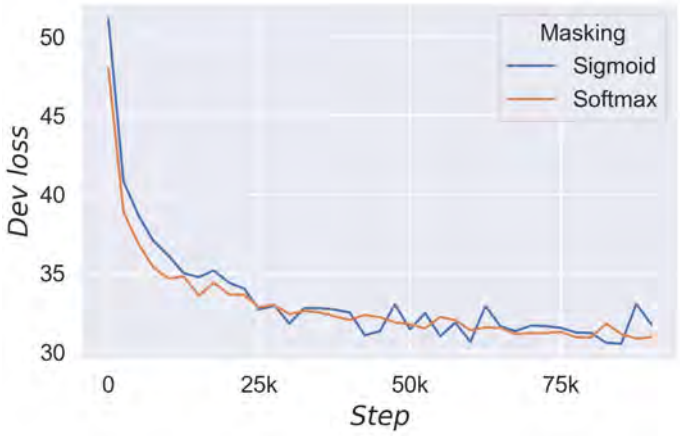


Fig. 4.5 Model convergence with different masking functions when only the downstream model is trained.

enforces sum-to-one masks, so the mixture of all output sources is about the same as the input which is consistent with the hidden assumption of MixIT. Figure 4.5 illustrates the evolution of development loss on AMI-full dataset. With Softmax activation, the model converged slightly faster with fewer fluctuations. However, Softmax activation did not improve the final performance (Table 4.16), but it promoted the source selection process (details in section 5.3.2).

Chapter 5

Automatic Transcription System

Experiments in chapter 4 emphasised the separation process without a target speaker, so during evaluation, the output permutation was determined by comparing the predictions with the ground-truth signals or transcriptions. In other words, the separation model is effective as long as some of the output sources contain the desired single speaker signals. Compared with source separation, speaker extraction is a more realistic scenario since it does not rely on references for evaluation. For conventional speaker extraction, the model only estimates a single-speaker audio stream from the overlapped speech. To identify the target speaker, the speaker embedding extracted from pre-recorded enrollment audio by an auxiliary branch is injected into the main network (Delcroix et al., 2018). In this thesis, speaker extraction was decoupled into two steps: source separation and source selection, so pre-trained separation models can be applied directly without modifying the architecture and retraining. Moreover, this approach does not require enrollment audio and can be easily plugged into a transcription system. In this chapter, speaker embeddings used in source selection are first introduced. Then, the proposed source selection methods are presented and examined. Finally, the whole separation system is evaluated in the context of an automatic transcription system.

5.1 Speaker Embedding

5.1.1 X-Vector

Speaker embeddings are low-dimensional representations that capture information about speaker identities. It is widely used in speaker diarisation and text-independent speaker verification. Modern deep learning-based speaker embeddings commonly involve three parts: a frame-level network that extracts temporal context from front-end features, a statistical pooling layer that maps variable length features to a fixed size vector, and a segment-level

network that follows the pooling layer.

The X-Vector (Snyder et al., 2017) was one of the first approaches for neural-based speaker embedding. It uses 20-dimensional MFCCs as inputs and adopts a 5-layer Time Delay Neural Network (TDNN) as the frame-level model. The TDNN provides a temporal context of 17 frames. Then, the statistics pooling layer computes and concatenates the global mean and variance. Finally, these statistics are fed into a 2-layer segment-level FFN. The model is trained through a speaker classification task with the multi-class cross-entropy loss. After training, the speaker embedding can be extracted from both layers of the segment-level network.

5.1.2 ECAPA-TDNN

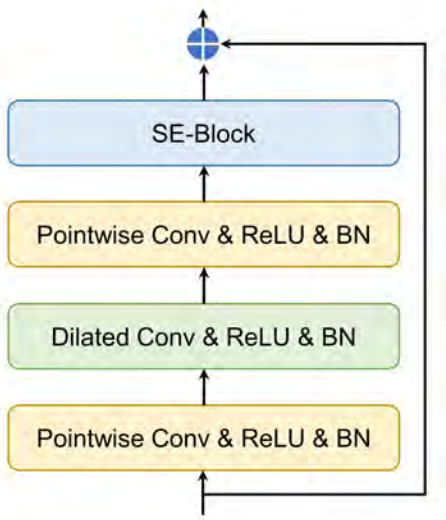


Fig. 5.1 Illustration of a SE-Res2Block.

ECAPA-TDNN (Desplanques et al., 2020) enhanced the X-Vector method by improving the frame-level network with SE-Res2Block. As shown in Figure 5.1, the SE-Res2Block first uses a pointwise convolution layer to reduce the number of channels. Then, the bottleneck features pass through a dilated convolution layer that expands the perceptual field. Another pointwise convolution layer follows the dilated convolution to recover the original feature size. Finally, the SE-Block exploits global statistics to scale each channel. Before the pooling layer, output features from all SE-Res2Blocks are concatenated and passed through a linear layer to utilise multi-level information.

Another improvement of ECAPA-TDNN is the integration of temporal and channel attention into the statistics pooling layer which allows the model to focus on multiple locations regarding different channels. As shown in Equation 5.1, the output of frame-level network $\mathbf{h}_t$ is transformed into attention scores $e_{t,c}$ which are normalised by Softmax activation (Equation 5.2). Then, the normalised attention scores $\alpha_{t,c}$ are used to compute the weighted mean of frame-level features (Equation 5.3) which replaces the non-weighted mean in the original statistics pooling layer. Furthermore, the standard deviation is also substituted by a weighted version shown in Equation 5.4. Finally, the output of the attentive pooling layer is the concatenation of weighted statistics $\hat{\mu}$ and $\hat{\sigma}$.

$$e_{t,c} = \mathbf{v}_c^T f(\mathbf{W}\mathbf{h}_t + \mathbf{b}) + k_c \tag{5.1}$$

$$\alpha_{t,c} = \frac{exp(e_{t,c})}{\sum_{\tau}^{T} exp(e_{\tau,c})} \tag{5.2}$$

$$\hat{\mu}_c = \sum_{t}^{T} \alpha_{t,c}\, h_{t,c} \tag{5.3}$$

$$\hat{\sigma}_c = \sqrt{\sum_{t}^{T} \alpha_{t,c}\, h_{t,c}^2 - \hat{\mu}_c^2} \tag{5.4}$$

where weight matrix $\mathbf{W}$ maps $\mathbf{h}_t$ to lower dimensional space to alleviate overfitting. $\mathbf{v}_c$ and k_c are channel-dependent weight and bias respectively.

