Music Separation Using Self-Supervised Learning

Yunyoung Lee
Department of Physics
Team 4
2019160102

Minsu Son
Department of Physics
Team 4
2019160101

Yejin Hong
Cyber Defense
Team 4
2021350218

Abstract

We propose a novel self-supervised learning framework for music source separation leveraging Transformer architectures, specifically Performer, to address limitations of local contextual processing inherent in existing methods like MixIT and ConvTasNet. Our approach eliminates the reliance on labeled data by generating effective training signals using self-supervised techniques within the MixIT paradigm. We replace ConvTasNet with Performer enhanced by FAVOR+, allowing efficient global context learning from long audio sequences. Additionally, Audio Masked Autoencoder (Audio MAE) pre-training facilitates extraction of robust global audio features.

Experiments conducted on the MOISESDB dataset, limited to mixtures with up to four stems, evaluated separation performance at segment lengths of 3, 5, 10, and 30 seconds using the Scale-Invariant Signal-to-Noise Ratio (SI-SNR). Results indicate that Performer consistently outperforms ConvTasNet in SI-SNR for equivalent separation times, with significant improvements observed as segment lengths increased. Furthermore, Performer demonstrated superior computational efficiency, reducing overall training time.

However, due to dataset size constraints, negative SI-SNR values emerged, highlighting issues related to overfitting. Despite this, our findings underscore the Performer model's capacity for capturing global context in complex musical patterns. Addressing performance limitations identified herein will require training on significantly larger datasets.

1. Introduction

Audio separation is an essential problem in audio processing, with significant practical implications in various domains ranging from music production to speech enhancement. The ability to effectively isolate individual sources from mixed audio signals enables numerous advanced audio applications.

1.1. Problem Definition and Applications

Audio separation, defined as isolating distinct sound sources from mixed audio, is pivotal for diverse applications including music remixing, vocal isolation, and background noise suppression. Extensive research aims to improve the accuracy and efficiency of source separation. However, conventional supervised methods heavily rely on extensive labeled datasets, resulting in high costs and limited scalability.

1.2. Self-Supervised Learning for Audio Separation

To overcome these challenges, Self-Supervised Learning (SSL) leverages unlabeled audio data, enabling models to autonomously generate meaningful training signals. This significantly reduces annotation costs and enhances model generalization capabilities. SSL is particularly beneficial for audio separation given the abundance of unlabeled audio data readily available.

1.3. Limitations of CNN-Based Approaches

Previous research has predominantly focused on combining MixIT with CNN-based models such as ConvTasNet. While ConvTasNet effectively captures local features in audio signals, it falls short in modeling global contextual relationships essential for music, which typically involves extended temporal patterns like rhythm, melody, and harmony. Therefore, effective music separation necessitates models capable of robustly capturing global audio contexts.

1.4. Transformer-Based Self-Supervised Audio Separation

Addressing these limitations, we hypothesize that Transformer architectures, specifically Performer enhanced by FAVOR+, are well-suited to capturing global musical patterns. We propose integrating Performer into the MixIT framework, further enhanced by self-supervised pre-training using an Audio Masked Autoencoder (Audio MAE). FAVOR+ significantly reduces computational complexity, enabling efficient handling of long audio sequences, while Audio MAE pre-training strengthens global audio feature representation.

Our primary contributions are:

- Proposing a Transformer-based SSL model specifically designed for effective music separation.
- Empirically validating that our proposed model achieves superior separation performance (measured by Scale-Invariant Signal-to-Noise Ratio, SI-SNR) and higher computational efficiency (in terms of training time and resource utilization) compared to traditional CNN-based models.
- Identifying performance limitations due to dataset constraints and highlighting future research directions involving large-scale datasets to further enhance model robustness.

2. Related Works

2.1. MixIT: Mixture of Mixtures Training Framework

MixIT is a widely adopted self-supervised framework for training audio source separation models using unlabeled data. By randomly mixing multiple audio sources, a new composite signal is generated, allowing the model to learn separation without access to ground-truth source labels. MixIT loss evaluates all possible combinations of the model's N output signals against the mixture and selects the optimal assignment that maximizes signal-to-noise ratio (SNR), thus guiding the training process.

This framework has been predominantly used with CNN-based architectures such as ConvTasNet.

