Singapore Computer Society
International Journal of Information Technology
Vol 31, Issue 1, 2025

Robust Music Source Separation as an LLM-Facing Perception Interface Using Gaussian Noise Injection

Zixi Yan[ab], Lu Gan[b] and Hongqing Liu[c]

[a]Nanyang Technological University 50 Nanyang Avenue, Singapore 639798
[b]Brunel University of London, Uxbridge UK, UB8 3PH
[c]Chongqing University of Posts and Telecommunications, No.2, Chongwen Road, Chongqing, China, 400065

zixi003@e.ntu.edu.sg, lu.gan@brunel.ac.uk, hongqingliu@cqupt.edu.cn

Abstract

Background: Music-aware multimodal large language models require reliable perceptual evidence to answer questions about lyrics, vocals, instruments, accompaniment, rhythm, genre, or performance characteristics. For music inputs, this perception interface is challenging because songs are polyphonic and source-overlapped, while real-world recordings often contain audience noise, equipment interference, and room effects. Without robust stem extraction, noisy and source-overlapped mixtures may provide unstable evidence to downstream audio encoders and language reasoning modules. This study investigates robust music source separation as an LLM-facing perception interface that converts noisy mixtures into stable stem-level musical evidence.

Methods: Using Hybrid Transformer Demucs as a Transformer-based music source separation front-end, we fine-tuned the model on MUSDB18-HQ with Gaussian noise injection. Two augmentation strategies were compared: injecting noise only into mixture tracks and injecting noise into both mixture and source tracks. Noise variance and injection probability were varied to evaluate their effects on separated vocals, drums, bass, and other stems.

Results: The best configuration, applying Gaussian noise only to mixture tracks with variance 0.001 and probability 0.8, improved source-to-distortion ratio by up to 0.12 dB under high-intensity noisy test conditions compared with the baseline. Although the gain is modest, it is obtained without changing the model architecture or adding inference cost, and the degradation with mix-and-source injection suggests that preserving clean target stems is important for robust evidence extraction.

Conclusions: Lightweight noise-stable fine-tuning can improve the robustness of Transformer-based music source separation interfaces. The findings support robust stem extraction as an adaptation strategy for music-aware multimodal large language model pipelines, while downstream language-model evaluation remains future work.

Keywords: Multimodal Large Language Models; Music Source Separation; LLM-Facing Perception Interface; HT Demucs; Gaussian Noise Injection; MUSDB18-HQ.

I. Introduction

Music-aware multimodal large language models are increasingly expected to reason about songs, lyrics, vocals, instruments, accompaniment, rhythm, genre, and performance characteristics. In such systems, language-level reasoning depends on reliable perceptual evidence extracted from the audio input before it is passed to downstream audio encoders or language reasoning modules. Unlike speech-only inputs, music recordings are naturally polyphonic and source-overlapped: vocals, drums, bass, and other accompaniment components are mixed into a single waveform. This makes music source separation a central front-end task for music-aware multimodal systems rather than merely a tool for remixing or studio production.

For music understanding, separated stems can provide structured evidence for different types of downstream reasoning. Vocal stems can support lyrics recognition, singing voice understanding, and questions about performed content; drums and bass can support rhythm, groove, and style analysis; and the remaining accompaniment can provide harmonic, timbral, and contextual evidence. Figure 1 illustrates the basic music source separation process. In this study, the separated stems are treated as stem-level musical evidence that may support later multimodal language reasoning, while the present evaluation remains focused on the robustness and quality of the separation front-end itself.

A practical challenge is that real-world music recordings are rarely clean. Live performances may include audience noise, room reverberation, microphone artefacts, and equipment interference, while studio recordings may still contain low-level capture noise. When a source separation

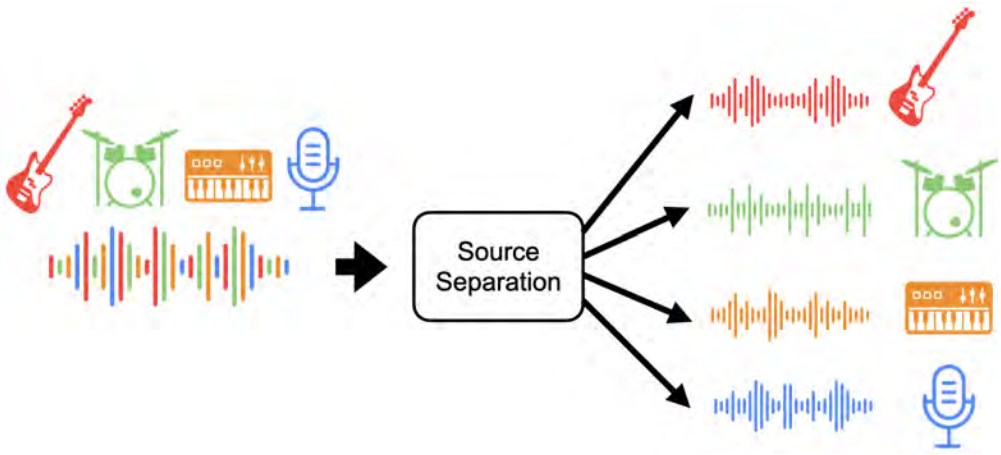


Fig 1: Music source separation process [40].

front-end is sensitive to such perturbations, the extracted stems can contain residual interference, distorted timbre, or blurred source boundaries. These errors reduce not only the objective quality of separated tracks, but also the reliability of the perceptual evidence available to music-aware multimodal large language model pipelines.

This paper studies robust music source separation as an LLM-facing perception interface. Specifically, Hybrid Transformer Demucs is used as a representative Transformer-based music source separation model, and Gaussian noise injection is applied during fine-tuning to improve stability under noisy input conditions. The method does not change the underlying separation architecture and does not introduce additional inference cost. Two training-time augmentation strategies are compared: injecting Gaussian noise only into mixture tracks, and injecting Gaussian noise into both mixture and source tracks.

The main contributions of this paper are as follows:

1. It reframes robust music source separation as an LLM-facing perception interface for music-aware multimodal systems, where separated stems serve as structured musical evidence for downstream language-level reasoning.
2. It investigates Gaussian noise injection as a lightweight robustness adaptation strategy for Hybrid Transformer Demucs without modifying the model architecture or increasing inference cost.

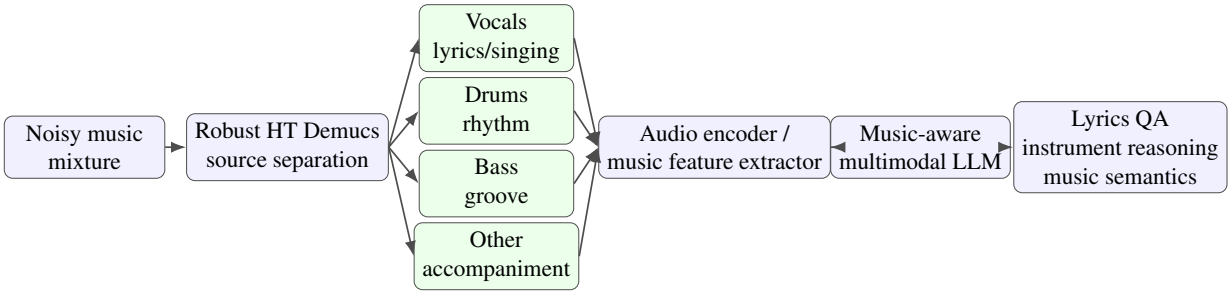


Fig 2: LLM-facing music perception interface. Robust music source separation converts a noisy music mixture into stem-level musical evidence before downstream audio encoding and language-level reasoning.

3. It compares mixture-only and mixture-and-source noise injection strategies on MUSDB18-HQ and analyzes their effects on vocals, drums, bass, and other stems using source-to-distortion ratio as a proxy for stem-level evidence quality.

The results show that the best mixture-only configuration yields a small but consistent source-to-distortion ratio improvement under high-intensity noisy test conditions, while injecting noise into both mixtures and source targets reduces performance. These findings suggest that preserving clean target stems during noisy-input adaptation is important when source separation is used as an evidence extraction interface. The study does not claim to train or improve a language model directly; instead, it focuses on improving the input-side robustness of a music perception component that can support future music-aware multimodal large language model pipelines.

