Weakly-supervised Audio Separation via Bi-modal Semantic Similarity: Supplementary Materials

Anonymous authors
Paper under double-blind review

A Overview of supplementary materials

We cover the following topics in our supplementary materials:

- All notations used in the paper are summarized in Appendix B.
- The proposed conditional U-Net architecture is detailed in Appendix C.
- The details of the training process for the proposed framework are illustrated in Appendix D.
- The datasets' details and the data prepossessing pipeline are presented in Appendix E.
- The details of evaluation metrics are presented in Appendix F.
- Additional experimental ablation studies are reported in Appendix G.
- And finally, a qualitative analysis of the model's performance can be found in Appendix I.

Table 1: The glossary of all notations used in the paper.

Type	Description	Notation
Scalar Parameters	Total number of mixtures in dataset	P
	Number of sampled mixtures in a batch	N
	Number of single source components in a mixture	K
	Temperature parameter in contrastive loss	τ
	Unsupervised reconstruction loss	$\mathcal{L}_{URL}$
	Contrastive loss	$\mathcal{L}_{CNT}$
	Consistency reconstruction loss	$\mathcal{L}_{CRL}$
	Total weak-supervision loss	$\mathcal{L}_{TWL}$
	Total semi-supervised learning loss	$\mathcal{L}_{SSL}$
	Weak supervision loss weights	α, β, γ
	Semi-supervised learning loss weights	λ_s, λ_u
Vectors/Matrices	Complete dataset	$\mathcal{D}$
	Batch of mixtures	$\mathcal{B}$
	Batch of synthetic mixture-of-mixturess	$\mathcal{B}'$
	i^{th} Sound mixture	$\mathcal{M}_i$
	Mixture of mixture	$\mathcal{M}'$
	Language prompt of i^{th} mixture	$\mathcal{T}_i$
	The k^{th} single source component in the i^{th} mixture	$\mathcal{S}_i^k$
	Language prompts of single source sound	$\mathcal{T}_i^k$
Models	Frozen CLAP language encoder	$\varepsilon_L(\cdot)$
	Frozen CLAP audio encoder	$\varepsilon_A(\cdot)$
	Conditional U-Net audio source separation model	$f_\theta(\cdot)$
	Conditional U-Net mask model	$g_\theta(\cdot)$
Operators/Functions	Magnitude function	$\lvert \cdot \rvert$
	Phase function	$\phi(\cdot)$
	Short Term Fourier Transform	$S(\cdot)$
	Softmax function with temperature parameter τ	$\zeta_\tau(\cdot)$
	Audio-language Cosine Similarity	c_{ikjt}
	L1 loss	$\lVert \cdot \rVert_{\ell_1}$
	Hadamard product	$\odot$

B The Notation Glossary

Table 1 presents the glossary of all notations used in the paper. We have divided the notations into four groups: scalars, vectors/matrices, models, and operators/functions.

C The Proposed Architecture

C.1 The Language-Conditional U-Net

To extract rich features for faithful reconstruction of the audio sources conditioned on the input prompt, we propose an enhanced conditional U-Net architecture. Our U-Net model operates on the magnitude spectrum of input mixtures, and estimates the segmentation mask for the corresponding source(s) based on conditional feature embedding. Prior works on conditional sound separation, mostly used unconditional U-Net with post conditioning on final generated features from the U-Net (Dong et al., 2022; Zhao et al., 2018). Some works (Gao & Grauman, 2019) used simple conditional feature concatenation at the innermost layer of the U-Net. Since most of these methods are primarily built for supervised separation, which is a much simpler problem, the vanilla U-Net architecture is often sufficient. However, since post-conditioning methods cannot leverage the conditional language features through the network, their performance can degrade significantly in the unsupervised setting. To overcome this, we redesign the conditional U-Net architecture by introducing multi-scale cross attention conditioning on the intermediate feature maps of the U-Net. The architecture is shown in Figure 2.

We incorporate three main building blocks into the proposed conditional U-Net: residual block (ResBlock), self-attention (SA), and cross-attention (CA) modules. The residual block is used for enhancing model capacity following He et al. (2016) at every scale of feature processing. For the input x, the operation can be represented by,

$$x = x + \text{ConvBlock}(x) \tag{1}$$

where *ConvBlock* represents two successive convolutional layers. The self-attention and cross-attention modules are designed using the multi-head attention (MHA) operations introduced by (Vaswani et al., 2017). Self-attention re-calibrates the feature space before applying the conditioning modulation. The self-attention operation for input x is given by,

$$x = x + \text{MHA}(Q = x, K = x, V = x) \tag{2}$$

where Q, K, V represent query, key, and values used in the *MHA* operation, respectively. In contrast, cross-attention selectively filters the relevant features based on conditional features. For condition embedding y with input x, the cross-attention mechanism is given by,

$$x = x + \text{MHA}(Q = x, K = y, V = y) \tag{3}$$

We divide the conditional U-Net model into two sub-networks: the *Head* network and the *Modulator* network. The Head network operates on the fine-grain features of the higher signal resolutions to generate coarse-grain features to be conditioned later by the language modality. Only ResBlocks with traditional skip connections are used at each scale of the Head network. In contrast, the Modulator network applies the feature modulation based on the conditional language embedding. We incorporate the self-attention and cross-attention operations in the skip connections of every Modulator network layer. In total, the U-Net contains 7 layers of encoder and decoder. The Head network contains top four layers of encoding and decoding, and the Modulator network contains the remaining three layers. Table 2 shows the architectural details of each block in our enhanced conditional U-Net.

C.2 The Inference Pipeline

For the inference, we use the similar pipeline as baseline methods. As shown in Figure 1, the conditional U-Net takes the input mixture and the language prompt (querying for the target source), and generates the (soft) magnitude mask. The mask is applied on the mixture's magnitude spectrogram, while the phase is directly copied from the input.

Table 2: Architectural details of proposed building blocks in the conditional U-Net model.

Block Type	**Parameters**	**Values**
Cross Attention Block (CA)	No. of attention heads	8
	No. of channels	512
	Head dimension	64
	Condition dimension	77x768
	No. of linear layers in attention	4
	Attention type	Softmax
	Normalization Layer	LayerNorm
	No. of linear layers in MLP	2
	MLP intermediate channels	1024
	MLP intermediate activation	GeLU
Self Attention Block (SA)	Num of attention heads	8
	Channel dimension	512
	Head dimension	64
	No. of linear layers	4
	Attention type	Softmax
	Normalization layer	LayerNorm
	No. of linear layers in MLP	2
	MLP intermediate channels	1024
	MLP intermediate activation	GeLU
Residual Block (ResBlock)	Conv kernel size	(3, 3)
	No. of convolutions	1
	Normalization layer	BatchNorm
	Activation	Leaky ReLU (th=0.2)
	Channel expansion ratio	1
Encoder Downsampler Module	Operator	Strided Convolution
	Kernel size	4x4
	Strides	2x2
	Channel expansion ratio	2
Decoder Upsampler Module	Spatial upsampler	Bilinear upsampling
	Scale	2.0
	Channel compressor	Convolution
	Channel compression ratio	2.0

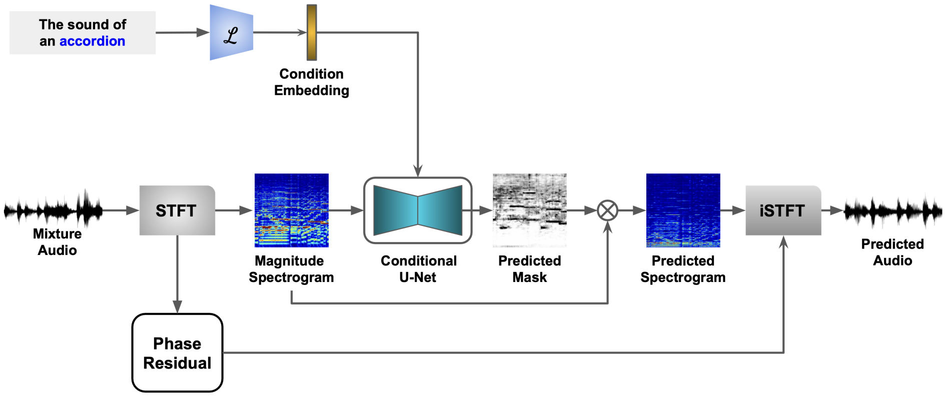


Figure 1: Inference pipeline for the proposed language conditional sound separation framework.