5.2 Source Selection

5.2.1 Input Embedding

Under the assumption that utterance boundaries are known and speech signals are sparsely overlapped, a source can be selected by comparing the embeddings of input and output sources. To be more specific, the input utterance is dominated by the main speaker and the overlap occupies a small duration. Hence, the model only separates a small portion of speech components. Therefore, the output source corresponding to the main speaker is the one that has the most similar features to the input. In practice, different feature spaces were compared. When using ECAPA-TDNN, the output source whose speaker embedding had the highest cosine similarity with the input embedding was chosen. Since input and the desired output are of the same length and have similar contents, content-dependent embeddings from SSL models can be used instead of fixed-length speaker representations. In this case, the output

source whose embedding extracted by an SSL model had the lowest Euclidean distance with the input's embedding was selected.

5.2.2 Iterative Selection

Ideally, one of the output sources contains all speech components of the target speaker. Therefore, the correct source can be identified if the speaker information is available. Commonly, prior information of target speakers is acquired by computing speaker embeddings from enrollment audios. In order not to rely on pre-recorded audio, it is assumed that input utterances come from a diarisation system that provides speaker labels. Consequently, average speaker embedding can be derived from utterances of the same speaker. However, these utterances are polluted by overlaps, so an iterative approach is proposed to refine the speaker embedding and some outlier utterances are removed. Detailed steps are as follows:

- **Step 1**: Compute fixed-length speaker embedding of every utterance using ECAPA-TDNN.
- **Step 2**: Compute the average embedding for each speaker.
- **Step 3**: For each speaker, remove part of outliers with high Euclidean distances to the average embedding.
- **Step 4**: Re-compute the average speaker embedding without outliers.
- **Step 5**: Select the output source whose embedding has the highest cosine similarity with the average speaker embedding.
- **Step 6**: Compute the embedding of every selected output source.
- **Step 7**: Return to step 2.

Compared with using input embedding, the iterative source selection approach is applicable for utterances with large overlaps. However, it requires speaker labels and adequate utterances for each speaker so that average embeddings are representative.

5.3 Experiments for Source Selection

To evaluate the source selection performance, source selection methods were compared with the orcale selection in terms of WERs. The orcale selection was determined by the

ground-truth transcription, so it minimized the WER of a particular ASR system [1]. The selection accuracy was also computed which was defined as the percentage of duration that the estimated selection matches the orcale selection.

5.3.1 Comparisons between Input Embeddings on LibriCSS

	Overlap ratio in %						
Embedding	0L	0S	10	20	30	40	AVG
Oracle	2.8	2.9	5.1	7.5	9.8	12.0	6.7
ECAPA	2.9	2.9	5.2	8.0	11.2	14.7	7.5
Wav2Vec2	2.9	2.9	5.4	7.9	11.1	14.7	7.5
WavLM	2.9	2.9	5.2	7.9	10.5	14.0	7.2

Table 5.1 The comparison of the source selection process using input embedding extracted from different models based on the LibriCSS dataset measured by WER $\downarrow$ (%). 'Oracle' means using the ground truth transcription to choose the output source that minimizes the WER.

For the LibirCSS dataset, the overlap ratios of all utterances are less than or equal to 40% but the number of utterances for each speaker is limited (i.e. around 10). Therefore, the LibriCSS dataset was adopted to assess the source selection method with input embeddings. Table 5.1 demonstrates the comparison between embeddings extracted from three different models. Separation was performed by the best WavLM-based model in section 4.3. When Wav2Vec2 and WavLM were used, the embeddings were taken from the last Transformer layer. It can be seen that WERs are similar for all types of embeddings with small overlaps (i.e. less or equal to 20%) and the selection accuracy is above 99% (Figure 5.2). However, when the overlap ratio is 30% and 40% in which case distinguishing between primary and interfering speakers is more difficult, WavLM's embedding gave rise to lower WERs and higher selection accuracy (red line) than Wav2Vec2 and ECAPA. The main reason is that WavLM was exposed to a large amount of overlapped data during self-supervised pre-training while ECAPA and Wav2Vec2 only saw single-speaker audios. Furthermore, during pre-training, the pseudo labels that WavLM needed to predict were derived from the main speaker, so it enforced WavLM to capture the main speaker's information while overlooking the interfering speaker. In general, the desired source can be successfully selected by comparing embeddings before and after separation. Although for higher overlap

[1] In this section, Wav2Vec2-Robust fine-tuned on Switchboard was used as the ASR model.

ratios, the performance gap compared with oracle selection is larger, with WavLM, the selection accuracy is still higher than 96%.