II. Related Work

A. Music Understanding in Multimodal Large Language Models

Recent progress in audio-language modeling and multimodal large language models has expanded language-model reasoning from text and images to speech, music, and general audio. AudioLM demonstrates that language modeling principles can be applied to audio generation [7], while MusicLM extends text-conditioned generation to music [1]. Contrastive audio-language models such as CLAP connect audio events with natural-language descriptions [19], and systems such

as SALMONN further explore generic hearing abilities for large language models [64]. These developments indicate that language reasoning over audio increasingly depends on reliable audio representations.

For music-oriented use cases, a multimodal model may be expected to answer questions about lyrics, vocals, instrumentation, rhythm, genre, performance style, and accompaniment. These capabilities depend on whether the audio front-end can provide stable perceptual evidence before language-level reasoning is performed. In noisy and polyphonic music recordings, the raw mixture alone may obscure the acoustic cues needed by downstream audio encoders or language reasoning modules.

This paper therefore treats music source separation as an input adaptation problem for music-aware multimodal systems. Instead of using source separation only as a production tool, the separated stems are interpreted as structured musical evidence: vocals can support lyrics and singing voice understanding, drums and bass can support rhythm and groove analysis, and other accompaniment components can support timbral and harmonic interpretation. Prior work has also shown that source separation can improve music representation and classification in downstream machine-listening settings [44]. This motivates studying whether the separation interface itself remains stable when the input mixture is noisy.

B. Robust Audio and Music Front-Ends

Robustness has long been a central problem in audio perception systems. In speech-oriented systems, noise-robust recognition and enhancement have been studied through multimodal learning, speech enhancement, and audio-visual separation [28, 45]. These studies show that the reliability of downstream recognition depends strongly on the quality of the front-end representation under

noisy conditions. Similar considerations apply to music-aware multimodal large language model pipelines, where corrupted stem extraction can propagate unreliable evidence into later reasoning modules.

For music recordings, robustness is further complicated by polyphony and source overlap. The desired output is not only a cleaner waveform, but a set of musically meaningful stems. Live recordings may include audience sounds and room effects, while studio recordings may contain equipment noise or leakage between sources. These conditions motivate training-time adaptation strategies that improve front-end stability without changing the downstream system. Gaussian noise injection provides a simple and controllable way to study this problem because it perturbs the training data while leaving the inference-time architecture unchanged.

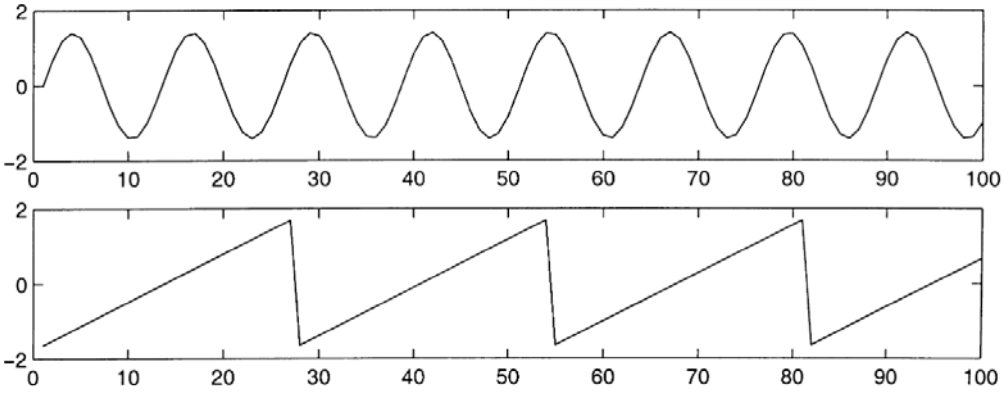
C. Blind Source Separation and Traditional Methods

Music source separation is closely related to the broader problem of Blind Source Separation, where the goal is to recover underlying source signals from observed mixtures without complete knowledge of the mixing process [11]. A simple linear mixture can be written as:

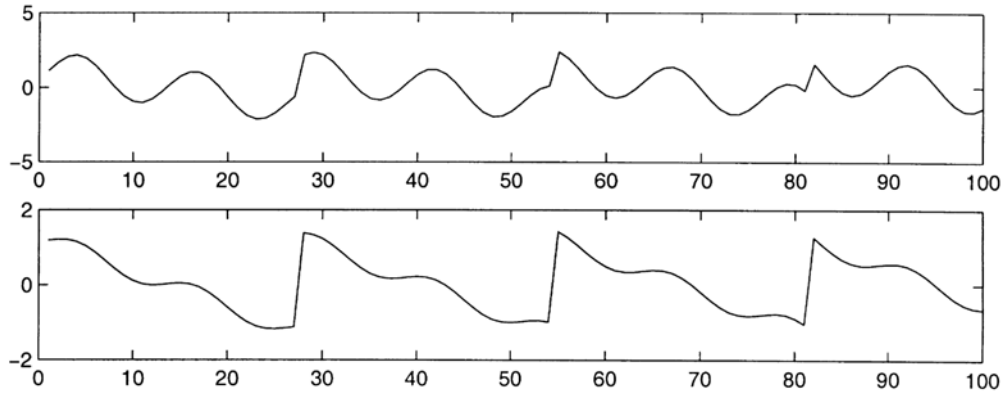
$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{1}$$

where $\mathbf{x}$ denotes the observed mixture, $\mathbf{s}$ denotes the latent source signals, and $\mathbf{A}$ denotes the mixing process. Figure 3 illustrates the relationship between source signals and their mixture.

Traditional approaches include Independent Component Analysis and Non-negative Matrix Factorization. Independent Component Analysis relies on assumptions such as statistical independence and non-Gaussianity of sources [29]. Algorithms such as FastICA, InfoMax, and JADE



(a) Signals from two sources.



(b) Mixed signals.

Fig 3: The signals from two sources and the mixed signals [29].

estimate an unmixing transform by optimizing non-Gaussianity measures [30, 3, 8]. However, music signals often violate these assumptions because instruments can be harmonically correlated, rhythmically synchronized, and mixed in complex stereo or single-channel settings.

Non-negative Matrix Factorization decomposes a non-negative time-frequency representation into basis patterns and activation weights [49, 36]. In music source separation, the basis matrix can be interpreted as spectral templates for instruments, while activation weights describe their temporal presence. Although these methods provide useful conceptual foundations, their reliance on simplified assumptions limits their effectiveness for modern polyphonic music. This limitation motivates deep learning approaches that learn separation functions directly from data.

D. Deep Learning Models

While traditional methods have demonstrated numerous practical applications, deep learning approaches have led to clear improvements in music source separation performance [59]. These approaches can be broadly categorized into two main streams: **spectral domain methods** and **time domain methods**, each with distinct advantages and limitations.

D-1. Spectral Domain Methods

Spectral domain methods are based on the fact that different components of music have distinct and characteristic spectral distributions, as shown in Figure 4. Those methods typically use time-frequency representations obtained from *Short-Time Fourier Transform* (Short-Time Fourier Transform). These methods generally adopt an encoder-decoder architecture where the network generates a mask for each source's magnitude spectrogram on a frame-by-frame basis.

Jansson et al. introduced the U-Net architecture from medical image processing into music source separation. The network treating the task as an image-to-image translation problem by converting mixed spectrograms into their component sources [31]. This approach was later enhanced by Takahashi et al. who proposed MMDenseLSTM. This model combined DenseNet with Long Short-Term Memory networks to better capture temporal dependencies [62].

Recent works prefer to use complex spectrograms both for inputs and outputs [12]. This provides a more comprehensive implementation and removes the limitations that come with ideal ratio masks. The Band-Split RNN[39], a strong spectrogram-domain model, combines multiple dual-path RNNs operating in carefully designed frequency bands, and achieved 8.9 dB Signal-to-Distortion Ratio on the MUSDB dataset.