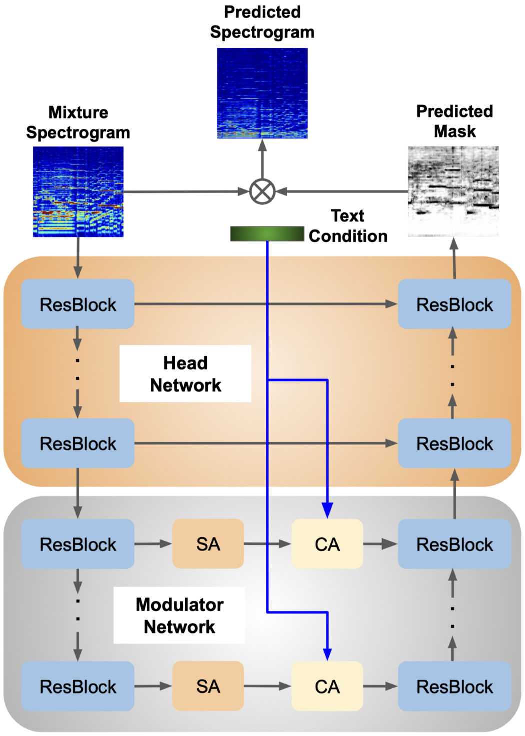


Figure 2: Proposed conditional U-Net architecture. We incorporate three building blocks: residual block (ResBlock), self-attention (SA), and cross-attention (CA) blocks. The model is divided into two modules: the head and the modulator. The *head* network operates on fine-grained features and generates latent embedding. The *modulator* network modulates latent features based on cross-attention conditioning.

D The Training Details

All the models are trained for 50 epochs with initial learning rate of 0.001. The learning rate drops by the factor of 0.1 after every 15 epochs. Adam optimizer (Kingma & Ba, 2014) is used with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ for backpropagation. All the training was carried out with 8 RTX-A6000 GPUs with 48GB memory. We validate the model after every training epoch. We use the batch size of 32 for the MUSIC dataset, and batch size of 64 for the VGGSound and AudioCaps datasets. We reproduce all baselines under the same settings. PyTorch library (Paszke et al., 2019)

Algorithm 1 The Proposed Weakly Supervised Training for Audio Source Separation

```
Input: Dataset $\mathcal{D} = \{\mathcal{M}_i, \mathcal{T}_i\}_{i=1}^{P}$, Single source text prompts $\{\mathcal{T}_i^k\}_{k=1}^{K}$, $K$: # of sources per
    mixture, masking U-Net $g_\theta$, Pre-trained joined embedding encoders $(\varepsilon_L, \varepsilon_A)$.
Require: Initialize weights of $g_\theta$, keep pre-trained joint embedding encoders $(\varepsilon_L, \varepsilon_A)$ frozen.
for $t \in [1, T]$ do                                                        ▷ T: Training iteration
    Sample $N$ Mixture with text prompts $\{\mathcal{M}_n, \mathcal{T}_n\}_{n=1}^{N} \in \mathcal{D}$               ▷ Batch size ← $N$
    for $n \in [1, N]$ do
        Sample another mixture $\{\mathcal{M}_m, \mathcal{T}_m\}$   ▷ No single sound source overlaps in $\mathcal{M}_n$ , $\mathcal{M}_m$
        Prepare MoM $\mathcal{M}' \leftarrow \mathcal{M}_n + \mathcal{M}_m$
        Predict $\widehat{\mathcal{M}}_n \leftarrow f_\theta(\mathcal{M}', \mathcal{T}_n)$
        Predict $\widehat{\mathcal{M}}_m \leftarrow f_\theta(\mathcal{M}', \mathcal{T}_m)$
        Compute $\mathcal{L}_{URL}(\mathcal{M}; \theta)$ with $(\mathcal{M}_n, \widehat{\mathcal{M}}_n; \mathcal{M}_m, \widehat{\mathcal{M}}_n)$ using  equation 2
        for $k \in [1, K]$ do
            Compute single source sound $\widehat{\mathcal{S}}_n^k \leftarrow f_\theta(\mathcal{M}_n, \mathcal{T}_n^k)$
            Compute single source sound $\widehat{\mathcal{S}}_m^k \leftarrow f_\theta(\mathcal{M}_m, \mathcal{T}_m^k)$
        end for
        Compute $\mathcal{L}_{CNT}(\mathcal{M}', \theta)$ with $\{\widehat{\mathcal{S}}_n^k, \mathcal{T}_n^k; \widehat{\mathcal{S}}_m^k, \mathcal{T}_m^k\}_{k=1}^{K}$ using  equation 3 and equation 4
        Reconstruct mixture $\widetilde{\mathcal{M}}_n \leftarrow \sum_{k=1}^{K} \widehat{\mathcal{S}}_n^k$ and $\widetilde{\mathcal{M}}_m \leftarrow \sum_{k=1}^{K} \widehat{\mathcal{S}}_m^k$
        Compute $\mathcal{L}_{CRL}(\mathcal{M}', \theta)$ with $(M_n, \widetilde{M}_n; \mathcal{M}_m, \widetilde{\mathcal{M}}_m)$ using equation 5
    end for
    Compute total loss $\mathcal{L}_{TWL}$ using $\mathcal{L}_{URL}$, $\mathcal{L}_{CRL}$, and $\mathcal{L}_{CNT}$ for all $N$ mixtures
    Back-propagate $\nabla \mathcal{L}_{TWL}$ and update weights of $g_\theta$
end for
```

is used to implement all the models. The complete training algorithm of the proposed framework is illustrated in Algorithm 1.

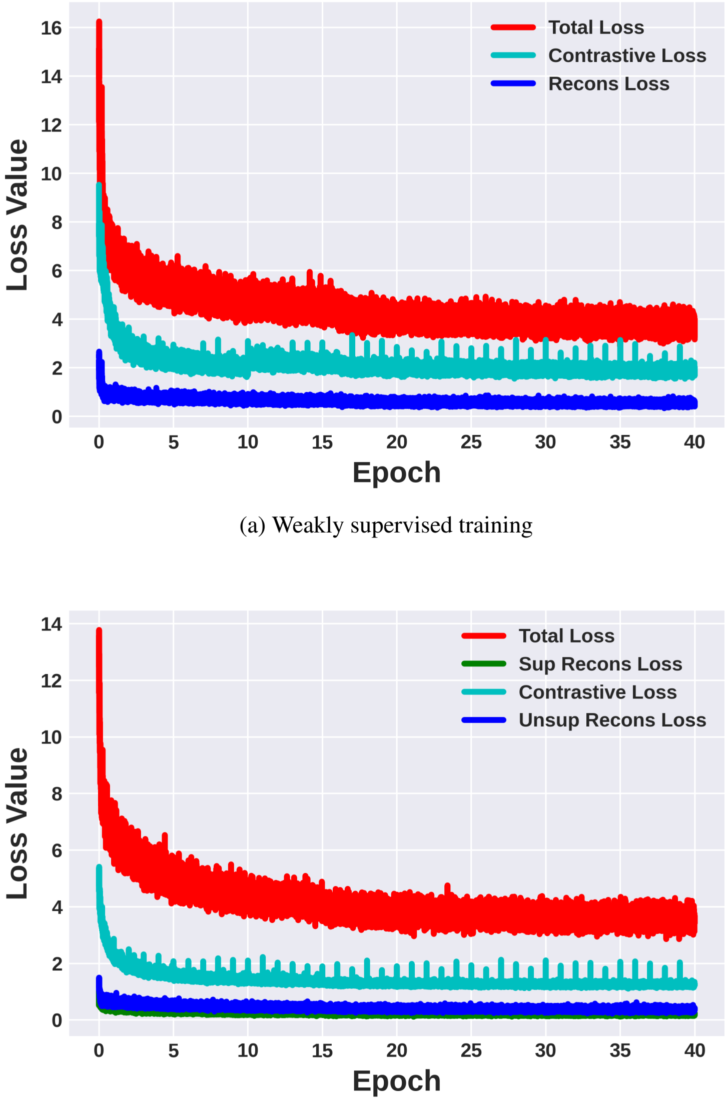
Furthermore, in Figure 3, we have visualized the detailed loss curves over the training course of the proposed weakly supervised training (Fig. 3a) and its semi-supervised flavor (Fig. 3b). We have combined both unsupervised reconstruction loss $\mathcal{L}_{URL}$ and consistency reconstruction loss ($\mathcal{L}_{CRL}$) in the reconstruction loss plot. For further analysis of the loss components as well as their weight hyper-parameter tuning details, please refer to Appendix G.2.

E Dataset Preparation

E.1 Datasets Description

MUSIC Dataset Following prior works, we experiment with MUSIC dataset (Zhao et al., 2018) for musical instrument separation task. Instead of using the original 11 instrument datasets, we use its extended version of MUSIC-21 containing around $1,200$ videos from 21 musical instruments. Since some videos are not available, our aggregated version contains $1,086$ videos in total. The video duration ranges $1 \sim 5$ minutes. We extract audios and class labels annotations from each video. We use 80% videos of each classes for training and the remaining for testing. For training, we randomly sample around 6s duration segments from each audio, while for testing, we prepare non-overlapping samples from the whole length audio. The dataset only contains sounds of single-source musical instruments. To use this datset for unsupervised training, we create synthetic training mixtures by sampling different combinations of K single source sounds. Text prompts are then generated using the class labels of the single source sounds. We use the common template for representing the single and multi-source language condition prompts, as presented in Table 4.