2.2. CNN-Based Audio Separation: ConvTasNet

ConvTasNet is a time-domain audio separation model using a 1D convolutional encoder-decoder architecture. It avoids explicit time-frequency transformation and excels at modeling local temporal patterns in audio. Despite its success, ConvTasNet's convolutional nature limits its ability to capture long-range dependencies, making it less suitable for music signals characterized by global structures like rhythm, harmony, and melody.

2.3. Transformer Models for Audio Processing

Vision Transformer (ViT): Originally proposed for image tasks, ViT processes images as sequences of patches, enabling strong global context modeling. Its strength in learning long-range dependencies makes it a promising candidate for audio tasks involving complex temporal dynamics.

Performer: The Performer model replaces the standard self-attention mechanism in Transformers with FAVOR+ (Fast Attention Via Positive Orthogonal Random Features), reducing the computational complexity from quadratic to linear with respect to sequence length. This makes Performer highly efficient for long-sequence audio processing, such as music.

Audio MAE: Audio Masked Autoencoder (Audio MAE) is a self-supervised Transformer model that masks parts of the input audio signal and learns to reconstruct them. Similar to ViT, it captures global representations of audio content, enabling powerful feature extraction even in the absence of labels.

2.4. SSL and Transformer Applications in Audio

Prior work in self-supervised audio learning has focused largely on speech separation and enhancement. Transformer-based models have demonstrated promising results in these areas. However, research on applying Transformer architectures to music separation remains limited, especially within the MixIT paradigm.

2.5. Distinctions of Our Approach

Our work is distinguished by the integration of Performer and Audio MAE into the MixIT framework for music separation. Unlike prior efforts limited to CNNs or focused solely on speech, we explore how self-supervised Transformers perform on music separation tasks across various segment lengths. We also analyze training efficiency and the effect of segment duration on separation quality, establishing a foundation for scaling to larger and more diverse music datasets.

3. Proposed Method

3.1. Overall Architecture Overview

MixIT Framework Overview. Given an unlabeled mixture signal $x = s_1 + s_2$, the goal is to estimate N separated signals $\{\hat{s}_i\}_{i=1}^{N}$. The MixIT framework allows self-supervised training by matching combinations of model outputs to the original mixtures using MixIT loss.

Segment-wise Processing. Each input music signal is segmented into fixed-length chunks (e.g., 3s, 5s, 10s, 30s). These segments are treated as independent training samples. This segmentation facilitates the learning of both local and global patterns by the model.

Performer-based Separation Network. Each segment $\mathbf{x} \in \mathbb{R}^T$ (where T denotes the number of audio samples in that segment) is processed by a Performer model utilizing FAVOR+ self-attention.

- *FAVOR+ Self-Attention.* Unlike traditional softmax-based attention with $\mathcal{O}(L^2)$ complexity, FAVOR+ uses orthogonal random features to linearize the attention computation to $\mathcal{O}(L \cdot r)$, where L is the sequence length and r is the

number of random features. This allows efficient processing of long audio sequences.
- *Audio MAE Embedding Layer.* During pretraining, portions of the input audio are masked and passed to a Transformer encoder which performs reconstruction of the masked frames. This enables the network to learn robust global audio representations.
- *Soft Masking and Separation.* Soft masks represent the relative strength of each source in every time–frequency bin. These masks are applied to the input spectrogram to produce separated spectrograms, which are then converted back to time-domain waveforms via inverse STFT.
- *Transcription to Separation.* The Transformer (Performer) block extracts source-specific features, and a decoder reconstructs the time-domain signals.

MixIT Loss. For the model outputs $\{\hat{s}_i\}$ and the original mixture group (x_1, x_2), all possible combinations are evaluated using the Scale-Invariant Signal-to-Noise Ratio (SI-SNR). For each pair of true source s and estimated signal $\hat{s}$, SI-SNR is computed as follows:

$$s_{\text{target}} = \frac{\langle \hat{s},\, s \rangle}{\|s\|^2}\, s, \tag{1}$$

$$e_{\text{noise}} = \hat{s} - s_{\text{target}}, \tag{2}$$

$$\text{SI-SNR}(\hat{s}, s) := 10 \log_{10} \frac{\|s_{\text{target}}\|^2}{\|e_{\text{noise}}\|^2}. \tag{3}$$

Here, $\langle \hat{s},\, s \rangle$ denotes the inner product between $\hat{s}$ and s, and $\|\cdot\|$ denotes the Euclidean norm. The optimal matching—i.e., the assignment of estimated sources to true mixtures that maximizes the sum of SI-SNRs—is selected. MixIT loss is then computed (e.g., via MSE on the best-matched assignment) to train the network.