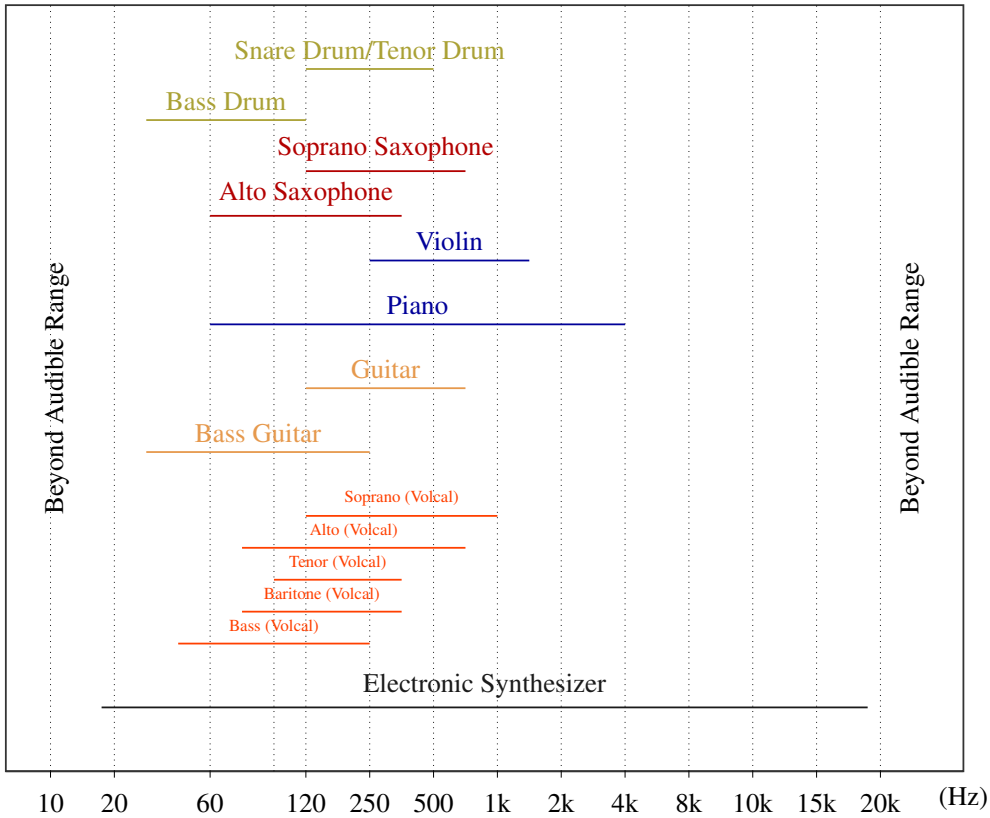


Fig 4: Frequency distribution of different musical instruments.

Although the spectral domain approach is very effective, it faces some limitations in capturing phase information, which is crucial for high-quality audio reconstruction.

D-2. Time Domain Methods

Time domain approaches work directly with waveforms, therefore it is not limited by spectral methods. Wave-U-Net [58] became a pioneer of this approach by directly mapping mixed waveforms to separated sources. Conv-TasNet [38], which was initially designed for speech separation, improved performance by replacing Short-Time Fourier Transform with learned basis functions through one-dimensional convolutions.

Demucs [18] further advanced time-domain separation by using bidirectional LSTMs between encoder and decoder components. The model achieved a Signal-to-Distortion Ratio of 5.73 dB on

the MUSDB18 dataset. More recent enhancements like Hybrid Demucs [14] have improved the architecture by incorporating cross-domain processing techniques.

D-3. Recent Advancements

Recent research focuses on addressing the limitations of both approaches. Hybrid models that use both time-domain and spectral-domain information have become a promising direction [51]. Hybrid Transformer Demucs replaces the innermost layers of Hybrid Demucs with a cross-domain Transformer encoder [55]. With additional training data, it reports 9.20 dB SDR, making it a strong baseline for studying robustness in Transformer-based music source separation.

Attentional mechanisms are also increasingly integrated into Music Source Separation systems to enhance feature representation. Dual-dimension sequence attention [4, 67] considers not only channel-wise feature differences but also temporal structures in music, recognizing that instruments follow certain patterns across different sections of a composition (intro, verse, chorus, etc.) [37].

In addition, by taking advantage of the complementary advantages of different technologies, the method of combining multiple models has also demonstrated excellent performance [43]. Through effective model blending, KUIELAB-MDX-Net can be an example of this trend. Additionally, in order to better capture both temporal and frequency characteristics of musical instruments, improved loss functions including multi-resolution spectral regularization have been proposed [32]. Table 1 shows a comparison of various models in the field of music source separation.

It can be seen from the table that most of the models operates on only single domain. The "more data" column indicates the amount of extra training data used by the model. This can affect model performance, as discussed in Section C.. Hybrid Transformer Demucs is chosen

Table 1: Comparison of music source separation models

Model	Domain	Extra data?	Overall SDR
Open-Unmix [60]	spectral	no	5.3
Demucs (v2)	waveform	no	6.3
ResUNetDecouple+ [34]	spectral	no	6.7
KUIELAB-MDX-Net [32]	**hybrid**	no	7.5
Band-Spit RNN [39]	spectral	no	**8.2**
Band-Spit RNN [39]	spectral	1.7k (mixes only)	**9.0**
Hybrid Demucs (v3)	**hybrid**	no	7.7
MMDenseLSTM [62]	spectral	804 songs	6.0
D3Net [63]	spectral	1.5k songs	6.7
Spleeter [26]	spectral	25k songs	5.9
Hybrid Transformer Demucs (v4)	**hybrid**	800 songs	**9.0**

because it combines time-domain and frequency-domain processing with a strong reported Signal-to-Distortion Ratio.

E. Music Source Separation Campaigns and the MUSDB18 Dataset

Similar to other fields, Music Source Separation has its own benchmark datasets and evaluation competitions. The Signal Separation Evaluation Campaign (SiSEC) 2015 [48] established early foundations for systematic Music Source Separation evaluation. This event introduced two datasets and utilised *BSS Eval* metrics to measure separation quality across speech mixtures, noise backgrounds, and professionally produced music recordings. The competition demonstrated that deep learning approaches were beginning to outperform traditional methods.

SiSEC 2018 [59] marked an important step with the release of **MUSDB18** [52]. The dataset contains nearly 10 hours of music with isolated stems across 150 tracks, and has become one of the most widely used datasets in the field. It has also been criticized for limited diversity and some unclean data [20].

The Music Demixing Challenge 2021 (MDX'21) [46] was the most influential competition

in the field. A key design choice was the introduction of MDXDB21, a hidden test set created exclusively for the challenge to ensure evaluation transparency. Competitors were also allowed to use extra data for training. The results showed that additional data can affect model performance.

Considering the limitations of the MUSDB18 dataset, the Sound Demixing Challenge 2023 (SDX'23) [20] mainly focus on **robust** music source separation, addressing real-world challenges of training with imperfect data. The competition simulated two types of data corruption: *label noise* (incorrect instrument tags) and *bleeding* (signal leakage between sources). This encouraged participants to adopt more creative designs rather than relying solely on the dataset.

This section reviews the literature needed to position robust music source separation as an LLM-facing perception interface. It first discusses why music-aware multimodal systems require stable stem-level evidence, then reviews robustness issues in audio and music front-ends. The section also summarizes the signal separation foundations of the task, including Blind Source Separation, Independent Component Analysis, and Non-negative Matrix Factorization, before reviewing deep learning approaches to music source separation. Finally, it discusses major evaluation campaigns and introduces the MUSDB18 dataset, which serves as the standard benchmark for music source separation research.

III. Methodology

Although SDX'23 has drawn attention to robustness in music demixing, many music source separation models are still not explicitly adapted for real-world perturbations and noisy music inputs. Défossez et al., the creators of Demucs, previously discussed pseudo-quantization noise for model compression and noted that Gaussian noise can improve robustness against certain disturbances [17]. However, this observation was primarily made in the context of model compression

rather than LLM-facing music source separation. This section introduces the problem formulation, the Hybrid Transformer Demucs architecture, and the theoretical basis for using Gaussian noise injection as a training-time robustness adaptation strategy.