VGGSound Dataset VGGSound (Chen et al., 2020) is a large-scale environmental sound datasets containing more than $190,000$ videos from 309 classes. Since many corresponding videos are not available in YouTube, our aggregated subset contains $175,599$ videos. We use the official train and test split of VGGSound that contains $162,199$ training videos and $13,398$ test videos. Every video contains an audio, mostly with a single source. Each video duration is 10s that contains single



(a) Weakly supervised training

(b) Semi-supervised training

Figure 3: Visualization of loss curves over training epochs in proposed (a) weakly supervised training, and (b) semi-supervised training.

source audio collected from different environments corrupted by natural noise. The sounding event duration varies from $1 \sim 10$s. Because of that, we use the full-length audio samples in VGGSound for our experiments. In order to use VGGSound for the unsupervised learning scenario, we mix K random single source samples for each training mixture. The corresponding text prompts are generated using the class labels of the the sounding sources in the mixture, similar to Table 4.

AudioCaps Dataset AudioCaps (Kim et al., 2019) contains around $50,000$ *natural* sound mixtures of 10s duration each. It also includes the complete captions of the prominent sources in each mixture. We use the official train and test splits that contain $45,182$ and $4,110$ mixtures, respectively. In general, each mixture contains $1 \sim 6$ single source components. To cover all sounding events included in the text caption, we use full-length audios of 10s. We use Constituent-Tree library (Halvani, 2023) to extract fine-grain phrases representing each sounding source from the full caption. We initially extract several sentence and noun phrases, then perform simple post-processing on them to eliminate the overlapping phrases. Some examples of extracted phrases from the full captions are given in Table 3. To handle different number of mixture components, we sample a fixed number of phrases from each caption. In case there are not enough sounding phrases in the text prompt, we re-sample some of the phrases, and introduce weighted reconstruction to ensure proper reconstruction of the mixture.

AudioCaps is primarily used to measure training performance on natural mixtures containing diverse sounding events, as opposed to synthetic mixtures. However, to evaluate the performance of the model, we prepare synthetic mixture-of-mixtures by combining two mixtures from the AudioCaps test set. At the test time, the model is queried with one full-length caption representing one of the mixtures in synthetic MoM, and evaluated using the corresponding mixture.

E.2 Data preprocessing Pipeline

We use the sampling rate of 11kHz for audio samples in all datasets. Only mono-channel audio is used. The audio clip length is chosen to be $65,535$ for MUSIC dataset, and $110,000$ for the AudioCaps and VGGSound datasets. Since AudioCaps and VGGSound datasets are noisy, and usually contain sounding regions on small portion of 10s duration, we use full length audio samples for these two datasets. For the MUSIC dataset, we extract consecutive $65,535$ segments from the complete duration of the samples representing 6s audio. We compute the spectrogram for each sample using short-term Fourier transform (STFT) with a window size of 1024, a filter length of 1024, and a hop size of 256.

The CLAP model is pre-trained with 10s duration audios of 48KHz sampling rate and has different pre-processing pipeline than ours. To integrate the pretrained CLAP model in our training pipeline, we initially reconstruct the sound waveform from the predicted spectrogram. For audio samples extracted from MUSIC dataset that contains 6s duration segments, we repeat the waveform to extract equivalent 10s duration of audios. Then, the audio waveform is resampled with 48KHz sampling rate. We use Torchaudio package (Yang et al., 2022) to process predicted audio samples in the training loop. To estimate the contrastive loss ($\mathcal{L}_{CNT}$) with the CLAP model, we follow the same pipeline of CLAP with the pretrained temperature value for τ. We note that the CLAP model is kept frozen throughout the entire training, as it is only used to generate weak supervision. For text conditioning signals, instead of the projected mean-pooled token representation of language prompts, we use the complete language embedding of dimension (77×768) representing 77 tokens, generated by the CLAP language encoder.

F Evaluation Metrics

We use three evaluation metrics in our experiments: SDR, SIR, and SAR. Here, we provide the detailed equations as well as the explanation of each metric. We note that a predicted sound $\mathcal{S}_{pred}$ can be represented as a combination of the true sound $\mathcal{S}_{true}$, the interference of other sources in the mixture $\mathcal{E}_{interf}$, and the artifacts generated during reconstruction $\mathcal{E}_{artifact}$; that is, $\mathcal{S}_{pred} = \mathcal{S}_{true} + \mathcal{E}_{interf} + \mathcal{E}_{noise} + \mathcal{E}_{artifact}$. According to Vincent et al. (2006), these evaluation metrics are described as follows:

Table 3: Examples of some extracted phrases from full-length AudioCaps (Kim et al., 2019) captions.

Complete Audio Captions	**Extracted Phrases**
A young female speaks, followed by spraying and a female screaming	A young female speaks. Spraying and a female screaming
Motor noise is followed by a horn honking and a siren wailing	Motor noise. A horn honking. A siren wailing.
Rustling occurs, ducks quack and water splashes, followed by an adult female and adult male speaking and duck calls being blown	Rustling occurs. Ducks quack and water splashes. An adult female and adult male speaking. Duck calls being blown.
An audience gives applause as a man yells and a group sings	An audience gives applause. A man yells. A group sings.
A man speaks over intermittent keyboard taps	A man speaks. Intermittent keyboard taps.
An airplane engine runs	An airplane engine.

Signal-to-Distortion Ratio (SDR): SDR is the primary metric used for evaluating sound separation performance in most prior work. It represents the overall measure of the sound quality considering all kinds of distortions. It is given by

$$\text{SDR} = 10 \log_{10} \frac{||\mathcal{S}_{true}||^2}{||\mathcal{E}_{interf} + \mathcal{E}_{noise} + \mathcal{E}_{artifact}||^2} \tag{4}$$

Signal-to-Interference Ratio (SIR): SIR is also widely used evaluation metric in sound separation. It represents the ”leakage” or ”bleed” from other sounding sources in the mixture to the predicted sound. SIR measures the quality of the predicted sound considering the amount of cross-interference from other sources. It is given by

$$\text{SIR} = 10 \log_{10} \frac{||\mathcal{S}_{true}||^2}{||\mathcal{E}_{interf}||^2} \tag{5}$$

Signal-to-Artifact Ratio (SAR): SAR is mostly used to measure how realistic the predicted sound is. It measures the amount of synthetic artifacts present in the predicted audio. Without any separation applied, the original mixture usually have very high SAR, as it does not contain that many of artifacts. However, as the model learns to separate single source components from the mixture, it is expected to introduce more artifacts.

$$\text{SAR} = 10 \log_{10} \frac{||\mathcal{S}_{true} + \mathcal{E}_{interf} + \mathcal{E}_{noise}||^2}{||\mathcal{E}_{artifact}||^2} \tag{6}$$

In order to calculate these metrics, we have used the Python package **torch-mir-eval** (Montesinos, 2021) which is the Pytorch implementation of **mir-eval** (Raffel et al., 2014).

Table 4: Text query templates with examples for single and synthetic multi-source sounds

Source	**Query Template**	**Example**
Single Source	The sound of {source}	The sound of **guitar**.
Multi-Source (2-Source)	The sound mixture of {source-1} and {source-2}	The sound mixture of **guitar** and **piano**.
Multi-Source (3-Source)	The sound mixture of {source-1}, {source-2}, and {source-3}	The sound mixture of **guitar**, **piano**, and **violin**.

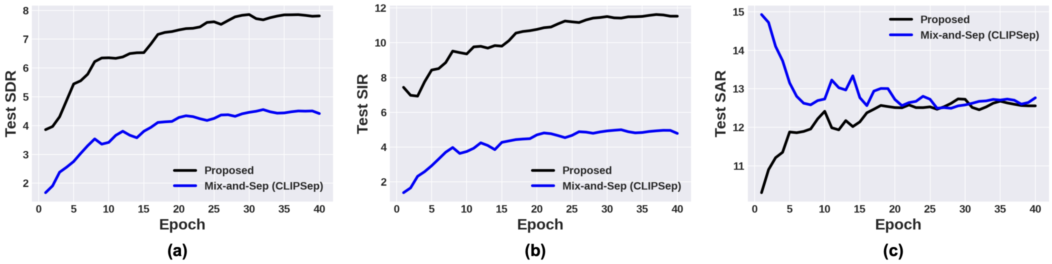


Figure 4: Test metric plots during training with two-source mixtures on MUSIC dataset. We use CLIPSep as the mix-and-separate baseline method. Our proposed method achieves significant improvement in terms of SDR and SIR by largely reducing noise and cross-interference in predictions, respectively. As the model tries to learn to separate single-source sounds, some artifacts are introduced, which in turn result in low SAR value in the early stages of training. However, most of these artifacts get removed over the course of training which in turn causes SAR to increase to the same level as the baseline method by the end of training. In other words, by the end of training, both our method and the baseline produce audio samples with reasonable quality regardless of their separation performance.

Table 5: Ablation on three building blocks of proposed conditional U-Net architecture: ResBlock, self-attention (SA), and cross-attention(CA). The vanilla U-Net contains single convolutional layer instead of ResBlock, and simple skip connections instead of attention modules. Test SDR on 2-source mixture is reported for various single and multi-source training scenarios on MUSIC dataset. For the single source training, simple *mix-and-separate* based on CLAPSep is used. For the multi-source training, proposed weakly supervised training is used. All three blocks contribute to considerable performance gain mostly in challenging multi-source scenarios. **Bold** and blue represents the best and second best performance in each group, respectively.