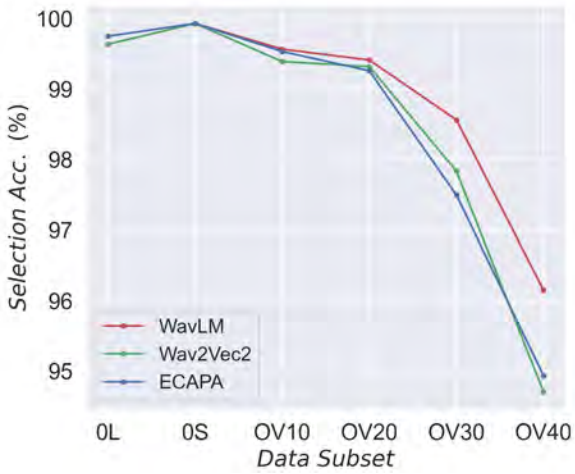


Fig. 5.2 The percentage of duration that the system selects the correct output source with different input embeddings.

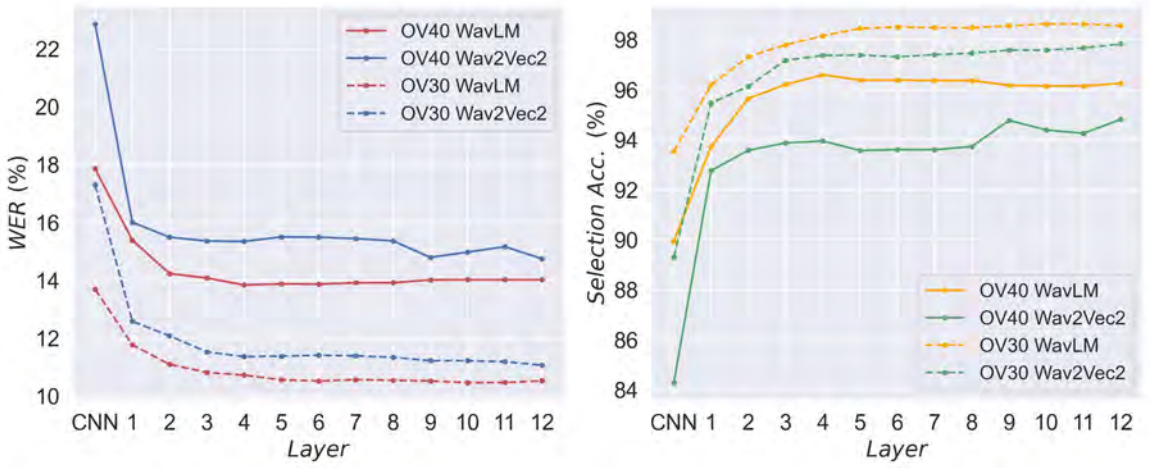


Fig. 5.3 The evaluation of the source selection process using input embedding extracted from different layers of WavLM and Wav2Vec2. The comparison is based on the LibriCSS dataset with 30% and 40% overlaps (OV30 and OV40).

Inspired by weight analyses, embeddings from multiple levels were compared. As shown in Figure 5.3, for both WavLM and Wav2Vec2, features from deeper layers contributed to lower WERs and higher selection accuracy, meaning they are more robust against overlaps. However, when the embeddings come from layers deeper than the fourth Transformer block, the performance did not change much, so the source selection efficiency can be improved by

using part of the SSL model. Though lower layers are less robust in terms of source selection, it was verified that their output features are crucial for source separation. This discrepancy can be explained by the fact that features from the primary speaker are favored for source selection whereas, for source separation, features containing rich information about both speakers are also important.

5.3.2 Source Selection on AMI Corpus

Method	Oracle	Input Embedding		Iterative Selection	
	WER ↓	WER ↓	Selection Acc. ↑	WER ↓	Selection Acc. ↑
PIT	38.3 / 35.1	41.6 / 38.4	84.9 / 85.3	39.8 / 36.6	91.0 / 91.1
PIT*	37.7 / 34.5	41.0 / 37.9	**90.7 / 91.0**	**39.0** / 35.9	**92.2 / 93.6**
PIT&MixIT	**37.0 / 33.5**	40.9 / 37.8	83.2 / 82.9	39.8 / 35.7	85.6 / 86.8
PIT&MixIT*	**37.0** / 33.6	**40.6 / 37.5**	83.6 / 84.8	39.1 / **35.5**	85.4 / 87.0

Table 5.2 The comparison between separation models on AMI datasets using different source selection methods. 'Test set/Development set' WERs (%) and Selection Accuracy (%) are provided. 'PIT' and 'PIT*' use two output sources where 'PIT' adopts the AMI-clean dataset directly whereas 'PIT*' first uses the LibirMix dataset and then transfers to the AMI-clean dataset. 'PIT&MixIT' and 'PIT&MixIT*' are semi-supervised training with four output sources that use Sigmoid and Softmax as masking functions respectively.

On the AMI Corpus, the WavLM embedding was first adopted for source selection and four WavLM-based separation models were assessed (the 3rd column of Table 5.2). The comparison between the first and the second row demonstrates that when the numbers of output sources are both two, the model with better oracle performance also achieved superior WERs after source selection as clear separation benefits both source selection and ASR. When the separation models with four output sources were trained by the semi-supervised method (the 3rd and the 4th row), the WERs corresponding to the oracle sources is notably lower than supervised models. However, after source selection using input embedding, the improvements are smaller because more output sources significantly increase the difficulty of source selection. With four output sources, the selection accuracy is less than 85% but for the best two-source model it is higher than 90%. Therefore, in a real-world scenario, the conventional permutation invariant evaluation is not fair when comparing models with different numbers of output sources. What truly matters is the trade-off between separation performance and selection accuracy. Another finding is that compared with masks generated by Sigmoid activation (the 3rd row), using the Softmax function (the 4th row) improved the selection accuracy and WERs. A possible explanation is that Softmax enforced complemen-