3.2. Detailed Performer + Audio MAE Architecture

Audio MAE Pretraining.

- The input signal x is divided into short, fixed-size frames. Each frame is embedded into a token vector.
- A fraction $p\%$ of these frame tokens are randomly masked (either zeroed out or replaced with noise).
- The remaining unmasked tokens are passed through a Performer-based encoder. The model's objective is to reconstruct the original (masked) frame tokens.
- Reconstruction loss is computed by mean squared error (MSE) between the predicted tokens and the true (unmasked) tokens.
- After pretraining on a large corpus of unlabeled audio, the encoder's weights capture robust global audio patterns.

Performer Network Details.

- *Input Embedding:* A 1D convolutional encoder transforms raw audio samples of length T into a sequence of L embedding tokens.
- *FAVOR+ Attention:* Orthogonal random features Q', K' are generated to approximate the usual dot-product attention:

$$\text{Attention}(Q, K, V) \approx \phi(Q)\,\big(\phi(K)^{\mathsf{T}} V\big),$$

  where $\phi(\cdot)$ is an orthogonal random feature mapping. This reduces complexity from $\mathcal{O}(L^2)$ to $\mathcal{O}(L \cdot r)$, with r being the number of random features.
- *Feed-Forward Network:* A standard Transformer feed-forward block (two linear layers with activation in between).
- *Layer Normalization & Residual Connections:* Applied at each sub-layer to stabilize training.

training MixIT.

- The Performer encoder pretrained via Audio MAE initializes the separation network.
- During training, each input segment is converted to a spectrogram via STFT. A "soft mask" head predicts N masks over the time–frequency representation.
- The masked spectrograms are converted back into time-domain signals via inverse STFT.
- MixIT loss is computed to match the N outputs $\{\hat{s}_i\}$ to the two original mixtures (x_1, x_2). Gradients are back-propagated through the full Performer+decoder network.

3.3. Hyperparameters and Training Configuration

MixIT Configuration.

- Number of masks: $M = 8$ (i.e., $N = 8$ separated streams).
- Batch size: 4.
- Learning rate: 10^{-4}.
- Optimizer: AdamW (weight decay = 0.01).

Segment Settings.

- Segment lengths: $\{3\,\text{s},\, 5\,\text{s},\, 10\,\text{s},\, 30\,\text{s}\}$.
- Sampling rate: 44.1kHz.

Audio MAE Configuration.

- Masking ratio: $p = 30\%$.
- Number of encoder (Performer) layers: 12.
- Number of random features r: 256.

Hardware Setup.

- GPU: NVIDIA Tesla V100 (32 GB).
- Training time: Preliminary results show Performer converges faster than ConvTasNet.

4. Experiments

4.1. Dataset and Preprocessing

Dataset. We use the MOISESDB dataset, which provides multi-instrument splits and stereo-mixed versions of each track. The total dataset size is approximately 500 hours of music. To match the conditions of the original MixIT paper, we restrict our experiments (with $M = 8$) to songs having four or fewer stems (i.e., `tracks` ≤ 4).

Preprocessing.

- *Segmentation.* Each track is divided into fixed-length segments using a sliding-window approach. We generate segments of length $\{3\,\mathrm{s},\ 5\,\mathrm{s},\ 10\,\mathrm{s},\ 30\,\mathrm{s}\}$.
- *Train/Validation Split.* For each segment length, we split the data into 80% training and 20% validation sets.
- *STFT Parameters.*
  - FFT size: 1024
  - Hop size: 256
  - Window function: Hann
- *Spectrogram Normalization.* We apply either global min–max normalization or standard z-score normalization to each spectrogram before input to the network.