ResBlock	SA	CA	Total Params (M)	Single Source	2-Source	3-Source	4-Source
✗	✗	✗	30.7M	7.4	6.7	5.8	4.9
✗	✗	✓	37.4M	7.7	7.1	6.3	5.5
✗	✗	✓✓	44.7M	7.8	7.3	6.5	5.7
✗	✓	✗	36.6M	7.6	6.9	6.1	5.3
✗	✓✓	✗	42.9M	7.5	6.8	5.9	5.1
✓	✗	✗	74.3M	7.6	6.9	6.1	5.2
✓✓	✗	✗	118.8M	7.1	6.4	5.6	5.0
✗	✓	✓	43.7M	7.9	7.4	6.7	5.8
✗	✓✓	✓✓	57.7M	7.9	7.2	6.6	5.9
✓	✓	✓	81.4M	**8.1**	**7.9**	**7.1**	**6.2**

G Additional Experimental Studies

In this section, we present additional experimental ablation studies for a deeper analysis of the proposed framework compared to the state-of-the-art baseline methods as well as some of our design choices.

G.1 Ablation study on conditional U-Net architecture

We have studied the contribution of all three building blocks in the proposed conditional U-Net architecture. We have experimented with both supervised and unsupervised training settings on the MUSIC dataset. The test set contains 2-source mixtures as before. The baseline vanilla U-Net contains single convolutional layer instead of ResBlock, and direct skip connections instead of self-attention and cross-attention modules. The results are given in Table 5. Utilizing all three building blocks results in $+0.7$, $+1.2$, $+1.3$, and $+1.3$ SDR improvements on single source, 2-source, 3-source, and 4-source training settings, respectively. We note that the performance improvements are comparatively higher in the challenging unsupervised setting compared to the supervised setting.

Table 6: Ablation on loss components of proposed weakly supervised training method with multi-source training mixtures from the MUSIC dataset. Test SDR on 2-source mixtures is reported for all cases. Unsupervised reconstruction loss ($\mathcal{L}_{URL}$) underperforms in higher mixtures due to the lack of fine-grain supervision. Contrastive loss ($\mathcal{L}_{CNT}$), on the other hand, produces weak supervision that performs the best combined with proposed consistency reconstruction loss ($\mathcal{L}_{CRL}$). Combining all three loss components achieves significant performance improvements. **Bold** and blue represents the best and second best performance in each group, respectively.

$\mathcal{L}_{\mathbf{URL}}$	$\mathcal{L}_{\mathbf{CNT}}$	$\mathcal{L}_{\mathbf{CRL}}$	2-Source	3-Source	4-Source
✓	✗	✗	5.5	4.3	3.5
✗	✓	✗	4.9	2.8	1.7
✗	✗	✓	2.4	1.3	0.6
✗	✓	✓	6.8	5.9	5.3
✓	✓	✓	**7.9**	**7.1**	**6.2**

Table 7: Ablation study on loss weights with proposed weakly supervised and semi-supervised learning. For the weakly supervised training, 2-source mixtures from the (a) MUSIC dataset, (b) VGGSound dataset, and (c) natural mixtures from the AudioCaps dataset are used. (d) For the semi-supervised training, 5% single-source and 95% 2-source mixtures are used. Test SDR on 2-source mixture separation is reported. **Bold** and blue represents the best and second best performance in each group, respectively.

(a) Weakly supervised training on MUSIC

$\alpha(\mathcal{L}_{\mathbf{CNT}})$	$\beta(\mathcal{L}_{\mathbf{CRL}})$	$\gamma(\mathcal{L}_{\mathbf{URL}})$	**SDR**
0.1	5	5	7.7
0.1	5	10	**7.9**
0.1	5	15	7.7
0.2	5	10	7.8
0.1	10	10	7.6

(b) Weakly supervised training on VGGSound

$\alpha(\mathcal{L}_{\mathbf{CNT}})$	$\beta(\mathcal{L}_{\mathbf{CRL}})$	$\gamma(\mathcal{L}_{\mathbf{URL}})$	**SDR**
0.2	0.5	10	2.0
0.2	1	10	2.1
0.2	2	10	**2.2**
0.1	5	15	1.9
0.3	5	5	1.7

(c) Weakly supervised training on AudioCaps

$\alpha(\mathcal{L}_{\mathbf{CNT}})$	$\beta(\mathcal{L}_{\mathbf{CRL}})$	$\gamma(\mathcal{L}_{\mathbf{URL}})$	**SDR**
0.2	0.5	10	2.8
0.2	1	10	**2.9**
0.2	2	10	2.7
0.1	5	15	2.5
0.3	5	5	2.4

(d) Semi-supervised training on MUSIC

$\lambda_s(\mathcal{L}_{\mathbf{URL}})$	$\lambda_u(\mathcal{L}_{\mathbf{TWL}})$	**SDR**
5	0.5	8.5
5	1	**8.8**
5	2	8.6
2.5	1	8.7
0.5	1	8.2

Since unsupervised training is mostly guided by weak supervision generated by the bi-modal CLAP model, we hypothesize that the multi-scale feature modulation based on conditional embedding becomes more critical in such cases.

Note that the increased number of parameters by the way of adding new blocks can be seen as the major contributor to the performance gain. To control for this effect, in some of the ablation scenarios, we have inserted the target block(s) twice in a row merely to increase the capacity of the model (shown by ✓✓ in Table 5). As the results show, while increasing the number of model's parameters contributes to the performance gain, it is *not* the major driver. In fact, some of our smaller candidates beat the larger ones by simply incorporating a new block type. This shows that our proposed blocks encode important inductive bias for our problem which can boost the model performance without overfitting.

Furthermore, for a fair comparison with the baseline methods, we have reproduced most of the prior work with our improved U-Net architecture, as shown in Table 1 and 10. We observe consistent improvements of performance by leveraging our modified U-Net. Nonetheless, the improved U-Net architecture increases computational burden in general, but the proposed weakly supervised training can still be applied in resource constrained scenarios by simply using the vanilla U-Net architecture.

G.2 Effects of different loss components

We have also studied the effects of different loss components in the proposed framework under the challenging unsupervised settings, as shown in Table 6. By only using the unsupervised reconstruction loss ($\mathcal{L}_{URL}$), we get sub-optimal performance due to the training and test distribution shift. On the other hand, the contrastive loss by itself ($\mathcal{L}_{CNT}$) results in performance drops with significant spectral loss due to the lack of fine-grain supervision to reconstruct the target signal. Similarly, the consistency reconstruction loss($\mathcal{L}_{CRL}$) by itself suffers from convergence issues due to the lack of any supervision to encourage the model for conditional single source separation. In other words, since the final reconstruction administered by the consistency reconstruction loss ($\mathcal{L}_{CRL}$) greatly depends on the quality of single-source predictions, a significant performance drop is inevitable without using any single-source supervision. However, by combining $\mathcal{L}_{CNT}$ and $\mathcal{L}_{CRL}$, we achieve $+1.3$, $+1.6$, and $+1.8$ SDR improvements over the $\mathcal{L}_{URL}$-only scenario for 2-source, 3-source, and 4-source training settings, respectively. Furthermore, by combining all three losses, we achieve significant performance improvements of $+2.4$, $+2.8$, $+2.7$ SDR over the $\mathcal{L}_{URL}$-only approach for 2-source, 3-source, and 4-source training settings, respectively.

In addition to the elimination study of different loss terms, we have performed an ablation study on 2-component mixtures from the MUSIC and VGGSound datasets, as well as on natural mixtures of AudioCaps dataset, to find the optimal relative weights of these components (i.e. α, β, and γ in equation 6). Table 7(a), 7(b), and 7(c) shows these results. It is interesting to observe that, in the optimal setting, $\mathcal{L}_{CNT}$ is weighed two orders of magnitude less than $\mathcal{L}_{URL}$, which suggests that the weak-supervision mechanism in our framework acts as an effective regularizer while the backpropagated supervision signal is mostly dominated by the reconstruction error. And yet this relatively small regularization effect makes a significant improvement to the final performance of the model during inference. Moreover, VGGSound, and AudioCaps dataset contain significant amount of environmental noises that result in noisy training particularly with the consistency reconstruction losses. As a result, the corresponding weight γ of the consistency reconstruction loss $\mathcal{L}_{CRL}$ is relatively reduced in VGGSound and AudioCaps dataset, while $\mathcal{L}_{CNT}$ coefficient α is slightly increased for the best performance. Similar study has been performed to find the optimal relative weights of the supervised and weakly-supervised components for the semi-supervised loss ($\mathcal{L}_{SSL}$) (i.e. λ_s and λ_u in equation 7). The results are summarized in Table 7(b).