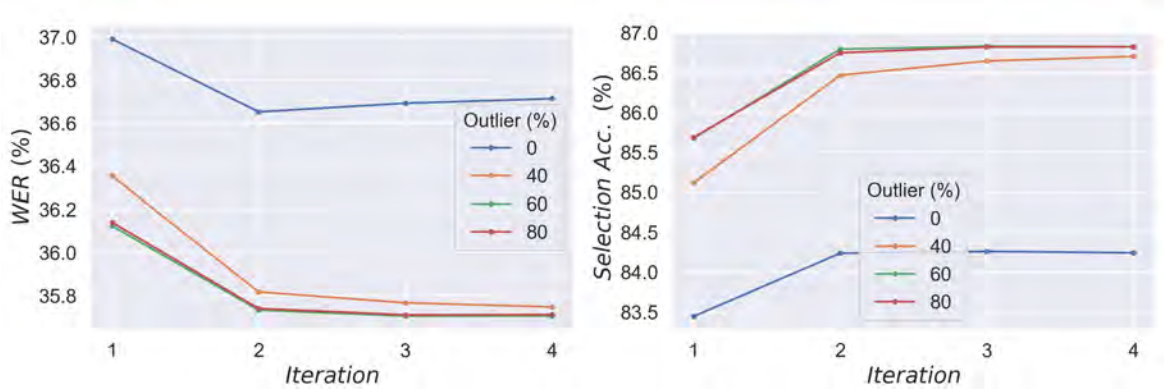


Fig. 5.4 The iterative source selection process on the AMI development set. This is an example of a model with four output sources trained through semi-supervised PIT&MixIT. 'Outlier' means the percentage of utterances that are removed before computing the average speaker embedding.

tary masks that sum to one, making output sources less likely to be similar.

A meeting session in AMI lasts for more than half an hour with around four participants. Hence, each speaker normally has more than 100 utterances. Although the majority of utterances are non-overlapped or sparsely overlapped, some utterances do have large overlaps. Thus, the iterative source selection method is more suitable. As illustrated in Figure 5.4, the effects of hyperparameters were first investigated on the development set. It can be seen that removing outliers before computing the average speaker embedding substantially improved the WERs and the selection accuracy compared with the blue line where all utterances are used. This is because outliers are distractors that include little information about the target speaker due to overlaps, noises, or extremely short duration. Overall the iterative process is helpful since the performance was enhanced with more iterations and the improvements are most obvious between the first and the second iteration. However, for some session recordings with high noise levels, there is a risk of divergence where incorrect source selections lead to even worse performance in future iterations. Therefore, in practice, only two iterations were performed and 60% of outliers were removed to mitigate divergence. As shown in Table 5.2, with additional information about speaker and the ability to handle fully-overlapped segments, the iterative selection approach significantly improved the WERs and selection accuracy for all separation models compared with the approach that utilises input embedding.

5.4 Experiments for the Transcription System

Experiments in the previous section focus on source selection so ground-truth speaker labels and utterance boundaries were used. In this section, a fully automatic transcription system is considered.

5.4.1 System Setup

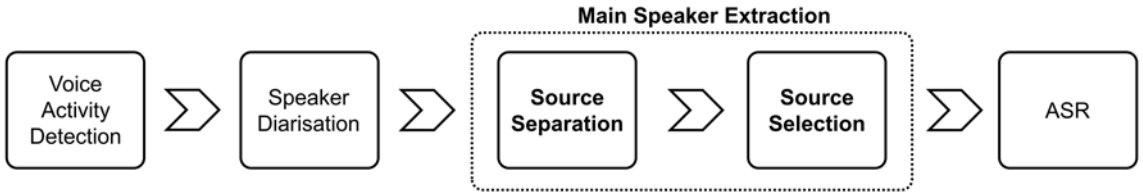


Fig. 5.5 The flowchart of the transcription system.

Figure 5.5 illustrates the flowchart of the transcription system. First, continuous session recordings were cropped into utterances and tagged with speaker labels by voice activity detection (VAD) and speaker diarisation. Then, overlaps were removed and the main speaker's speech signals were kept through source separation and source selection. Finally, an ASR model was applied to produce transcriptions. The details are as follows:

- **VAD**: Wav2Vec2-base was fine-tuned on AMI to estimate the presence/absence of speech components. In detail, a feed-forward layer followed by Softmax activation was added on top of Wav2Vec2 to map feature vectors to probabilities. Time intervals between speech regions were constrained to be more than 0.04s and non-speech regions less than 0.4s were converted to speech regions.
- **Diarisation**: ECAPA-TDNN (Dawalatabad et al., 2021) which was trained on VoxCeleb1 and VoxCeleb2 datasets (Chung et al., 2018; Nagrani et al., 2017) was used to generate speaker embeddings with a window size of 3 seconds and stride of 1.5 seconds. Then, spectral clustering was applied with a maximum speaker number of 10. If overlaps were not considered and estimated VAD was used, this approach achieved diarisation error rates (DERs) (Anguera, 2008) of 7.81 and 7.61 on AMI development and test sets respectively. If overlaps were taken into account, the DERs increased to 11.07 and 10.09 on AMI development and test sets. Note that the diarisation model assumed a single active speaker.

- **Source Separation**: According to the performance in previous sections, the WavLM-based model fine-tuned on LibirMix and AMI-full datasets through semi-supervised training was chosen. The model estimates four masks with Softmax activation.
- **Source Selection**: The iterative source selection method was adopted where speaker embeddings were generated from the same ECAPA-TDNN model used in diarisation. The number of iterations was set to 2 and 60% of outliers were removed before computing average speaker embeddings.
- **ASR**: So far, ASR models were trained on non-overlapped data (i.e. LibirSpeech and Switchboard). Now ASR models fine-tuned on AMI will be evaluated. Three ASR models that use Wav2Vec2-Robust as the base model were compared: 1) W2V2-SWB was fine-tuned on the Switchboard; 2) W2V2-AMI was fine-tuned on AMI; 3) W2V2-AMI-Sep was a further fine-tuned version of W2V2-AMI, using separated audio from AMI (details in Appendix B). To make a fair comparison, the total numbers of epochs were same for W2V2-AMI and W2V2-AMI-Sep.
- **Evaluation**: Since oracle utterance segmentations were not used, the ground-truth transcription for each utterance was unavailable. Therefore, the traditional WER can not be computed. An alternative is to calculate the cpWER-us (Zheng et al., 2022) at the session level. First, the hypotheses and references were concatenated respectively for each speaker in each session following chronological order. Then, WERs of all possible speaker permutations were computed. For redundant speaker, hypotheses were removed. Finally, the lowest WER among them was chosen as the cpWER-us.