4.2. Experimental Setup

4.2.1. Model Variants

- **Baseline: MixIT + ConvTasNet.** We re-implement the original MixIT + ConvTasNet setup with segment length of 3 s and $M = 8$ masks. ConvTasNet follows an encoder–separator–decoder design using 1D convolutions.
- **Proposed: MixIT + Performer (+ Audio MAE).** We replace ConvTasNet with our Performer-based separation network, pretrained via Audio MAE, and fine-tune with MixIT. Experiments are run at segment lengths of 3 s, 5 s, 10 s, and 30 s.

4.2.2. Hyperparameters

- *Training Epochs:* Up to 100 epochs (with early stopping if validation loss does not improve for 10 consecutive epochs).
- *Learning Rate Schedule:*

$$\mathrm{LR}(t) = 10^{-4} \longrightarrow 10^{-5} \quad \text{(cosine decay schedule)}$$

- *Batch Size:* 4 (limited by GPU memory).
- *Dropout:* 0.1 applied in Performer's FFN layers.
- *Weight Decay:* 1×10^{-2} (AdamW).

4.2.3. Evaluation Metrics

We measure separation performance using the Scale-Invariant Signal-to-Noise Ratio (SI-SNR). Given a reference signal s and an estimated signal $\hat{s}$, SI-SNR is defined as:

$$\mathrm{SI\text{-}SNR}(\hat{s}, s) = 10 \log_{10} \frac{\|\mathrm{proj}_s(\hat{s})\|^2}{\|\hat{s} - \mathrm{proj}_s(\hat{s})\|^2}, \qquad \mathrm{proj}_s(\hat{s}) = \frac{\langle \hat{s},\, s\rangle}{\|s\|^2}\, s. \tag{4}$$

Higher SI-SNR indicates better separation (closer to the original waveform).

In addition, we track:

- **Training Time.**
  1. Time per epoch (in minutes).
  2. Total training time until convergence.
- **GPU Memory Usage.**
  - Peak memory consumption (monitored via `nvidia-smi`).

4.2.4. Experimental Procedure

1. **Baseline Reproduction.**
   - Train MixIT + ConvTasNet with segment length of 3 s and $M = 8$.
   - Verify if our reproduction matches reported performance in the original MixIT paper.
2. **Proposed Model Training.**
   - *Stage 1 (Pretraining).* Train the Performer encoder via Audio MAE on the entire MOISESDB corpus.
   - *Stage 2 (Fine-tuning).* For each segment length (3 s, 5 s, 10 s, 30 s), fine-tune the Performer-based separation model with MixIT loss.
3. **Evaluation.**
   - Compute SI-SNR on the held-out test set (20% of data) for each model variant.
   - Analyze SI-SNR as a function of segment length.
   - Compare training time and GPU memory usage between Baseline and Proposed methods.

4.3. Alternative Evaluation Metrics without GT

In conventional source separation tasks, the availability of ground truth sources allows for direct evaluation using objective metrics such as SI-SNR. However, in our work, we constructed a 4-stem mixture dataset from MOISESDB18 without access to source-level segmentation labels. To address this, we propose two alternative evaluation metrics that estimate the quality of source separation without relying on ground truth: (1) Mask Overlapping Measure and (2) Effective Mask Count via Entropy. We also introduce auxiliary losses to directly enhance the model's performance in terms of these metrics.

4.3.1. Mask Overlapping Measure

The Mask Overlapping Measure is designed to assess how much the estimated source masks overlap with each other. Ideally, each mask should capture a distinct audio source. If multiple masks exhibit high similarity, it implies redundancy and ineffective separation.

Formally, let $m_{b,i} \in \mathbb{R}^T$ denote the i-th mask in a batch of size B. We first normalize each mask to unit L_2 norm. Then, the cosine similarity between all distinct mask pairs $(i \neq j)$ is computed, and the average similarity over the batch is used as the overlap score. A higher overlap score indicates poor disentanglement between sources, while a lower score indicates that each mask captures unique information.

4.3.2. Effective Mask Count via Entropy

Even if the model outputs M masks, it is possible that only a few of them are actually utilized, while others contribute negligibly. To quantify how many masks are meaningfully used, we compute the entropy of the energy distribution across masks.

Let p_m denote the normalized average energy of the m-th mask across time:

$$H = -\sum_{m=1}^{M} p_m \log p_m \quad \text{(Shannon entropy)}$$

Then, the effective number of masks is estimated by exponentiating the entropy:

$$\text{effective\_masks} = \exp(H) \quad \text{(effective mask)}$$

This value lies in the range $[1, M]$. A value close to M implies that the model distributes energy evenly across all masks, while a value near 1 indicates that only a single mask dominates.