G.3 Analysis of evaluation metrics for unsupervised training

The metric plots in Figure 4 demonstrate comparative analysis of the evaluation metric curves during the unsupervised (2-source) training. To represent the baseline *mix-and-separate* framework, we have used the CLIPSep (Dong et al., 2022) method. The mix-and-separate baseline attempts to extract the single-source components from the mixture without having any supervision on single-source predictions during training; this results in large noise and cross-interference. In contrast, our proposed weakly-supervised method significantly reduces noise and cross-interference during unsupervised training by leveraging weak supervision through the language modality, and results in a much higher SDR (Figure 4(a) and SIR(Figure 4(b), respectively. However, separating single-source components is susceptible to producing artifacts that cause lower SAR in the early stages of training, as shown in Figure 4(c). Nevertheless, as the training continues, such artifacts are largely eliminated which subsequently improves SAR. In general, SAR represents an style metric measuring the amounts of artifacts presents in audio. Also note that SAR can be quite high even for an audio mixture that doesn't contain any artifact. Initial drops of SAR followed by subsequent improvements demonstrate that our proposed method attempts to learn to extract single-source components from the very early stages of training.

G.4 Performance comparison on additional datasets

We have also conducted additional comparisons between the proposed method and other baselines on VGGSound (Chen et al., 2020), and AudioCaps (Kim et al., 2019) datasets. Both datasets contain a large variety of sounding source categories as well as significant environmental noise types that make the single-source separation task particularly challenging. For a fair comparison, we have reproduced all the baselines under the same setting. Moreover, we have replaced the vanilla U-Net with our improved U-Net in most baselines.

Table 8: Performance comparison on **VGGSound Dataset** under supervised and unsupervised training scenarios. Same test set of 2-Source separation is used for all cases. All methods are reproduced under the same setting. * denotes implementation with our improved U-Net model. Our proposed method largely closes the performance gap between supervised and unsupervised settings. **Bold** and blue represents the best and second best performance in each group, respectively.

Method	**Single-Source (Supervised)**	**Multi-Source (Unsupervised)**	
		2-Source	**3-Source**
Unconditional			
PIT* (Yu et al., 2017)	2.1 ± 0.33	-	-
MixIT (Wisdom et al., 2020)	-	-1.7 ± 0.44	-2.9 ± 0.51
MixPIT (Karamatlı & Kırbız, 2022)	-	-1.4 ± 0.51	-3.1 ± 0.39
Image Conditional			
CLIPSep-Img (Dong et al., 2022)	1.3 ± 0.34	-0.5 ± 0.27	-1.2 ± 0.35
CLIPSep-Img* (Dong et al., 2022)	1.7 ± 0.36	0.4 ± 0.31	-0.6 ± 0.28
SOP* (Zhao et al., 2018)	1.6 ± 0.23	0.3 ± 0.41	-0.9 ± 0.26
Language Conditional			
CLIPSep-Text (Dong et al., 2022)	2.1 ± 0.26	0.8 ± 0.31	-0.1 ± 0.27
CLIPSep-Text* (Dong et al., 2022)	**2.5** ± 0.29	1.2 ± 0.44	0.5 ± 0.38
BertSep*	2.0 ± 0.27	0.7 ± 0.31	0.3 ± 0.22
CLAPSep*	2.3 ± 0.32	1.1 ± 0.36	0.5 ± 0.28
LASS-Net (Liu et al., 2022)	2.2 ± 0.31	0.9 ± 0.28	0.2 ± 0.29
Weak-Sup (Pishdadian et al., 2020)	-	0.6 ± 0.39	-0.8 ± 0.33
Proposed Framework	-	**2.2** ± 0.35	**1.7** ± 0.39

Table 9: Performance comparison on **AudioCaps Dataset** representing natural multi-source mixture training. Same test set of 2-Mixture separation is used for all cases. All methods are reproduced under the same setting. * denotes implementation with our improved U-Net model. Our proposed method significantly improves the performance over the baselines. **Bold** and blue represents the best and second best performance in each group, respectively.

Method	**Test SDR**
Image Conditional	
CLIPSep-Image (Dong et al., 2022)	-0.7 ± 0.47
CLIPSep-Image* (Dong et al., 2022)	0.4 ± 0.33
SOP* (Zhao et al., 2018)	0.2 ± 0.25
Language Conditional	
CLIPSep-Text (Dong et al., 2022)	0.7 ± 0.36
CLIPSep-Text* (Dong et al., 2022)	1.3 ± 0.31
BertSep*	0.9 ± 0.29
CLAPSep*	1.2 ± 0.41
LASS-Net (Liu et al., 2022)	0.8 ± 0.38
Proposed Framework	**2.9** ± 0.35

Table 10: SDR comparison on **3-source separation test set** form the MUSIC Dataset under supervised and unsupervised training scenarios. All methods are reproduced under the same setting. * denotes implementation with our improved U-Net model. **Bold** and blue represents the best and second best performance in each group, respectively.

Method	Single-Source (Supervised)	Multi-Source (Unsupervised)		
		2-Source	3-Source	4-Source
Unconditional				
PIT* (Yu et al., 2017)	2.3 ± 0.26	-	-	-
MixIT (Wisdom et al., 2020)	-	-2.3 ± 0.34	-3.1 ± 0.57	-4.2 ± 0.35
MixPIT (Karamatlı & Kırbız, 2022)	-	-1.9 ± 0.46	-2.8 ± 0.41	-3.9 ± 0.35
Image Conditional				
CLIPSep-Img (Dong et al., 2022)	0.7 ± 0.25	-0.8 ± 0.27	-1.7 ± 0.35	-2.9 ± 0.32
CLIPSep-Img* (Dong et al., 2022)	1.6 ± 0.22	0.1 ± 0.31	-0.9 ± 0.28	-1.8 ± 0.43
CoSep* (Gao & Grauman, 2019)	1.8 ± 0.28	0.4 ± 0.37	-0.2 ± 0.29	-0.7 ± 0.36
SOP* (Zhao et al., 2018)	1.3 ± 0.23	-0.5 ± 0.41	-1.6 ± 0.26	-2.6 ± 0.42
Language Conditional				
CLIPSep-Text (Dong et al., 2022)	1.8 ± 0.21	-0.2 ± 0.35	-1.1 ± 0.27	-2.1 ± 0.45
CLIPSep-Text* (Dong et al., 2022)	**2.4** ± 0.27	0.9 ± 0.41	0.3 ± 0.32	-0.4 ± 0.41
BertSep*	1.9 ± 0.27	0.4 ± 0.31	-0.2 ± 0.22	-1.1 ± 0.27
CLAPSep*	2.2 ± 0.31	0.6 ± 0.36	0.1 ± 0.28	-0.8 ± 0.33
LASS-Net (Liu et al., 2022)	2.1 ± 0.25	0.5 ± 0.26	0.3 ± 0.29	-0.9 ± 0.36
Weak-Sup (Pishdadian et al., 2020)	-	-1.1 ± 0.47	-2.3 ± 0.38	-3.2 ± 0.33
Proposed	-	**3.5** ± 0.35	**2.7** ± 0.42	**2.2** ± 0.38

Comparisons on VGGSound: We have conducted performance comparison on VGGSound dataset under supervised and unsupervised with synthetic, 2- & 3-source mixtures training scenarios, as shown in Table 8. We observe unconditional methods suffer from convergence issues during unsupervised training that results in significant performance drops. Conditional methods, on the other hand, achieve considerably higher performance in general compared to their unconditional counterparts. Since the dataset contains variable length of sounding events with large amount of noise components, image-conditional methods in general achieves lower SDR compared to language-conditional methods. However, we observe our method achieves 88%, 68% of the supervised method's performance on 2-source separation test set in 2-source and 3-source training scenarios, respectively. Furthermore, we achieve 1.83x, and 3.4x SDR improvements over the second best method in 2-source and 3-source training scenarios, respectively, while using the same model architecture.

Comparisons on AudioCaps: AudioCaps contains natural mixtures with $1 \sim 6$ single source components in each mixture which makes the separation task particularly challenging. To test on AudioCaps, we prepare a synthetic mixture-of-mixtures (MoM) test set by mixing random mixture pairs. Table 9 shows the comparison results. Due to severe convergence issues in unconditional methods, we only present comparisons for image and language conditional methods. Since the audio contains variable number of sounding sources with different durations, it becomes increasingly difficult to condition with images compared to text prompts which results in lower SDR for image conditional baselines. Nonetheless, our proposed method achieves superior performance outperforming the second highest baseline by achieving 2.3x SDR improvement under the same setting.

G.5 Comparisons on higher order mixture test sets

So far, we have shown all performance comparisons on a common test set of 2-source mixtures. Here, we present the same comparisons on more challenging test mixtures containing combinations of three single source components from the MUSIC dataset. The results are reported in Table 10. In general, we observe significant performance drops compared to 2-source test setting for all methods including ours. In particular, the mix-and-separate-based baselines suffer from 62.5% SDR drop in the supervised setting in the 2-source mixture training scenario. Interestingly, our weakly supervised approach outperforms the supervised method achieving 1.5x and 1.1x higher SDR on the 3-source test set when trained on 2-source, and 3-source mixtures, respectively. This result reveals

Table 11: Ablation on the effect of CLAP-constraint in supervised single-source training. Same test set of 2-Source separation is used for all cases. All methods are reproduced under the same setting. * denotes implementation with our improved U-Net model. Bi-modal semantic CLAP constraint introduces additional regularization in supervised training, which results in notable performance improvement. **Bold** and blue represents the best and second best performance in each group, respectively.