5.4.2 Results

Table 5.3 shows the influence of source separation on the transcription system. For the model trained on non-overlapped data, the cpWER-us significantly decreased (the 1st and the 2nd row). However, if the model was fine-tuned on AMI, source separation noticeably harmed the performance (the 3rd and the 4th row) due to the increase of deletion errors. There are several reasons: 1) If the model has seen overlapped data during training, it implicitly learned to identify the main speaker and ignore the speaker in the background. After separation, more speech components may be incorrectly recognised as background. 2) The separation model introduced out-of-domain data like utterances with longer silence and system noise. The model overfitted the AMI dataset, so it failed to generalise to the data after separation. 3) The source separation and selection essentially removed audio components, so their mistakes

ID	ASR	Separation	cpWER-us (%)	Sub. (k)	Ins. (k)	Del. (k)
1	W2V2-SWB	×	46.0 / 43.9	16.7 / 16.4	1.3 / 1.3	23.2 / 24.0
2	W2V2-SWB	✓	43.8 / 40.8	15.6 / 14.9	1.1 / 1.1	22.6 / 22.7
3	W2V2-AMI	×	35.1 / 34.2	10.3 / 11.2	1.7 / 1.9	19.5 / 19.4
4	W2V2-AMI	✓	36.8 / 36.4	9.2 / 9.8	1.1 / 1.3	22.8 / 23.4
5	W2V2-AMI-Sep	×	34.4 / 33.5	10.9 / 11.8	2.0 / 2.3	18.0 / 17.6
6	W2V2-AMI-Sep	✓	**33.6 / 32.3**	9.7 / 10.2	1.5 / 1.7	19.0 / 18.7
7	ROVER (2&5&6)	✓	**33.0 / 31.8**	9.3 / 9.9	1.3 / 1.6	19.0 / 18.7

Table 5.3 The comparison of transcription systems with or without source separation. 'Test set/Development set' cpWER-us are provided. All ASR models utilised the pre-trained Wav2Vec2-Robust model (Hsu et al., 2021b). W2V2-SWB and W2V2-AMI were fine-tuned on the Switchboard and AMI respectively; W2V2-AMI-Sep was a further fine-tuned version of W2V2-AMI, using separated audio from AMI.

will lead to deletion errors.

To tackle the first two issues, the ASR model was fine-tuned on the separated data. By comparing the fourth and the sixth row, it can be seen that after fine-tuning, the model adapted to the separated data, and its performance was substantially boosted. Furthermore, compared with the third row, source separation improved the cpWER-us with absolute reductions of 1.5%/1.9% on the AMI test/development sets (the 6th row). The model fine-tuned on separated data (the 5th row) also displayed better performance with non-separated data which implies that overlaps overcomplicate the training of ASR models.

To gain benefits from multiple systems that use different ASR models and input audio (i.e. before or after separation), ROVER (Fiscus, 1997) was employed for system fusion which first aligns word transition networks and then carries out majority voting. Hypotheses from three experiments (the 2nd, the 5th, and the 6th row) were combined at the utterance level. In this case, 2 out of 3 systems used separated audio and ASR models of 2 systems have been fine-tuned on AMI. It can be seen from the last row that ROVER further improved the performance with absolute cpWER-us reductions of 0.6%/0.5% on the test/development sets.

5.4.3 Error Analysis

Deletions are the main errors made by the transcription system. To understand the reason, the system was evaluated with different prior information. Table 5.4 (a) shows the results when ground-truth utterance boundaries and speaker labels are available. In other words,

ASR	Separation	cpWER-us (%)	Sub. (k)	Ins. (k)	Del. (k)
W2V2-AMI	×	28.4 / 27.4	14.6 / 14.3	1.5 / 1.7	9.4 / 9.9
W2V2-AMI-Sep	✓	28.6 / 27.5	14.8 / 14.9	1.5 / 1.7	9.4 / 9.5

(a) Ground-truth diarisation.

ASR	Separation	cpWER-us (%)	Sub. (k)	Ins. (k)	Del. (k)
W2V2-AMI	×	34.4 / 33.0	12.6 / 12.9	2.7 / 2.9	15.6 / 15.5
W2V2-AMI-Sep	✓	33.7 / 32.7	12.4 / 12.5	2.2 / 2.9	15.6 / 15.6

(b) Ground-truth utterance boundaries and ECAPA-based diarisation.

ASR	Separation	cpWER-us (%)	Sub. (k)	Ins. (k)	Del. (k)
W2V2-AMI	×	35.8 / 35.2	10.3 / 10.8	1.5 / 1.5	20.3 / 21.1
W2V2-AMI-Sep	✓	34.4 / 33.6	9.8 / 10.1	1.2 / 1.4	19.8 / 20.4

(c) Ground-truth VAD and ECAPA-based diarisation.

ASR	Separation	cpWER-us (%)	Sub. (k)	Ins. (k)	Del. (k)
W2V2-AMI	×	35.1 / 34.2	10.3 / 11.2	1.7 / 1.9	19.5 / 19.4
W2V2-AMI-Sep	✓	33.6 / 32.3	9.7 / 10.2	1.5 / 1.7	19.0 / 18.7

(d) W2V2-based VAD and ECAPA-based diarisation (the fully automatic system).