A. Problem Formulation

Let x denote a mixed music signal and let s_i denote one of its target stems, where i corresponds to vocals, drums, bass, or other accompaniment. A music source separation model f_θ maps the mixture to estimated stems $\hat{s}_i = f_\theta(x)_i$. In an LLM-facing music perception pipeline, these estimated stems are not treated only as audio production outputs, but also as stem-level musical evidence that may be passed to downstream audio encoders and language reasoning modules.

This study focuses on noisy-input robustness. During testing, the mixture may be corrupted by an additive perturbation ϵ, producing $\tilde{x} = x + \epsilon$. The desired behavior is that $f_\theta(\tilde{x})$ remains close to the clean target stems so that vocals, drums, bass, and accompaniment evidence remain reliable. Gaussian noise injection is therefore used during fine-tuning as a controlled way to expose the model to noisy mixtures. The evaluation uses source-to-distortion ratio as a proxy for the quality of stem-level musical evidence, rather than as a direct measure of downstream large language model performance.

B. Model Architecture

This study uses the Hybrid Transformer Demucs model [55], developed by Meta AI Research, as the baseline music source separation interface.

Figure 5 shows the architecture of the Hybrid Transformer Demucs model. The overall model is composed of the following components:

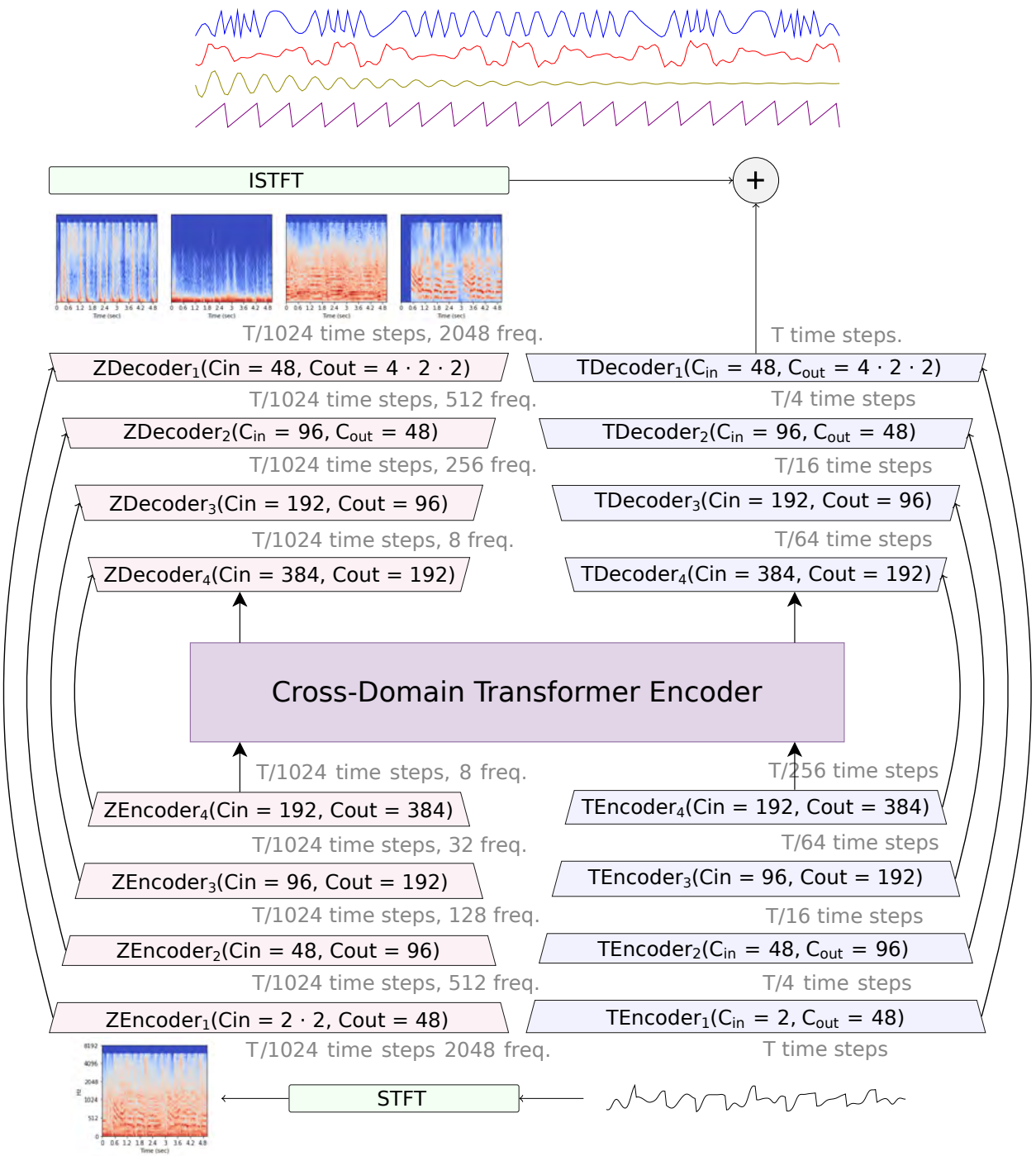


Fig 5: The architecture of the Hybrid Transformer Demucs model used as the LLM-facing music source separation interface [55].

- Two sets of encoders and decoders for the time domain and frequency domain, respectively.

- Each domain uses U-Net skip connections within its architecture.

- Cross-domain Transformer modules in the innermost layer.

Since its initial release, this model has evolved through three major iterations: Demucs (v2), Hybrid Demucs (v3), and Hybrid Transformer Demucs (v4). This section will present the model design following the chronological development of these three versions.

B-1. Mathematical Model

The aim of Demucs is to separate the mix signal into bass, drums, vocals, and other sources. If a waveform $\boldsymbol{x}_s \in \mathbb{R}^{C,T}$ is used to represent the signal from each source, where C represents the number of channels and T represents the number of time steps sampled, the whole music signal can be represented as:

$$\boldsymbol{x} = \boldsymbol{x}_{\mathrm{b}} + \boldsymbol{x}_{\mathrm{d}} + \boldsymbol{x}_{\mathrm{v}} + \boldsymbol{x}_{\mathrm{o}} \tag{2}$$

Here, $\boldsymbol{x}_{\mathrm{b}}$, $\boldsymbol{x}_{\mathrm{d}}$, $\boldsymbol{x}_{\mathrm{v}}$, and $\boldsymbol{x}_{\mathrm{o}}$ represent the bass, drums, vocals, and other sources, respectively. Then, a model g is trained to obtain each source using the mixed signal and parameters θ:

$$\boldsymbol{x}_s = g(\boldsymbol{x}, \theta) \tag{3}$$

On a Dataset $\mathcal{D}$, the parameters of the model g need to minimize the loss. In this way, the mathematical model of Demucs is obtained as follows:

$$\min_{\theta} \sum_{\boldsymbol{x} \in \mathcal{D}} \sum_{s=1}^{S} L(g(\boldsymbol{x}; \theta), \boldsymbol{x}_s) \tag{4}$$

B-2. Time Domain Architecture

Convolutional Encoders and Decoders Convoluting is an effective way to extract features from local regions of a signal. In the Hybrid Transformer Demucs architecture, 4 convolutional layers are used for both the encoder and decoder. Figure 6 illustrates the convolution process. The