4.3.3. Motivation for Auxiliary Losses

To improve the model's separation quality without ground truth references, we introduce two auxiliary loss terms: **Diversity Loss** and **Sparsity Loss**. These losses are specifically designed to enhance the proposed evaluation metrics — Mask Overlapping and Effective Mask Count — which are discussed in Sections 4.3.1 and 4.3.2, respectively.

Diversity Loss. The Diversity Loss encourages the model to produce mutually dissimilar masks that capture distinct audio sources. This loss is directly tied to reducing mask overlap.

Let $m_{b,i} \in \mathbb{R}^T$ denote the i-th mask in the b-th sample of the batch. We first normalize each mask by its L_2 norm:

$$\widetilde{m}_{b,i} = \frac{m_{b,i}}{\|m_{b,i}\|_2} \tag{5}$$

Then, we compute the absolute cosine similarity between every pair of masks $(i \neq j)$ within a sample and take the average across the batch:

$$\mathcal{L}_{\text{div}} = \frac{1}{B}\sum_{b=1}^{B}\left(\frac{1}{N(N-1)}\sum_{i\neq j}|\langle \widetilde{m}_{b,i}, \widetilde{m}_{b,j}\rangle|\right) \tag{6}$$

A lower $\mathcal{L}_{\text{div}}$ indicates that the masks are less correlated, implying better disentanglement and reduced redundancy among sources.

Sparsity Loss. While Diversity Loss encourages decorrelation, Sparsity Loss promotes energy concentration in fewer masks. This complements the Effective Mask Count metric by penalizing widespread energy dispersion.

Let $\hat{s}_{b,i}(t)$ be the waveform of the i-th separated source and $x_b(t)$ be the original mixture. We first compute the root-mean-square (RMS) energy of each separated source and the mixture, adding a small ε for numerical stability:

$$r_{b,i} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\hat{s}_{b,i}(t)^2 + \varepsilon} \tag{7}$$

$$r_{b,\text{mix}} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}x_b(t)^2 + \varepsilon} \tag{8}$$

We then define the sparsity loss as the sum of normalized RMS energies averaged over the batch:

$$\mathcal{L}_{\text{sp}} = \frac{1}{B}\sum_{b=1}^{B}\left(\frac{1}{r_{b,\text{mix}}}\sum_{i=1}^{N} r_{b,i}\right) \tag{9}$$

A lower $\mathcal{L}_{\text{sp}}$ implies that fewer masks carry meaningful energy, encouraging the model to utilize only a subset of the available channels. This regularization is particularly helpful when the true number of sources is smaller than N.

Combined Objective. These two losses are combined with the original MixIT loss to form the final training objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MixIT}} + \lambda_{\text{div}}\mathcal{L}_{\text{div}} + \lambda_{\text{sp}}\mathcal{L}_{\text{sp}} \tag{10}$$

where λ_{div} and λ_{sp} are tunable hyperparameters controlling the influence of each auxiliary loss. In our experiments, we empirically set both values to 0.1.

This combined objective enables the model to learn disentangled and sparse representations even without source-level ground truth, improving both qualitative behavior and interpretability.

5. Results and Analysis

5.1. SI-SNR Performance Comparison

Table 1 shows the SI-SNR for ConvTasNet (Baseline) and Performer (Proposed) at different segment lengths, as extracted from the experimental results.

Seg Length (s)	ConvTasNet SNR	Performer SNR
3	-9.11	$+1.66$
5	-8.52	$+2.00$
10	-9.73	$+2.33$
30	× (CUDA OOM)	$+3.52$

Table 1. Comparison of SI-SNRs by segment length for ConvTasNet and Performer.

Analysis. The SI-SNR results demonstrate a clear performance advantage of the Performer over ConvTasNet across all segment lengths. At 3 seconds, ConvTasNet achieved -9.11 dB while the Performer attained $+1.66$ dB, indicating a substantial improvement. Similarly, for the 5-second segments, ConvTasNet recorded -8.52 dB compared to $+2.00$ dB for the Performer. At 10 seconds, the trend persisted with ConvTasNet at -9.73 dB and Performer at $+2.33$ dB. Notably, ConvTasNet was unable to process the 30-second segments due to CUDA out-of-memory issues, whereas the Performer successfully reached $+3.52$ dB. These findings highlight the Performer's superior capability to model long-term dependencies and handle extended input durations efficiently, particularly in memory-constrained environments where traditional convolutional models fail.