Method	MUSIC Dataset		VGGSound Dataset	
	w/o CLAP	w/ CLAP	w/o CLAP	w/ CLAP
CLIPSep-Text (Dong et al., 2022)	7.7 ± 0.21	8.2 ± 0.32	2.1 ± 0.26	2.5 ± 0.31
CLIPSep-Text* (Dong et al., 2022)	**8.3** ± 0.27	**8.8** ± 0.41	**2.5** ± 0.29	**2.9** ± 0.44
BertSep*	7.9 ± 0.27	8.3 ± 0.35	2.0 ± 0.27	2.5 ± 0.31
CLAPSep*	8.1 ± 0.31	8.7 ± 0.34	2.3 ± 0.32	2.8 ± 0.31

a key feature of our framework that is consistent with our observations through our other ablation studies; namely, the weak supervision proposed in our framework acts as an effective regularization mechanism, that can significantly improve the model's generalization, especially when we test it on the 3-source mixture set. In other words, the supervised method tends to overfit to the separation task on its training distribution, and that is why it experiences larger performance drop when the test distribution shifts. Whereas, in our framework, due to the inherent regularization properties of the weak supervision mechanism, the performance drop is less dramatic when the test distribution shifts.

G.6 Effect of Bi-modal CLAP constraint on supervised training

Apart from using bi-modal CLAP constraint as weak-supervision for multi-source (unsupervised) training, we study its impact on single source (supervised) training on the baseline methods. In particular, we add the bi-modal semantic CLAP-constraint in the form $\mathcal{L}_{CNT}$ loss to the mix-and-separate $\mathcal{L}_{URL}$ loss, while training using supervised single-source samples from the MUSIC and VGGSound datasets. The results are reported in Table 11. As the results show, there is a consistent SDR improvement across the board when we incorporate the CLAP constraint in supervised learning, even though, intuitively speaking, the weak supervision obtained from $\mathcal{L}_{CNT}$ should be impertinent in the presence of the strong supervision signal coming from the supervised loss. We hypothesize that the integration of CLAP constraint here introduces additional regularization to supervised training by transferring the knowledge obtained through CLAP's large-scale pre-training to the problem of audio source separation. This result further shows that our proposed framework not only boost the separation quality in unsupervised and semi-supervised training scenarios, it can also help the supervised training itself by introducing extra cross-domain regularization.

G.7 Additional comparisons for semi-supervised training

Table 12 depicts additional performance comparisons between supervised training on single source sounds, unsupervised training on multi-source mixture sounds, and proposed semi-supervised training on both single-source and multi-source mixture sounds. We split the MUSIC training dataset with different ratios for single-source and multi-source training as mentioned before. Multi-source mixtures are composed of two single-source components here. In general, semi-supervised training significantly outperforms both supervised and unsupervised training. With the increase in single source data portion, we note the performance improves in general. Similarly, unsupervised performance on multi-source mixtures also depend on available training data. Also note that the unsupervised performance with different splits of training data largely closes the performance gap in comparison with single-source supervised training, which is consistent with our prior observations. More notably, however, by combining both single-source and multi-source training mixtures in the proposed semi-supervised learning framework, we achieve considerable performance improvement compared to 100% supervised performance reaching 9.5 SDR, which is 28% higher than the 100% scenario for the supervised baseline. This result, again, suggests the regularization effects of the pro-

Table 12: Performance comparisons between supervised, unsupervised, and semi-supervised settings using both single source and multi source (2-source) training data from the MUSIC dataset. Test SDR on 2-source mixtures is reported. $x\%$ of training data is used for single-source supervised training, while $(1 - x)\%$ of data is used for multi-source weakly-supervised training with synthetic mixtures. CLAPSep is used for supervised training, while our proposed weakly supervised training is applied for the multi-source data. Lastly, the semi-supervised training is applied on the combination of single source and multi-source data. Semi-supervised learning consistently achieves better performance in all data splits. Training on single-source, multi-source, and joint single- and multi-source data are referred as “Supervised”, “Unsupervised”, and “Semi-supervised” method, respectively. **Bold** and blue represents the best and second best performance in each group, respectively.

Single Source Split	Two Source Split	SDR Performance		
		Supervised	Unsupervised	Semi-Supervised
-	100%	-	**7.9**	-
5%	95%	2.6	7.6	8.8
10%	90%	3.9	7.4	8.9
25%	75%	5.3	7.1	9.2
50%	50%	6.6	6.2	9.4
75%	25%	7.4	4.9	**9.5**
100%	-	**8.1**	-	-

Table 13: The effects of prompt tuning for our proposed framework. We use the 2-source training mixtures from MUSIC dataset. Test SDR on 2-source mixtures is reported. We initially separate several full length single source audio samples from each category for training learnable prompts. **Bold** and blue represents the best and second best performance in each group, respectively.

(a) Ablation study on learnable prompt length with number of training audio samples per category. Each full-length audio represents a single source audio. We extract several overlapping frames from audio samples to train the learnable text prompts.

Prompt Length	#Audios/Category	Test SDR
8	1	8.1
	2	8.4
	5	8.6
16	1	8.2
	2	8.5
	5	8.7
32	1	8.0
	2	8.6
	5	**8.8**

(b) Comparison between the learnable and the template-based prompts.

Prompt Type	Test SDR
Template-based	7.9
Learnable	8.8
OCT (Tzinis et al., 2023)	8.7
OCT++ (Tzinis et al., 2023)	**9.0**

posed framework which can significantly reduce the reliance on single-source data and supervised training for conditional sound separation.

G.8 Effects of prompt tuning for the CLAP Model

We primarily use the bi-modal CLAP model to generate weak supervision for single-source separation from the corresponding language entity. Since the CLAP model is trained on large corpora of audio-language pairs, it can effectively generate weak supervision signals for a target dataset based on hand-crafted language prompts. We have studied the performance impacts of the CLAP model customization on a target dataset by tuning language prompts with few-shot single-source reference audio samples. The results are given in Table 13. For this experiment, first we separate few samples of full-length single-source audio samples for each category to incorporate learnable language prompts instead of the template-based ones. We then randomly sample single-source audio

Table 14: Subjective evaluation performance analysis. * denotes implementation with our improved U-Net model. **Bold** and blue represent best and second best performance, respectively.

Training Data	Method	Correct (%) ↑	Wrong (%) ↓	Both (%) ↓	None (%) ↓
Supervised	CLIPSep-Text* (Dong et al., 2022)	71.1	0.9	26.9	1.1
Unsupervised	CLIPSep-Text* (Dong et al., 2022)	30.4	20.5	40.6	8.5
	Proposed Framework	68.9	1.5	27.4	2.2
Semi-supervised	Proposed Framework	**82.6**	0.4	16.2	0.8

segments from a hold-out dataset to train learnable language prompts. By learning such prompts, we can customize the CLAP model for our target dataset to generate more informative supervision signal. In Table 13a, we report the effects of the prompt lengths as well as the number of full-length audio samples per category on the 2-Source test set using the proposed weakly supervised training on 2-source mixtures. By using 5-shot single-source audio samples per category and the prompt length of 32, we achieve around 12% SDR improvement compared to the template-based prompts (Table 13b).

Apart from the text-based prompt tuning using CLAP model, our proposed framework can also integrate heterogeneous prompting with other cues of the target source. Following Tzinis et al. (2023), we experiment with heterogeneous training conditions, such as text description, signal energy, and harmonicity of the target sound for source separation. We use the hold-out single source samples (5/Category) for each category to estimate the additional cues for prompting target sounds in mixtures. The baseline OCT method performs on-par with our learnable text prompting technique (8.7 vs. 8.8). We note that OCT with the embedding refinement approach (OCT++) achieves the best performance of 9.0 SDR. Hence, our proposed framework can effectively integrate advanced prompting techniques to separate the target sounds from the mixture.

H Subjective Evaluation

We conduct a subjective human evaluation to compare different models' performances based on human perception. Following prior work (Zhao et al., 2018; Dong et al., 2022), we have randomly sampled separated sounds from 2-source mixtures and presented them to the evaluators, who are then asked "Which sound do you hear? 1. A, 2.B, 3. Both, or 4. None of them". Here, A and B are replaced by the single-source sounding entities present in the input mixture, *e.g.* A. cat meowing, B. pigeon, dove cooing. In Table 14, we present the percentages of predicted samples that are correctly identified by the evaluator as the source class (Correct), which are incorrectly perceived by the evaluator (Wrong), which contains audible sounds of both sources (Both), and which doesn't contain any of the target sounds (None). We use the same 30 sample predictions on $2-$source mixture test sets for comparing models trained with supervised single-source, unsupervised multi-source, and semi-supervised single with multi-source data. 20 human evaluators have participated in this evaluation. We use the CLIPSep (Dong et al., 2022) method as the competitive baseline of the *mix-and-separate* framework with the text prompts.