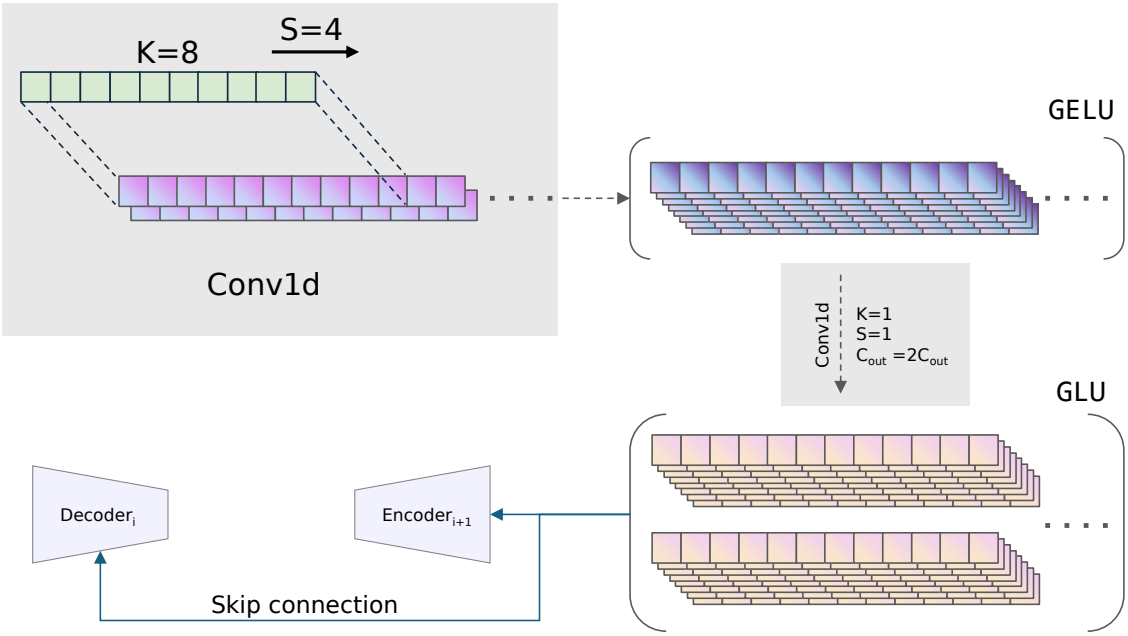


Fig 6: The convolution process.

input signal is first transformed into a waveform representation and sampled in time. Assume this generates a matrix of shape $[C, T]$, where C is the number of input channels and T is the number of time steps. The input and output channels of each block are denoted by C_{in} and C_{out}, respectively. In most contexts, the number of channels is 2 for stereo audio. In this way, $C_0 = C = 2$. Then, 48 one-dimensional convolution kernels of size $K = 8$ and stride $S = 4$ are applied to the $[2, T]$ matrix, resulting in an output matrix of shape $[48, \frac{T}{4}]$. Here, the output channels are increased to 48 ($C_1 = 48$), representing more complex features. After Layer Normalization, which makes the

feature distribution of each sample more stable, the output is activated by the GELU function [25]:

$$f(x) = x \cdot \Phi(x) \tag{5}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. The original Demucs model used the ReLU activation function, but with a smoother transition, GELU makes the model more capable of learning. Then architecture uses a GLU function [13] to adjust the "pass-through rate" for different input features. Before the GLU function, the output is divided into two parts: x_1 and x_2, where x_1 is used as the content stream and x_2 is used as the control stream. The control stream is passed through a sigmoid function to generate a gating mechanism, and then performed element-wise multiplication with the content stream:

$$\mathrm{GLU}(x) = x_1 \bigotimes \sigma(x_2) \tag{6}$$

It can be seen that this requires double the number of channels. Therefore, the output of the GELU block is convoluted again using a kernel of size $K = 1$ and stride $S = 1$ to double the channels.

This process is then repeated for the next three blocks, with the number of channels doubled each time ($C_i = 2C_{i-1}$). In the last block, C_{out} is equal to 384.

The decoder adapts the symmetric structure of the encoder. In essence, the encoders learn the features of the input signal, while the decoders transform the features back into the waveform representation.

U-Net Skip Connections The model uses skip connections similar to those in Wave-U-Net [31]. In the context of Music, phase information is more difficult to reconstruct. The skip connections

make the phase representation directly available to the decoder. Additionally, in the symmetric structure of corresponding encoder and decoder layers, features are maintained. This allows the decoder to understand both superficial and abstract features. Different from the original U-Net, the Demucs model employs transposed convolutions instead of linear interpolation combined with standard convolutions. This reduces the computation and memory requirements to 1/4 of the original.

Loss Function For the loss function in Equation 4, two functions are tested: L1 and L2:

$$\begin{aligned} L_1(g(\boldsymbol{x};\theta),\boldsymbol{x}_s) &= \frac{1}{T}\sum_{t=1}^{T}|g(\boldsymbol{x};\theta)-\boldsymbol{x}_s| \\ L_2(g(\boldsymbol{x};\theta),\boldsymbol{x}_s) &= \frac{1}{T}\sum_{t=1}^{T}(g(\boldsymbol{x};\theta)-\boldsymbol{x}_s)^2 \end{aligned} \tag{7}$$

Ablation experiments from the original research show that the performance difference between the two loss functions is not significant, and chose L1 as the loss function [18]. But this study uses L2 (Mean Squared Error) as the loss function. The reason will be discussed in Section C..

Shifting Strategy For a signal in music, an important characteristic is that a shift in time does not affect the judgment of its features. For example, a drum sound can be recognized as a drum, no matter it occurs sooner or later. Therefore, a change of the inputs signals in the mix will result in the same change in the outputs. This is also called time equivalence [38]. Based on this property, a shifting strategy is used. The model shifts the mix signal multiple times, and then takes the average of each separation result as the final output. This results in some extra computing expenses but brings 0.3 dB improvement in the Signal-to-Distortion Ratio score [18].

B-3. Frequency Domain Architecture

The time domain design can capture transient information in audio signals, such as drum strikes. However, for sustained characteristics such as pitch and harmony, the frequency domain design can better capture these features. The proposed model combines the two domains and achieves strong performance [46].

As shown in Figure 7, time and frequency branches shares the same inner layer. Therefore, the spectral branch needs to align with the temporal branch. The parameters in the frequency domain are designed to exactly match those in the time domain.

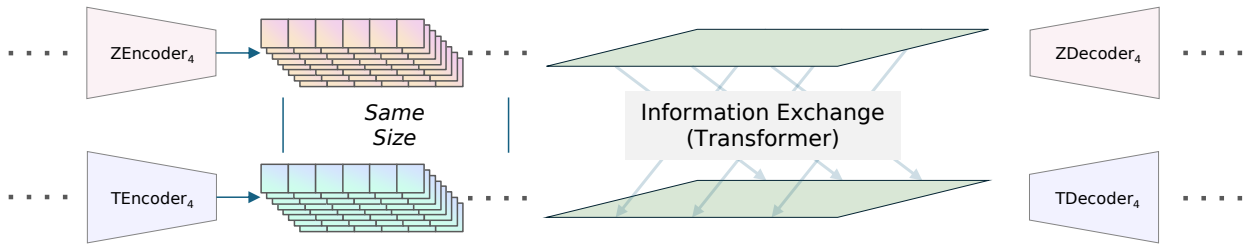


Fig 7: The collaboration of time and frequency domains.

Spectral Convolutions

Non-negative Matrix Factorization To convert the input audio into a time-frequency representation, the *Short-Time Fourier Transform* (Short-Time Fourier Transform) is applied before the convolutional layers. 4096 points are used for the Short-Time Fourier Transform in the frequency domain branch, with a hop length of 1024.

Frequency Embedding One important aspect of this is that signals differs greatly in high and low frequencies. That is to say, music does not have the same translational invariance in frequency

as it does in time. As a solution, Demucs uses *frequency embedding* to assign an "identity label" to each frequency component. And the labels of frequencies that are close to each other on the frequency axis are similar.

Padding In the convoluting process, paddings are applied to ensure that the output dimensions match those of the waveform domain. If the input length is L and padding P is applied, the output of the convolution will be $\frac{L-K+2P}{S}+1$. In order to achieve an output length of $\frac{L}{S}$, the padding P needs to be $\frac{K-S}{2}$ [35].

Convoluting In the time-frequency matrix, Demucs uses a special convolution method, which performs 2D convolutions along the frequency axis instead of the time axis. For an Short-Time Fourier Transform input window of length N, $\frac{N}{2}+1$ frequency points are generated. The highest point was dropped to make the number of frequency points a power of 2. Consequently, the input C_{in} of the first layer is 2048 frequency points. After performing the same channel operations as in the time domain, the output of the fourth layer is also 384 channels. The output of is then fed into a Cross-Domain Transformer module, which will later be discussed Section B-4..