5.2. Training Efficiency Comparison

Table 2 reports the average training time (minutes per epoch) for ConvTasNet (Baseline) and Performer (Proposed) across segment lengths.

	ConvTasNet	Performer
Seg Length (s)	Time (min/epoch)	Time (min/epoch)
3	17.05	12.57
5	20.33	16.04
10	43.25	35.42
30	× (CUDA OOM)	46.33

Table 2. Comparison of training times by segment length for ConvTasNet and Performer.

Analysis. In terms of training efficiency, the Performer consistently demonstrated faster training times compared to ConvTasNet across all segment lengths. For 3-second segments, ConvTasNet required 17.05 minutes per epoch, whereas the Performer completed an epoch in only 12.57 minutes. At 5 seconds, the Performer again showed an advantage with 16.04 minutes per epoch compared to 20.33 minutes for ConvTasNet. For 10-second segments, the training time gap widened, with ConvTasNet taking 43.25 minutes versus 35.42 minutes for the Performer. Notably, ConvTasNet failed to train on 30-second segments due to CUDA out-of-memory errors, while the Performer successfully completed training with an average of 46.33 minutes per epoch. These results confirm that the Performer's linearized attention mechanism enables more efficient training, particularly for longer audio sequences where convolutional models encounter scalability and memory limitations.

Seg Length (s)	w/ AudioMAE	w/o AudioMAE
3	$+1.66$	$+2.60$
5	$+2.00$	$+3.16$
10	$+2.33$	$+3.52$
30	$+3.52$	× (CUDA OOM)

Table 3. Ablation study on Audio MAE pretraining: Comparison of SI-SNR performance with and without Audio MAE initialization.

5.3. Ablation Study on Audio MAE

We additionally conducted an ablation study to investigate the impact of the Audio MAE pretraining on Performer performance. To isolate its influence, we trained a Performer variant without Audio MAE initialization (random initialization). The resulting SI-SNR values across segment lengths were as follows Table 3.

These results indicate that removing Audio MAE pretraining yields marginally higher SI-SNR scores at shorter segments compared to the original Performer results reported in Table 3. This suggests that using Audio MAE, which is transformer-based, may introduce excessive feature transformations when combined with the Performer, potentially degrading final separation quality. However, the absence of feature compression through MAE negatively affects memory efficiency, as demonstrated by the failure to handle 30-second segments due to memory limitations. Therefore, Audio MAE pretraining appears beneficial primarily for enhancing memory efficiency rather than directly improving short-duration separation performance. Further experiments addressing potential redundancy or information loss in feature transformations are planned, including investigations into ”cheating” scenarios to validate these observations rigorously.

5.4. Ablation Study on Auxiliary Losses

To better understand the impact of the auxiliary losses (Diversity and Sparsity), we conducted an ablation study comparing two experimental conditions: one utilizing only the original MixIT loss and the other incorporating the additional Diversity and Sparsity losses.

5.4.1. Evaluation Metrics

As discussed previously, conventional metrics like SI-SNR alone might not fully capture model behavior in scenarios lacking ground truth segmentation. Thus, we evaluated models using three metrics:

- **SI-SNR:** Indicates the waveform reconstruction quality.
- **Mask Overlapping:** Measures the redundancy between separated masks.
- **Effective Mask Count ($\mathrm{N_E}$):** Indicates how evenly energy is distributed across the available masks, reflecting the effectiveness of source separation.

5.4.2. Results

Tables 4 and 5 summarize the ablation results across different segment lengths. Table 4 reports the baseline performance using only the MixIT loss, while Table 5 shows the results when both Diversity and Sparsity losses are incorporated alongside MixIT. Each table includes SI-SNR, Mask Overlapping, and the number of effective masks ($\mathrm{N_E}$) as evaluation metrics.

Segment Length	SI-SNR	Overlapping	$\mathrm{N_E}$
3 sec	2.60	0.59	6.36
5 sec	3.16	0.61	5.93
10 sec	3.87	0.52	6.29

Table 4. Results using only MixIT loss.