Table 5.4 The comparison of transcription systems with different prior information. 'Test set/Development set' cpWER-us is provided.

Method	Number of Utterances	Total Duration	Average Duration
Ground Truth	17100	25164s	1.47s
Ground-truth VAD & ECAPA	10588	22815s	2.15s
W2V2-based VAD & ECAPA	7178	23104s	3.22s

Table 5.5 A summary of the processed data after diarisation.

VAD and diarisation were assumed to be perfect and overlaps were considered. In this case, the lowest cpWER-us and deletion errors were observed. Therefore, mistakes in VAD and diarisation can be the main causes of deletion errors.

Taking one step further, ground-truth utterance boundaries were still used but speaker labels were predicted with ECAPA-TDNN. It can be seen from Table 5.4 (b) that cpWER-us and deletion errors noticeably increase compared with Table 5.4 (a). There are two reasons:

1) the diarisation model does not know the number of speakers in each session, so if it predicts more speakers than expected, the deletion errors increase; 2) if the utterance is misclassified, it can cause deletion errors for one speaker and insertion errors for another speaker. Table 5.4 (c) shows the results with oracle VAD but in the speech regions, speaker transition points need to be determined by the diarisation model. Since ECAPA-based diarisation used a 3-second window [2] and assumed a single active speaker, many short utterances and overlaps were ignored. Thus, the number of utterances and total duration notably decrease (the 2nd row of Table 5.5), leading to much higher deletion errors. Comparing Table 5.4 (c) and (d), unexpected improvements were observed when the ground-truth VAD was replaced by Wav2Vec2-based VAD. A possible reason is that the average utterance duration with Wav2Vec2-based VAD is longer (the last row of Table 5.5) and the ASR model performs better with more context. According to the observations above, it can be concluded that high deletion error is mainly caused by the diarisation model. There are two aspects: wrong speaker label prediction and the inability to handle overlaps.

Comparing the first and the second row in each sub-table of Table 5.4, it was found that source separation was not effective with ground-truth diarisation but for all other situations, the separation model improved cpWER-us. Moreover, source separation was most helpful in the fully automatic transcription system indicated by the highest cpWER-us reductions. It proves that source separation can mitigate the error made by VAD and diarisation when the boundaries between speakers are inaccurate.

[2] The window can be shorter than 3 seconds if the speech region is shorter than 3 seconds.

Chapter 6

Conclusion and Future Work

6.1 Summary

In this thesis, a source separation system was built based on SSL representations to facilitate ASR models' robustness against overlaps. First, basic setups were thoroughly investigated on simulated datasets. Several conclusions can be derived from the results: 1) Considering phase information using PSM is helpful. 2) Low-level acoustic features like spectrogram are difficult to process for a shallow downstream model, so it is better to use SSL representations solely. 3) Fine-tuning SSL models can enhance the performance but to maintain decent generalisability, it is crucial to first train the downstream model. Then experiments were carried out to compare four SSL models. Owing to the large training set and data augmentations, WavLM and UniSpeech-SAT outperformed TERA and Wav2Vec2, and the combined model between WavLM and TERA further improved the performance. Based on the results, it can be concluded that SSL using large-scale datasets not only enables the model to extract informative features but endows the model with better generalisation ability.

The concentration was to apply source separation to real overlapped speech corpus which matches poorly with the synthetic training set. Therefore, MixIT was adopted to perform in-domain fine-tuning with noisy targets. It significantly improved WERs on AMI corpus compared with PIT in terms of permutation invariant evaluations. The separation model outputs several sources which need to be selected automatically to apply it in a transcription system. To this end, source selection methods were explored. The iterative source selection approach surpassed the approach using input embedding by utilizing speaker information. Results also showed that more output sources required by MixIT make source selection more challenging. Experiments for the automatic transcription system suggested that source separation is effective if the ASR model was trained on non-overlapped speech. However, if

the ASR model has seen overlapped data during training, it failed to generalise to speech after separation. Fine-tuning the ASR model on separated data can solve this problem and resulted in cpWER-us of 33.6% and 32.3% on AMI test and development sets which outperformed the system without separation. The lowest cpWER-us of 33.0% and 31.8% on AMI test and development sets were achieved by combining hypotheses of multiple systems through ROVER.

In summary, SSL models can be efficiently fine-tuned for source separation and the effectiveness was verified on both simulated and real datasets. It was also proved that using source separation to improve ASR performance is practical.

6.2 Future Work

There are several directions for future work. First, AMI is primarily meeting recordings, so the model should be fine-tuned and tested on more challenging datasets like CHiME-6 (Watanabe et al., 2020) which includes diverse ambient noises. In this case, MixIT can be more useful as clean references are more difficult to obtain. Second, the transcription system can be further optimised. The current diarisation system ignores overlaps causing deletion errors. If the separation model can preprocess the continuous recordings before diarisation, this problem can be alleviated (Raj et al., 2021). However, source selection does not work without utterance boundaries and speaker information, so the diarisation system should be modified to handle multiple streams. Finally, no previous work has applied SSL models in time-domain source separation due to stride mismatch. Therefore, the architecture of time-domain downstream models can be explored. A possible structure may include temporal downsampling and upsampling (Tzinis et al., 2020) so that SSL representations can be incorporated at a suitable level.

References

Allen, J. B. and Berkley, D. A. (1979). Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, 65(4):943–950.

Anguera, X. (2008). Diarization error rate. *http://www.xavieranguera.com/phdthesis/node108.html*.

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *Proc. NIPS*.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). Wav2Vec 2.0: A framework for self-supervised learning of speech representations. *Proc. NIPS*.