The remaining part of the frequency domain adopts the same U-Net structure as the time domain. In the last layer, the outputs of the two domains are combined.

B-4. Cross-Domain Transformer

The latest version of Demucs replaces the BiLSTM with a Transformer, which enhances the model's ability to understand longer audio sequences. The innovative point is that two kinds of attention mechanisms, self-attention and cross-attention, are used in a complementary way.

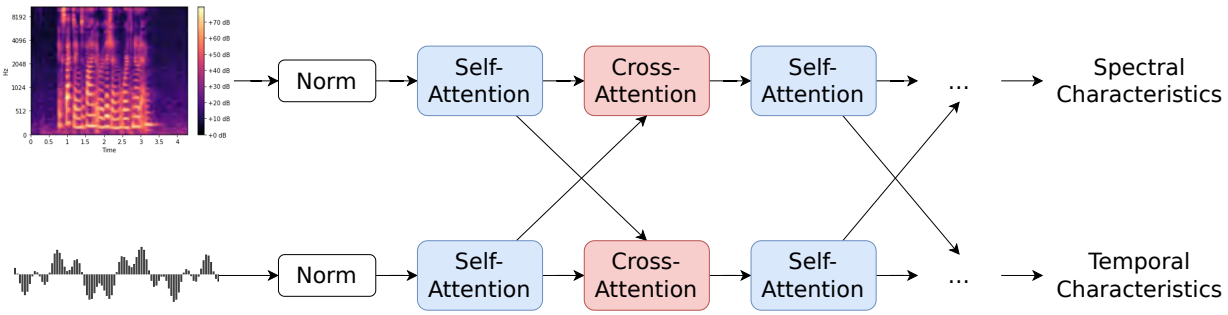


Fig 8: Illustration of the Transformer architecture used in Demucs.

Self-Attention In each domain, the self-attention mechanism is used to discover specific patterns in the music signal. When similar patterns are encountered again, the model can better separate them.

Cross-Attention The cross-attention mechanism allows the time and frequency domain information to communicate with each other. In the time-to-frequency direction, the waveform features are used as Query, while the spectral features are used as Key and Value. In the frequency-to-time direction, the spectral features are used as Query, while the waveform features are used as Key and Value.

Additionally, Hybrid Transformer Demucs utilises a *sparse attention mechanism* that employs Locality-Sensitive Hashing (LSH) to identify the most relevant positions. It only computes the 10% most relevant connections while ignoring the 90% less relevant ones. This significantly reduces computational pressure.

However, it is essential to note that more data is required to fully use these advantages, which is consistent with findings in other fields.

C. Rationale for Gaussian Noise Injection

To address noise issues in models, many approaches choose to add a neural network for denoising before the Music Source Separation model. However, this approach has several potential drawbacks. The preliminary denoising network might remove signal features that are crucial for the separation task. Additionally, cascading two models increases computational overhead, and will result in worse performance compared with directly incorporating noise into the Music Source Separation model training. For instance, Nazare et al. [47] compared preliminary denoising versus noise injection training methods, and proves the effectiveness of the latter. Goodfellow et al. also mentioned in their well-known deep learning book [21] that applying noise to the input is a method to enhance model robustness.

Previous study have shown a significant preference for optimizing models using Gaussian noise in the training phase [23, 33]. In audio tasks, the use of Gaussian noise injection also has its advantages.

Optimal Statistical Properties Gaussian noise has optimal properties in mathematical theory. This can be explained using *Maximum Likelihood Estimation* (Maximum Likelihood Estimation). For many objective functions, the Gaussian noise assumption leads to Maximum Likelihood Estimation [66]. If the noise in the data follows a Gaussian distribution, a mathematical "coincidence" occurs: Assume the true value is μ, and the observed value is $x = \mu + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$ (i.e., the noise follows a normal distribution with mean 0 and variance σ^2). In this case, finding the Maximum Likelihood Estimation is equivalent to minimizing the *Mean Squared Error* (Mean

Squared Error):

$$\text{MLE} \Leftrightarrow \text{MSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{8}$$

This provides a theoretical basis for using the Mean Squared Error loss function.

Natural Distribution Characteristics In nature, noise is often the result of the superposition of multiple random processes. For traditional methods based on Short-Time Fourier Transform, the spectral coefficients are often treated as complex Gaussian distribution and are independent of each other [24, 71]. In statistical terms, they are also often considered as additive independent random processes. Therefore, generally noise is the sum of multiple independent sources, the Central Limit Theorem ensures that the observed noise process tends to follow a Gaussian distribution [24]. From a mathematical perspective, Gaussian noise is only determined by mean and variance, making it the *Maximum Entropy Probability Distribution* [70].

Training Stability During the model training process, overfitting is a common issue. The introduction of Gaussian noise serves as a regularization technique, and enhances the model's generalisation capabilities. Multiple studies have demonstrated that introducing Gaussian noise to training phase is mathematically equivalent to incorporating penalty terms into the model's loss function [54, 42, 6, 2, 22]. Additionally, in deep neural networks, particularly in generative adversarial networks, Gaussian noise is often used to enhance training stability [61, 57, 56]. For general deep neural networks, training with Gaussian noise enables the model to adapt to various forms of perturbations beyond Gaussian noise [71].

This section has introduced the architecture of the Hybrid Transformer Demucs model and the theoretical basis for Gaussian noise injection. From the perspective of model architecture, it

analyzes how Hybrid Transformer Demucs combines time-domain and frequency-domain processing to capture multidimensional audio characteristics, and how the Transformer module facilitates information exchange between the two branches. The discussion of Gaussian noise injection provides the basis for treating noise-stable fine-tuning as a lightweight adaptation strategy for an LLM-facing music perception interface.

IV. Implementation

This section describes the experimental design and implementation details, including the experimental environment, dataset selection, noise injection mechanism, training details, and evaluation metrics. The experimental design emphasizes the implementation and parameter settings of Gaussian noise injection in order to evaluate whether training-time perturbation improves the robustness of the music source separation interface under noisy input conditions.

A. Experimental Environment and Dataset

Fine-tuning experiments were conducted on an Nvidia A100 GPU with 40GB of memory using single-precision format. Evaluation was performed separately on an Nvidia 3060 12GB GPU. This separation of training and evaluation allowed the same evaluation protocol to be applied consistently to all fine-tuned models.

The dataset used in this study is MUSDB18-HQ [53], which contains 150 full-length songs from various genres. It consists of 100 training songs and 50 test songs. Each song includes five uncompressed stereo WAV files: mixture, drums, bass, other, and vocals, sampled at 44.1kHz. This stem structure matches the LLM-facing interface formulation in this paper, where separated vocals, drums, bass, and other stems are interpreted as musical evidence for downstream understanding.

Alternative datasets were considered but not used in the reported experiments. MoisesDB [50] provides a larger six-track setup, which is less directly aligned with the four-stem MUSDB18-HQ configuration used by Hybrid Transformer Demucs. WHAM! [69] was also considered as a source of external noise, but its short noise segments and partial music contamination would require additional preprocessing. Therefore, this study uses controlled Gaussian perturbations to isolate the effect of noise injection strategy.

B. Noise Injection

Noise injection is implemented through a `GaussianNoiseWrapper` class, following the wrapper pattern used by other Demucs data augmentation modules such as `RepitchedWrapper`. The wrapper injects noise when samples are loaded, so the original MUSDB18-HQ files remain unchanged.

The main parameters are `variance`, `proba`, `apply_to_mix`, and `apply_to_sources`. The `variance` parameter controls noise intensity, while `proba` controls how often a sample is perturbed during training. The two target parameters define the two strategies compared in this paper: mixture-only injection and mixture-and-source injection. Mixture-only injection simulates noisy input with clean target evidence, whereas mixture-and-source injection corrupts both the input mixture and the target stems.

The evaluation pipeline can also add controlled noise to test samples independently of training. This allows the same baseline and fine-tuned models to be compared under different noisy input conditions.

C. *Training Details*

The training process uses the PyTorch implementation of Demucs [16], and experiments are managed with Dora [15]. Each experiment is identified by a unique signature determined by its hyperparameter configuration, which supports reproducible grid evaluation.