Segment Length	SI-SNR	Overlapping	$\mathrm{N_E}$
3 sec	2.56	0.49	5.60
5 sec	3.32	0.35	4.48
10 sec	2.26	0.44	5.45

Table 5. Results using MixIT + Diversity + Sparsity losses.

Analysis. The experimental results demonstrate that adding auxiliary losses consistently improves both Mask Overlapping and Effective Mask Count metrics. Specifically, the Diversity and Sparsity losses effectively reduced mask overlap, indicating that the model produces more distinct and specialized masks for different audio sources. Similarly, the number of effective masks ($\mathrm{N_E}$) decreased, suggesting improved energy concentration and reduced redundancy across masks. However, SI-SNR scores did not improve significantly after introducing auxiliary losses. This lack of improvement likely results from data limitations, as previously discussed in Section 5.5. Given that the primary goal of these auxiliary losses was not direct waveform reconstruction improvement but rather enhanced mask utilization and reduced redundancy, the achieved improvements in mask-related metrics align with our objectives. In summary, auxiliary losses effectively guided the model toward more structured and meaningful separation behavior, highlighting the importance of metrics beyond SI-SNR in evaluating self-supervised audio separation models.

5.5. Limitations and Overfitting Issues

Negative SI-SNR Causes.

- **Insufficient Training Data:** Only ≈ 10 hours of 4-stem music from MOISESDB were available, limiting generalization.
- **Overfitting Signs:** $\mathrm{SI\text{-}SNR_{train}} \approx +3\text{–}+4$ dB, whereas $\mathrm{SI\text{-}SNR_{val/test}} \ll 0$ dB.
- **Regularization Limits:** BatchNorm and Dropout (0.1) did not fully prevent overfitting.

Additional Experiments. Although additional regularization techniques were explored, data augmentation methods such as time-stretching and pitch-shifting did not lead to noticeable improvements in SI-SNR performance.

Performer's Global Context Learning.

- Performer's SI-SNR gains increase with segment length, demonstrating its ability to capture long-term dependencies (rhythm, melody, harmony).
- Global attention enables improved separation quality in complex musical passages.

ConvTasNet's Locality Limitation. ConvTasNet shows minimal performance variation across segment lengths and cannot process very long segments, underscoring its reliance on local pattern learning.

Impact of Audio MAE Pretraining.

- Performer initialized with Audio MAE weights shows faster early convergence compared to random initialization.
- Final SI-SNR remains limited by data size, indicating more diverse pretraining data is needed.

These observations collectively reinforce the need for not only stronger regularization but also richer and more diverse training datasets. Our results emphasize the necessity of large-scale pretraining and rich mixture data for robust self-supervised music separation.

6. Discussion

Building on the results and limitations discussed in Section 5, this section highlights the strengths of the proposed model, addresses its key shortcomings, and outlines possible future directions for improvement and extension.

6.1. Advantages of the Proposed Model

- **Global Context Learning.** The Performer's FAVOR+ attention mechanism efficiently captures long-term dependencies in extended audio sequences. This allows the model to leverage global musical structure (e.g., rhythm, melody, harmony) that is difficult for convolution-based networks to learn.
- **Computational Efficiency.** Thanks to the linearized attention complexity ($\mathcal{O}(L \cdot r)$), the Performer requires less training time and GPU memory than ConvTasNet, especially on longer segments. This improved efficiency makes the proposed method more practical and scalable for real-world deployment.
- **Pretraining-based Generalization.** By pretraining the encoder with Audio MAE on unlabeled audio, the model learns robust global features before fine-tuning. This initialization stabilizes early training and accelerates convergence compared to random initialization.

6.2. Limitations and Potential Improvements

Dataset Size.

- We trained on only ≈ 10 hours of eligible 4-stem music. For complex domains like music separation, this is insufficient to cover the diversity of instruments, styles, and recording conditions.
- **Improvement:** Acquire a larger pretraining corpus (e.g., 100–1000 hours of multi-instrument mixes) to mitigate overfitting and improve generalization.

Excessive Dimensionality Reduction.