As the results show, our proposed framework improves over the CLIPSep baseline's correct percentage statistics of 30.4% in the unsupervised setting by more than twice, reaching 68.9% and almost closing the gap with the performance of CLIPSep under the supervised regime (*i.e.* 71.1%). Furthermore, our framework's performance under the semi-supervised training setup goes even beyond that of the supervised setting by significantly reducing the number of under-separated instances from 26.9% to 16.2%, leading to 11.5% increase in correct percentage statistics to the total of 82.6%. This result shows the efficacy of our weakly-supervised training strategy under both unsupervised and semi-supervised training regimes. But more importantly, these results are consistent with our quantitative evaluation results, which further corroborate our conclusions.

I Qualitative comparisons

In this appendix, we present qualitative comparisons between the proposed method and the mix-and-separate approach represented by the CLIPSep Dong et al. (2022) framework under the unsupervised training scenario. The MUSIC dataset is used for this analysis. The models are trained and tested using 2-source mixtures. The results are given in Figure 5 - Figure 8. Due to the lack of single-source supervision in the mix-and-separate approach, most of its predictions exhibit significant spectral leakage, and large cross-interference. In contrast, our proposed method significantly reduces the spectral loss and cross interference. Also, separation under challenging cases of spectral overlap produces reasonable performance. These examples demonstrate the effectiveness of the proposed weakly supervised training method in disentangling single-source audio components form the input mixture.

REFERENCES

Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 721–725. IEEE, 2020.

Hao-Wen Dong, Naoya Takahashi, Yuki Mitsufuji, Julian McAuley, and Taylor Berg-Kirkpatrick. Clipsep: Learning text-queried sound separation with noisy unlabeled videos. *arXiv preprint arXiv:2212.07065*, 2022.

Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3879–3888, 2019.

Oren Halvani. Constituent Treelib - A Lightweight Python Library for Constructing, Processing, and Visualizing Constituent Trees., 3 2023. URL `https://github.com/Halvani/constituent-treelib`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ertuğ Karamatlı and Serap Kırbız. Mixcycle: Unsupervised speech separation via cyclic mixture permutation invariant training. *IEEE Signal Processing Letters*, 29:2637–2641, 2022.

Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 119–132, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Xubo Liu, Haohe Liu, Qiuqiang Kong, Xinhao Mei, Jinzheng Zhao, Qiushi Huang, Mark D Plumbley, and Wenwu Wang. Separate what you describe: Language-queried audio source separation. *arXiv preprint arXiv:2203.15147*, 2022.

Juan F. Montesinos. Torch-mir-eval: Pytorch implementation of mir-eval, 2021. URL `https://github.com/JuanFMontesinos/torch_mir_eval`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Fatemeh Pishdadian, Gordon Wichern, and Jonathan Le Roux. Finding strength in weakness: Learning to separate sounds with weak supervision. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2386–2399, 2020.

Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. Mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, volume 10, pp. 2014, 2014.

Efthymios Tzinis, Gordon Wichern, Paris Smaragdis, and Jonathan Le Roux. Optimal condition training for target source separation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey. Unsupervised sound separation using mixture invariant training. *Advances in Neural Information Processing Systems*, 33:3846–3857, 2020.

Yao-Yuan Yang, Moto Hira, Zhaoheng Ni, Artyom Astafurov, Caroline Chen, Christian Puhrsch, David Pollack, Dmitriy Genzel, Donny Greenberg, Edward Z Yang, et al. Torchaudio: Building blocks for audio and speech processing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6982–6986. IEEE, 2022.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 241–245. IEEE, 2017.

Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 570–586, 2018.

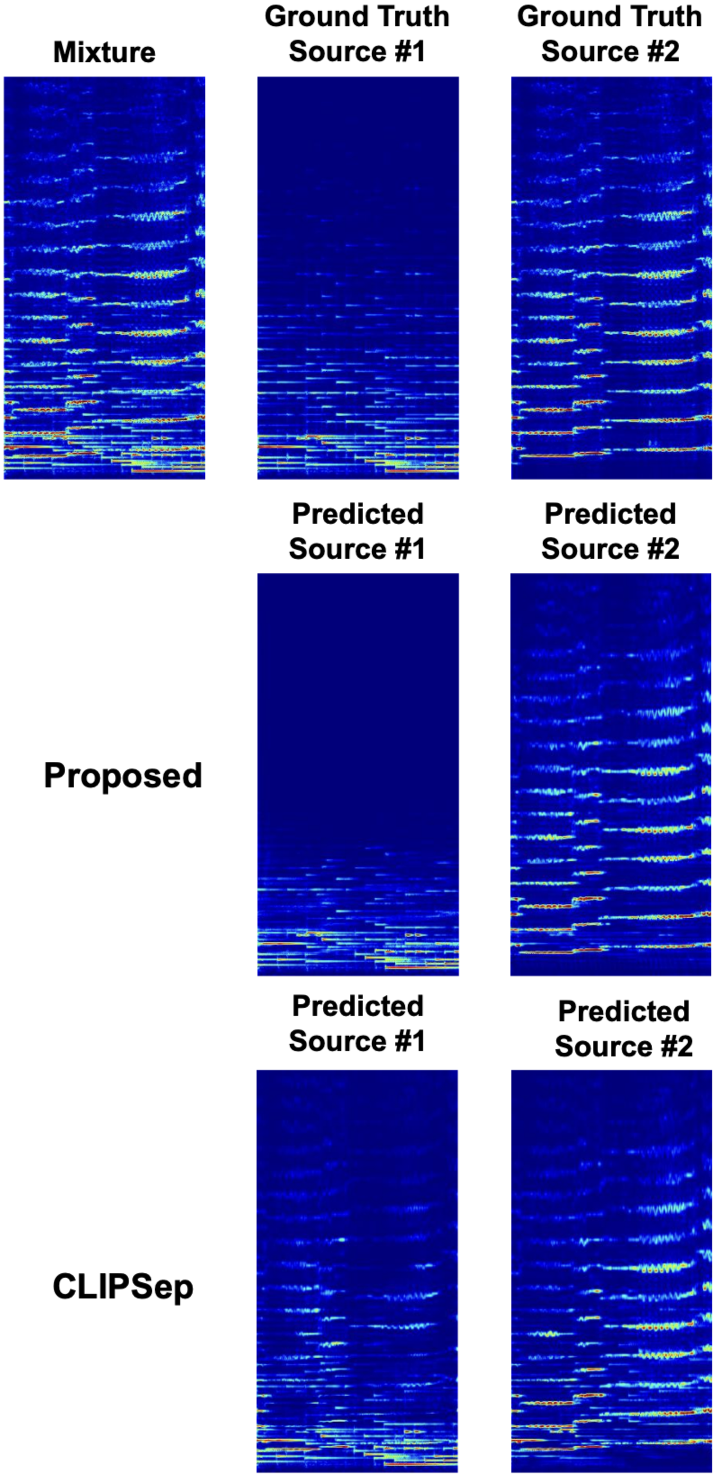


Figure 5: Qualitative comparisons between the proposed method and the mix-and-separate approach (CLIPSep (Dong et al., 2022)): The input mixture contains *piano* (source 1) and *violin*(source 2) sounds. For the lack of single source supervision in CLIPSep, large cross-interference is visible in its prediction for the *piano* source . In contrast, our method significantly reduces cross-interference.