Çetin, Ö. and Shriberg, E. (2006). Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition. *International conference on spoken language processing*.

Chen, G., Chai, S., Wang, G., Du, J., Zhang, W.-Q., Weng, C., Su, D., Povey, D., Trmal, J., Zhang, J., et al. (2021a). GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., et al. (2022a). WavLM: Large-scale self-supervised pre-training for full stack speech processing. *Journal of Selected Topics in Signal Processing*.

Chen, S., Wu, Y., Chen, Z., Wu, J., Li, J., Yoshioka, T., Wang, C., Liu, S., and Zhou, M. (2021b). Continuous speech separation with Conformer. *Proc. ICASSP*.

Chen, S., Wu, Y., Wang, C., Chen, Z., Chen, Z., Liu, S., Wu, J., Qian, Y., Wei, F., Li, J., et al. (2022b). UniSpeech-SAT: Universal speech representation learning with speaker aware pre-training. *Proc. ICASSP*.

Chen, Z., Yoshioka, T., Lu, L., Zhou, T., Meng, Z., Luo, Y., Wu, J., Xiao, X., and Li, J. (2020). Continuous speech separation: Dataset and analysis. *Proc. ICASSP*.

Chung, J. S., Nagrani, A., and Zisserman, A. (2018). VoxCeleb2: Deep speaker recognition. *Proc. Interspeech*.

Cosentino, J., Pariente, M., Cornell, S., Deleforge, A., and Vincent, E. (2020). LibriMix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. (2017). Language modeling with gated convolutional networks. *Proc. ICML*.

Dawalatabad, N., Ravanelli, M., Grondin, F., Thienpondt, J., Desplanques, B., and Na, H. (2021). ECAPA-TDNN embeddings for speaker diarization. *Proc. Interspeech*.

Delcroix, M., Zmolikova, K., Kinoshita, K., Ogawa, A., and Nakatani, T. (2018). Single channel target speaker extraction and recognition with speaker beam. *Proc. ICASSP*.

Desplanques, B., Thienpondt, J., and Demuynck, K. (2020). ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *Proc. Interspeech*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proc. NAACL-NLT*.

Erdogan, H., Hershey, J. R., Watanabe, S., and Le Roux, J. (2015). Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. *Proc. ICASSP*.

Erdogan, H., Hershey, J. R., Watanabe, S., and Roux, J. L. (2017). Deep recurrent networks for separation and recognition of single-channel speech in nonstationary background audio. pages 165–186.

Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). *Automatic Speech Recognition and Understanding Workshop*.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. (1992). Switchboard: Telephone speech corpus for research and development. *Proc. ICASSP*.

Graves, A., Fernández, S., and Schmidhuber, J. (2005). Bidirectional LSTM networks for improved phoneme classification and recognition. *International conference on artificial neural networks*.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *Proc. ICASSP*.

Griffin, D. and Lim, J. (1984). Signal estimation from modified short-time fourier transform. *Transactions on acoustics, speech, and signal processing*, 32(2):236–243.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. (2020). Conformer: Convolution-augmented transformer for speech recognition. *Proc. Interspeech*.

Habets, E. A. and Gannot, S. (2007). Generating sensor signals in isotropic noise fields. *Journal of the Acoustical Society of America*, 122(6):3464–3470.

Han, J. and Long, Y. (2022). Heterogeneous separation consistency training for adaptation of unsupervised speech separation. *arXiv preprint arXiv:2204.11032*.

Haykin, S. and Chen, Z. (2005). The cocktail party problem. *Neural computation*, 17(9):1875–1902.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proc. CVPR*.

Hershey, J. R., Chen, Z., Le Roux, J., and Watanabe, S. (2016). Deep clustering: Discriminative embeddings for segmentation and separation. *Proc. ICASSP*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021a). HuBert: Self-supervised speech representation learning by masked prediction of hidden units. *Transactions on Audio, Speech, and Language Processing*.

Hsu, W.-N., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., et al. (2021b). Robust Wav2Vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027*.

Huang, Z., Watanabe, S., Yang, S.-w., García, P., and Khudanpur, S. (2022). Investigating self-supervised learning for speech enhancement and separation. *Proc. ICASSP*.

Hung, K.-H., Fu, S.-w., Tseng, H.-H., Chiang, H.-T., Tsao, Y., and Lin, C.-W. (2022). Boosting self-supervised embeddings for speech enhancement. *arXiv preprint arXiv:2204.03339*.

Jang, E., Gu, S., and Poole, B. (2017). Categorical reparameterization with Gumbel Softmax. *Proc. ICLR*.

Kahn, J., Rivière, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., et al. (2020). LibriLight: A benchmark for ASR with limited or no supervision. *Proc. ICASSP*.

Kolbæk, M., Yu, D., Tan, Z.-H., and Jensen, J. (2017). Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *Transactions on Audio, Speech, and Language Processing*, 25(10):1901–1913.

Kraaij, W., Hain, T., Lincoln, M., and Post, W. (2005). The AMI meeting corpus.

Liu, A. T., Li, S.-W., and Lee, H.-y. (2021). TERA: Self-supervised learning of transformer encoder representation for speech. *Transactions on Audio, Speech, and Language Processing*, 29:2351–2366.

Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. *Proc. ICLR*.

Luo, Y., Chen, Z., and Yoshioka, T. (2020). Dual-path RNN: efficient long sequence modeling for time-domain single-channel speech separation. *Proc. ICASSP*.

Luo, Y. and Mesgarani, N. (2018). TasNet: time-domain audio separation network for real-time, single-channel speech separation. *Proc. ICASSP*.

Luo, Y. and Mesgarani, N. (2019). Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. *Transactions on audio, speech, and language processing*, 27(8):1256–1266.