- FAVOR+ attention compresses high-dimensional audio embeddings into a lower-dimension representation. During the masking and reconstruction process, some fine-grained details may be lost, and the decoder's upsampling may not fully recover them.
- **Improvement:**
  - Enhance upsampling modules (e.g., use transposed convolutions or adjacent-sample interpolation) to better preserve detailed structure.
  - Introduce multi-scale feature fusion by combining low-level (high-resolution) and high-level (contextual) representations during decoding.

MixIT Framework Constraints.

- In our experiments, MixIT is configured with $N = 8$ output streams. However, real music often contains more than eight simultaneous sources (e.g., multiple instruments, percussion, backing vocals).
- **Improvement:** Develop a more flexible mechanism for estimating the number of active sources per mixture, or allow N to adapt dynamically during training.

6.3. Proposed Additional Experiments

To further validate and extend the proposed model, we suggest several promising directions for future experimentation and evaluation.

Large-Scale Pretraining.

- Pretrain Audio MAE on a vastly larger and more diverse music corpus (varied genres, instruments, and recording setups).
- This should further improve the encoder's ability to extract transferable global features, reducing overfitting on smaller downstream datasets.

Expanded Evaluation Metrics.

- In addition to SI-SNR, include other separation metrics such as Signal-to-Distortion Ratio (SDR), Perceptual Evaluation of Speech Quality (PESQ), and Scale-Invariant SDR (SI-SDR).
- A multi-metric evaluation will provide a more comprehensive assessment of perceptual quality and distortion.

Downstream Task Applications.

- Use the separated vocals and instrument tracks for music transcription, music recommendation, remixing, or accompaniment generation.
- Evaluate how improved separation quality affects the performance of these downstream tasks.

Multichannel Extension.

- Extend from mono to stereo and surround (e.g., 5.1) audio.
- Investigate how spatial cues can be incorporated into the Performer model to further enhance separation quality in multi-channel settings.

Real-Time Separation Feasibility.

- Evaluate the model's latency and throughput on streaming audio inputs.
- If latency is sufficiently low, such a system could be used for live performance, broadcast, or karaoke applications.

7. Conclusion

In this work, we proposed replacing the ConvTasNet-based separation module in the MixIT self-supervised framework with a Performer-based model pretrained using Audio MAE. Leveraging FAVOR+ attention, the Performer effectively captures long-range dependencies in audio, resulting in improved music separation performance and better computational efficiency.

Key Findings

- **Performance Improvement.** Performer achieves significantly higher SI-SNR on longer segments (e.g., $+3.52\ \mathrm{dB}$ on 30 s), whereas ConvTasNet could not complete training due to memory limitations.
- **Computational Efficiency.** Training time is reduced by 20–30% and GPU memory usage is lowered by up to 4 GB compared to ConvTasNet.
- **Global Context Benefits.** Even with limited data, Performer's global context modeling yields consistent improvements over CNN-based approaches.

Limitations and Future Work

- **Dataset Size and Overfitting.** Training on only ≈ 10 hours of 4-stem music leads to overfitting and suboptimal SI-SNR. Future work will explore scaling to 100–1000 hours of diverse music.
- **Information Loss from Dimensionality Reduction.** The linearized attention in FAVOR+ reduces embedding dimensionality, which may limit fine detail retention. We plan to improve upsampling and incorporate multi-scale feature fusion.
- **Multi-channel and Real-time Extensions.** We will extend our approach to stereo/multichannel inputs and assess real-time feasibility for applications such as live performance or broadcasting.

References

[1] S. Wisdom, et al. *Unsupervised Sound Separation Using Mixture Invariant Training*. arXiv preprint arXiv:2006.12701, 2020.

[2] K. Choromanski, et al. *Rethinking Attention with Performers*. arXiv preprint arXiv:2009.14794, 2020.

[3] A. Vaswani, et al. *Attention Is All You Need*. arXiv preprint arXiv:1706.03762, 2017.

[4] Y. Luo, N. Mesgarani. *Conv-TasNet: Surpassing Ideal Time–Frequency Masking for Speech Separation*. arXiv preprint arXiv:1809.07454, 2018.

[5] A. Katharopoulos, et al. *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention*. arXiv preprint arXiv:2006.16236, 2020.

[6] Z. Song, et al. *Learning Audio Representations with Masked Autoencoders*. arXiv preprint arXiv:2307.15913, 2023.