Following the architecture described in Section B., music data is processed in both time and frequency domains, encoded by convolutional layers, passed through Transformer layers, and decoded through skip-connected decoders. Fine-tuning starts from the pre-trained `htdemucs` model with experiment signature `955717e8`.

Preliminary inspection showed that noise variance above 0.15 heavily obscured the original musical structure in both listening and waveform/spectrogram analysis. The main experiments therefore use smaller variance values to study robustness adaptation without completely masking the source characteristics.

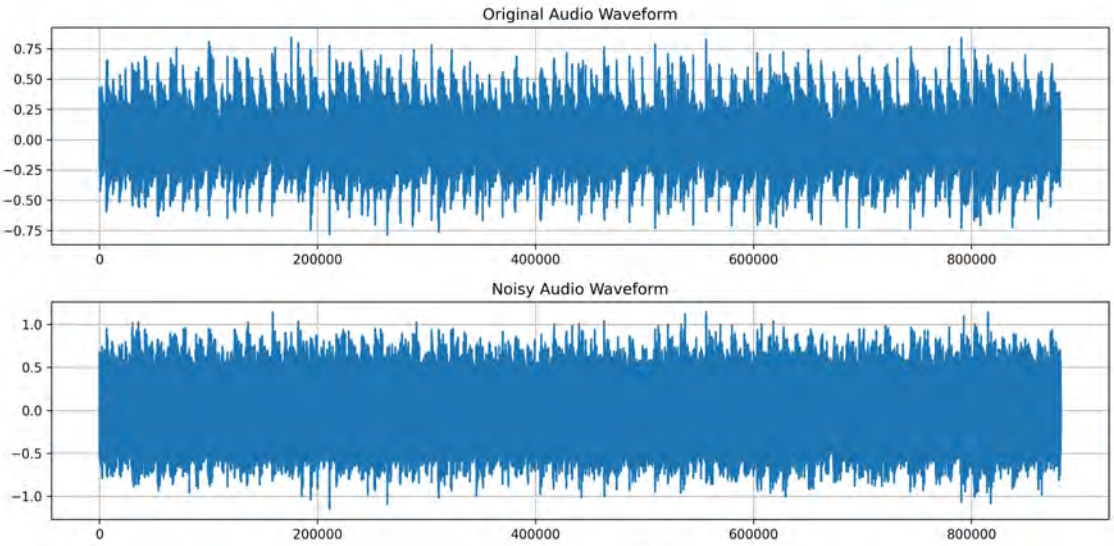
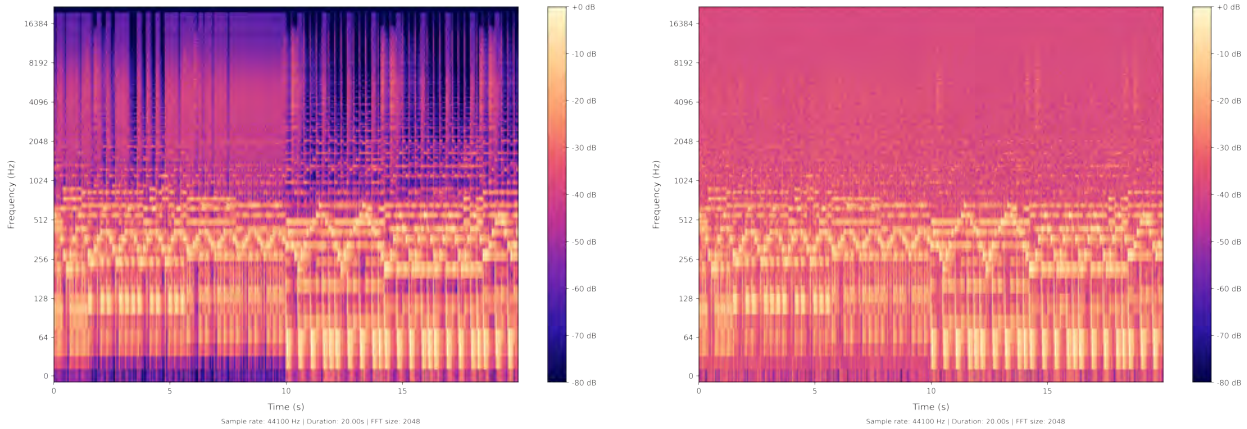


Fig 9: Waveform of the Music, with excessive variance (0.25) of noise injected.

The experimental design contains two noise injection categories: mixture-only injection and mixture-and-source injection. For each category, five variance levels (0.001, 0.003, 0.005, 0.01,



(a) Original Spectrogram (b) Excessive Noise Injected

Fig 10: Comparison of Spectrograms with Excessive Noise Injected

0.02) were combined with three injection probabilities (0.4, 0.6, 0.8), resulting in 30 fine-tuned models.

Table 2: Model Configurations with Different Noise Injection Strategies

Target	Var.	Prob.	Signature	Target	Var.	Prob.	Signature
Mix & Source	0.001	0.8	326b0d1f	Mix Only	0.001	0.8	bd243b64
		0.6	d1148df2			0.6	856fe329
		0.4	344803ff			0.4	d49b2b74
	0.003	0.8	6eece43f		0.003	0.8	f87be88f
		0.6	e0b3313c			0.6	aa3b6ebb
		0.4	7df26c74			0.4	292a7207
	0.005	0.8	2d3035dd		0.005	0.8	db03d81b
		0.6	debf633b			0.6	b5061298
		0.4	16862099			0.4	b9dd3bb4
	0.01	0.8	f3b99dbf		0.01	0.8	114aeffd
		0.6	67284ea1			0.6	21d70436
		0.4	c9d88364			0.4	b1ff6312
	0.02	0.8	c611a32a		0.02	0.8	14b72251
		0.6	d6900f95			0.6	1e679987
		0.4	6c20752c			0.4	7bbbe888

Table 2 records the relationship between each experimental configuration and its corresponding signature.

All fine-tuning experiments reached their minimum loss at approximately 95 iterations (Figure 11). The same early-stopping point was therefore used across fine-tuning runs to keep the

comparison consistent.

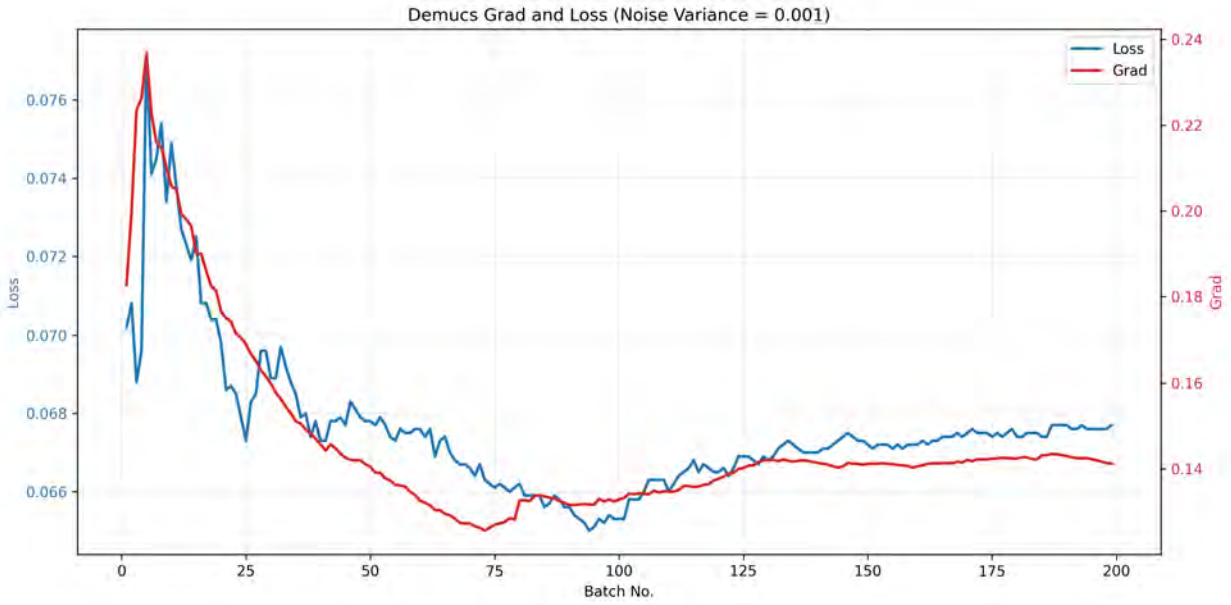


Fig 11: Loss Curve of the Training Process. Note the location of the lowest point.