Maciejewski, M., Wichern, G., McQuinn, E., and Le Roux, J. (2020). WHAMR!: Noisy and reverberant single-channel speech separation. *Proc. ICASSP*.

Nagrani, A., Chung, J. S., and Zisserman, A. (2017). VoxCeleb: A large-scale speaker identification dataset. *Proc. Interspeech*.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). LibriSpeech: an ASR corpus based on public domain audio books. *Proc. ICASSP*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*.

Raj, D., Denisov, P., Chen, Z., Erdogan, H., Huang, Z., He, M., Watanabe, S., Du, J., Yoshioka, T., Luo, Y., et al. (2021). Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis. *Spoken Language Technology Workshop*.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Sivaraman, A., Wisdom, S., Erdogan, H., and Hershey, J. R. (2022). Adapting speech separation to real-world meetings using mixture invariant training. *Proc. ICASSP*.

Snyder, D., Garcia-Romero, D., Povey, D., and Khudanpur, S. (2017). Deep neural network embeddings for text-independent speaker verification. *Proc. Interspeech*, 2017:999–1003.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Subakan, C., Ravanelli, M., Cornell, S., Bronzi, M., and Zhong, J. (2021). Attention is all you need in speech separation. *Proc. ICASSP*.

Sun, G., Zhang, C., and Woodland, P. C. (2021). Combination of deep speaker embeddings for diarisation. *Neural Networks*, 141:372–384.

Tzinis, E., Wang, Z., and Smaragdis, P. (2020). Sudo rm-rf: Efficient networks for universal audio source separation. *Machine Learning for Signal Processing Workshop*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Proc. NIPS*.

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., and Dupoux, E. (2021). VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *Proc. ACL*.

Wang, D. (2005). On ideal binary mask as the computational goal of auditory scene analysis. pages 181–197.

Wang, Y., Narayanan, A., and Wang, D. (2014). On training targets for supervised speech separation. *Transactions on audio, speech, and language processing*, 22(12):1849–1858.

Watanabe, S., Hori, T., Kim, S., Hershey, J. R., and Hayashi, T. (2017). Hybrid CTC/attention architecture for end-to-end speech recognition. *Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Watanabe, S., Mandel, M., Barker, J., Vincent, E., Arora, A., Chang, X., Khudanpur, S., Manohar, V., Povey, D., Raj, D., et al. (2020). CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. *Speech Processing in Everyday Environments Workshop*.

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Roux, J. L., Hershey, J. R., and Schuller, B. (2015). Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR. *International conference on latent variable analysis and signal separation*.

Wichern, G., Antognini, J., Flynn, M., Zhu, L. R., McQuinn, E., Crow, D., Manilow, E., and Roux, J. L. (2019). WHAM!: Extending speech separation to noisy environments. *Proc. Interspeech*.

Wisdom, S., Tzinis, E., Erdogan, H., Weiss, R., Wilson, K., and Hershey, J. (2020). Unsupervised sound separation using mixture invariant training. *Proc. NIPS*.

Wu, Z., Liu, Z., Lin, J., Lin, Y., and Han, S. (2020). Lite Transformer with long-short range attention. *Proc. ICLR*.

Yang, S.-w., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., et al. (2021). SUPERB: Speech processing universal performance benchmark. *Proc. Interspeech*.

Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A. E.-D., Jin, W., and Schuller, B. (2018). Deep learning for environmentally robust speech recognition: An overview of recent developments. *Transactions on Intelligent Systems and Technology*, 9(5):1–28.

Zheng, X., Zhang, C., and Woodland, P. C. (2022). Tandem multitask training of speaker diarisation and speech recognition for meeting transcription. *arXiv preprint arXiv:2207.03852*.

Appendix A

Dataset Details

Dataset		Duration (hr)	Segments	Ground-Truth
LibriMix (Cosentino et al., 2020)	Train	208.79	50800	audio
	Dev	7.60	3000	audio
	Test	7.01	3000	audio
LibriCSS (Chen et al., 2020)	Test	10.43	5023	transcription
SparseLibriMix (Cosentino et al., 2020)	Test	5.58	3000	audio
AMI (Kraaij et al., 2005)	Train	80.99	66418	transcription
	Dev	9.50	8665	transcription
	Test	9.17	7490	transcription
AMI-clean	Train	28.2	14575	audio
	Dev	0.98	512	audio
AMI-full	Train	34.6	16905	audio
	Dev	1.76	979	audio
Syn-AMI	Test	4.22	2087	audio

Table A.1 Dataset Details. LibriCSS, SparseLibriMix, and Syn-AMI are test-only datasets where LibriCSS and SpaseLibriMix provide subsets with different overlap ratios but Syn-AMI is fully overlapped. Only datasets with ground-truth audio can be used to train the separation model. AMI-clean was created from non-overlapped data in AMI whereas AMI-full utilised the full AMI dataset, so the ground-truth audio may contain multiple speakers. For SparseLibriMix, a segment contains multiple utterances. Otherwise, segments and utterances are equivalent.

Appendix B

Fine-Tuning of the ASR Model

W2V2-AMI: A linear layer was added on top of Wav2Vec2-Robust (Hsu et al., 2021b). The model was fine-tuned using Connectionist Temporal Classification (CTC) loss for 40 epochs on the AMI training set with 5% of data selected as the validation set. Adam optimizer was adopted with a tri-stage learning rate scheduler similar to the one in (Baevski et al., 2020). In the first 10% of steps, only the linear layer was updated, then the Transformer blocks were fine-tuned.

W2V2-AMI-Sep: The W2V2-AMI was further fine-tuned on separated speech of AMI training set for 10 epochs. In this case, we used ground-truth transcriptions to select between separated sources. To make W2V2-AMI and W2V2-AMI-Sep comparable, we also trained W2V2-AMI for another 10 epochs on the AMI training set without separation.