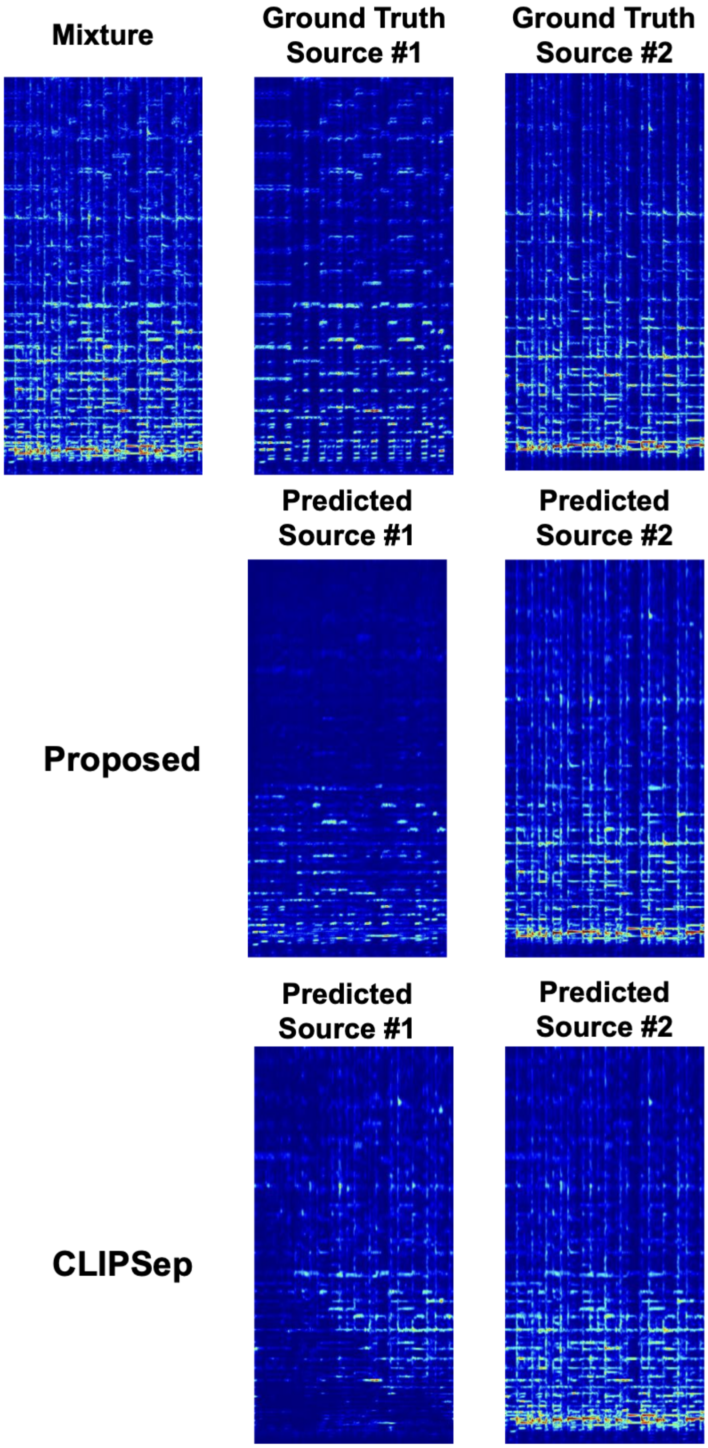


Figure 6: Qualitative comparisons between the proposed method and the mix-and-separate approach (CLIPSep (Dong et al., 2022)): The input mixture contains *accordion* (source 1) and *ukulele*(source 2) sounds. For *accordion* sound separation, CLIPSep exhibits significant spectral loss, while for *ukulele* sound separation, it shows cross-interference. However, our method largely reduces both the spectral loss for *accordion* sound segmentation and the cross-interference for *ukulele* sound segmentation.

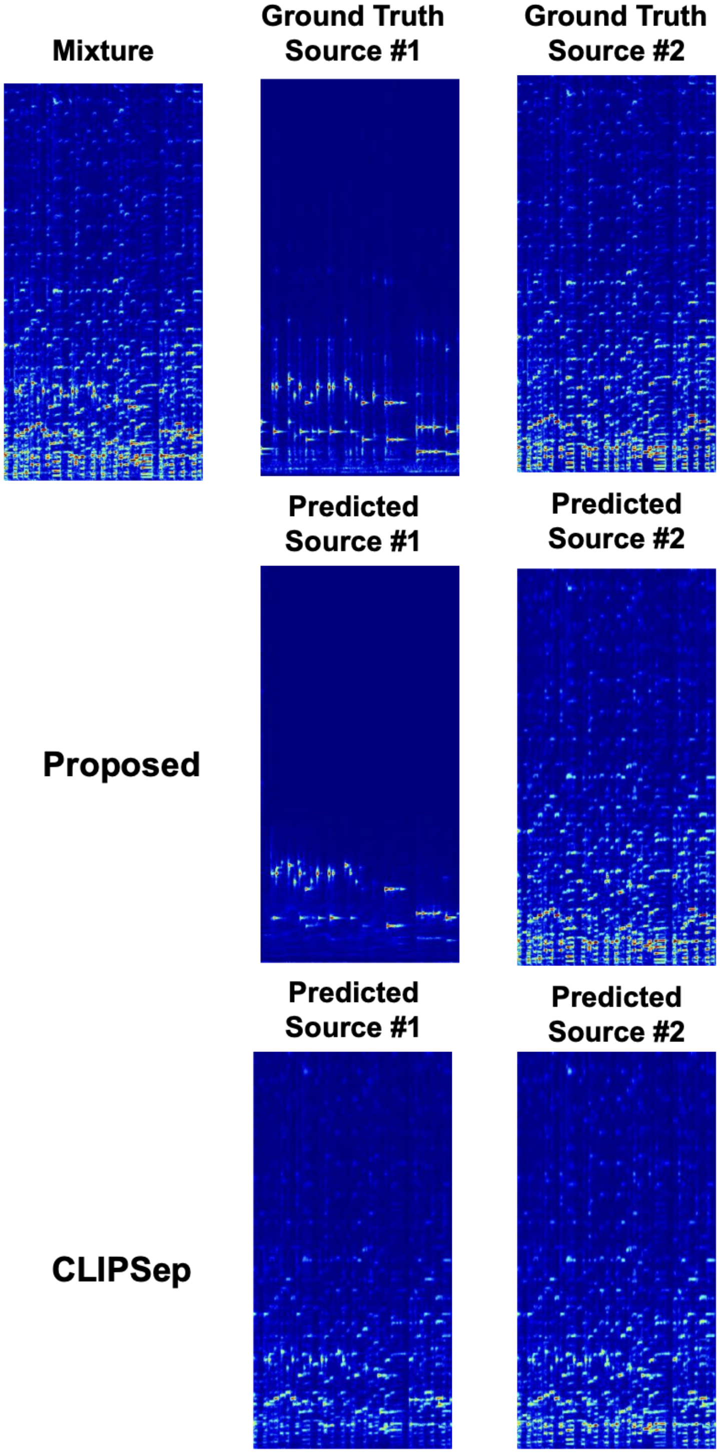


Figure 7: Qualitative comparisons between the proposed method and the mix-and-separate approach (CLIPSep (Dong et al., 2022)): The input mixture contains *xylophone* (source 1) and *accordion*(source 2) sounds. CLIPSep can hardly differentiate between these two sources due to a large spectral overlap. Our method, however, reasonably separates the two sounds despite showing some spectral loss for the *accordion* prediction.

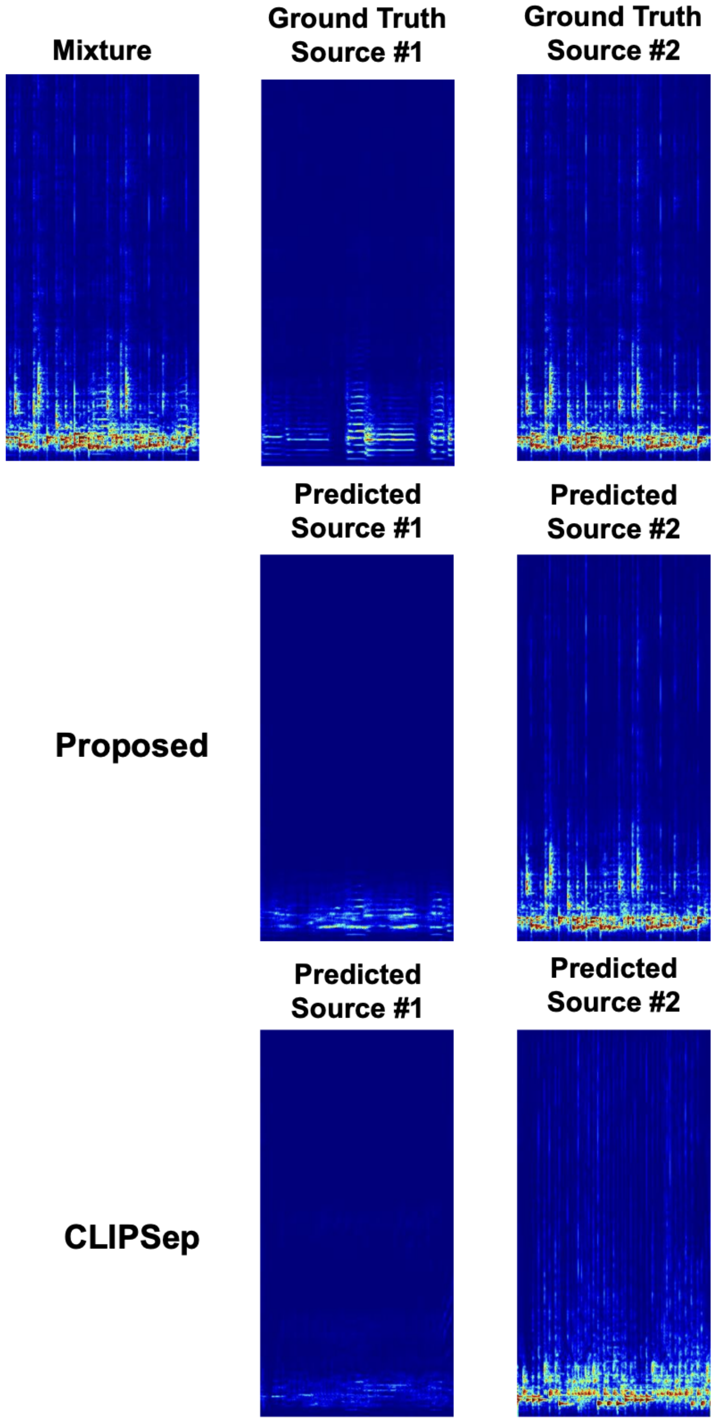


Figure 8: Qualitative comparisons between the proposed method and the mix-and-separate approach (CLIPSep (Dong et al., 2022)): The input mixture contains *tuba* (source 1) and *congas*(source 2) sounds. CLIPSep cannot properly identify the *tuba* sound due to the lack of single source training. Whereas, our method achieves considerable results in separating the *tuba* sound. nevertheless, some spectral loss can be observed for the *tuba* sound separation, which is reasonable considering the significant spectral overlaps of the two sounds.