D. Evaluation Metrics

The *Signal-to-Distortion Ratio* (Signal-to-Distortion Ratio) is proposed by Vincent et al. [65]. It was calculated as:

$$SDR = 10\log_{10}\frac{\sum_n \|\mathbf{x}(n)\|^2 + \epsilon}{\sum_n \|\mathbf{x}(n) - \hat{\mathbf{x}}(n)\|^2 + \epsilon}, \tag{9}$$

where $\mathbf{x}$ is the original source, $\hat{\mathbf{x}}$ is the estimated source, and $\epsilon = 10^{-7}$ is a small constant to avoid zero division.

SiSEC 2018 [59] introduced the `BSS Eval v4` as the toolbox for evaluating Blind Source Separation, which is also the evaluation metrics of the Hybrid Transformer Demucs paper [55]. This method calculates the Signal-to-Distortion Ratio for each frame of one second in the whole song and takes the median of all frames as the final Signal-to-Distortion Ratio result.

However, `BSS Eval v4` is very computationally expensive. Therefore, this study uses the "**global SDR**" proposed in MDX 2021 instead. The global Signal-to-Distortion Ratio directly uses Equation 9 and calculates the Signal-to-Distortion Ratio for the entire song. According to experiments, this metrics is highly correlated with `BSS Eval v4`, reaching 0.9 [46], so it can be used. The Signal-to-Distortion Ratio of the whole song is calculated as the average of the Signal-to-Distortion Ratio of each source.

$$SDR = \frac{1}{4}(SDR_{\mathrm{b}} + SDR_{\mathrm{d}} + SDR_{\mathrm{v}} + SDR_{\mathrm{o}}) \tag{10}$$

The Signal-to-Distortion Ratio mentioned later in this paper refers to this global SDR. To distinguish it from the `BSS Eval v4`, it is referred to as Global SDR.

Each model in Table 2 that was obtained from fine-tuning was evaluated using Global SDR. The evaluation process also introduces noise injection parameters that are the same as those used during training, but is independent of the parameters of the training process. It was observed that the "average Signal-to-Distortion Ratio" metric fluctuated more significantly. This is possibly related to the diversity of the testing samples. Therefore, the *median* of the Signal-to-Distortion Ratio was used as the final evaluation metric. 150 evaluations was performed.

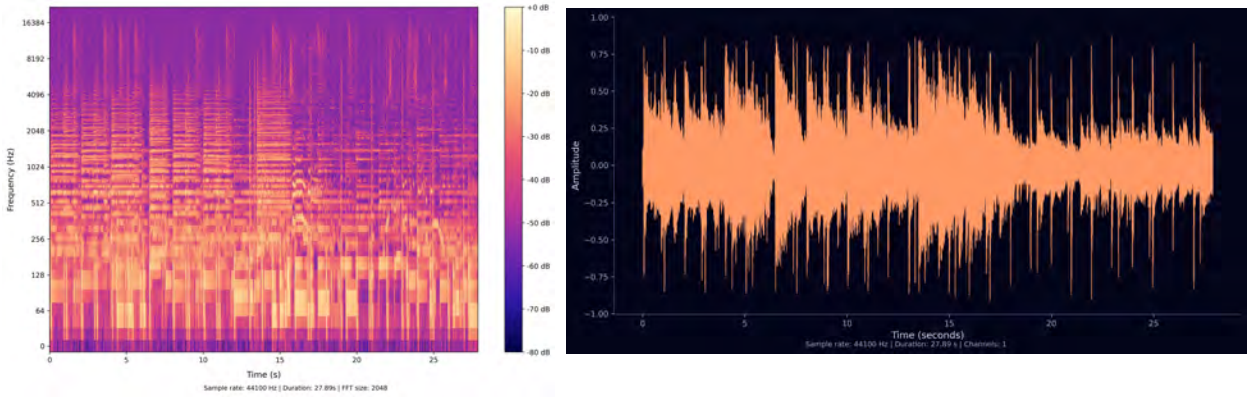
This section describes the experimental design and implementation details. It introduces the hardware environment and the MUSDB18-HQ dataset, explains the rationale for selecting this dataset over alternatives, and describes the implementation mechanism of Gaussian noise injection, including the design of the GaussianNoiseWrapper class and its parameter configuration options. It also discusses model initialization, early stopping during fine-tuning, the noise parameter ranges determined through preliminary testing, and the adoption of global Signal-to-Distortion Ratio as

the evaluation metric. These details provide the basis for analyzing the robustness of the LLM-facing music source separation interface.

V. Experiment Results and Analysis

This section evaluates whether Gaussian noise injection improves the robustness of the music source separation interface and analyzes how different noise parameters affect stem-level evidence quality.

A music clip of 27 seconds is used to demonstrate the performance of the model. Figure 12 shows the spectrogram and waveform of the original music clip, while Figure 13 and 14 shows the separated results.

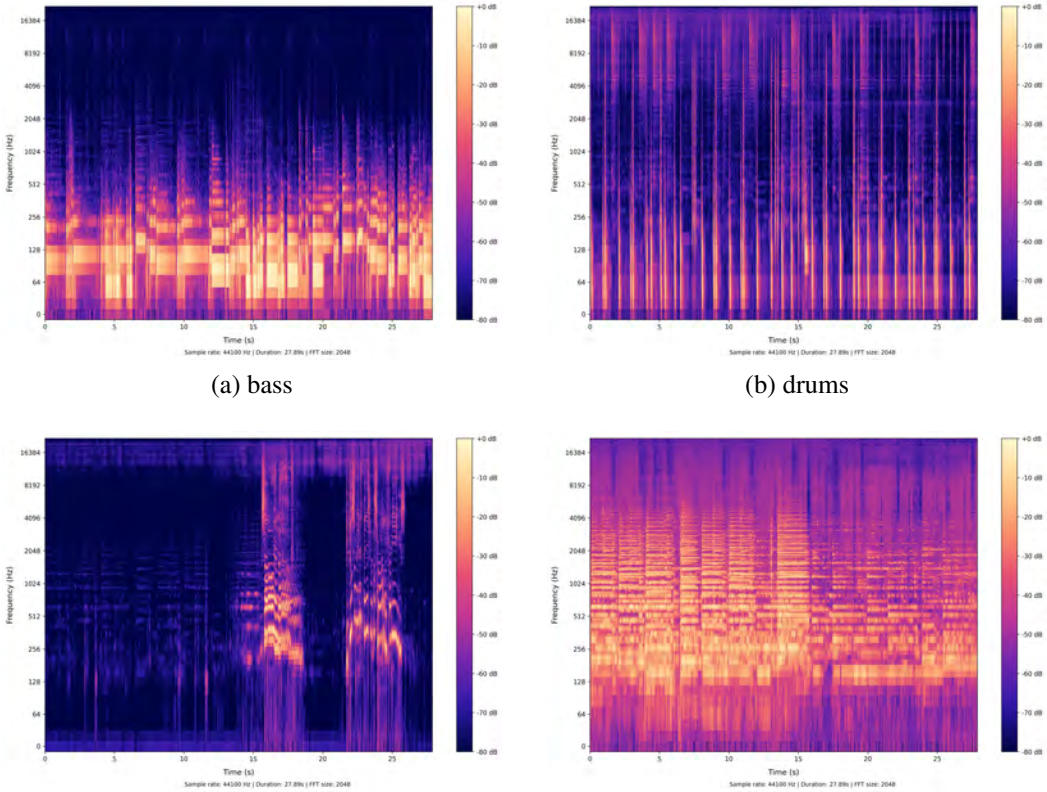


(a) spectrogram (b) waveform

Fig 12: Waveform and spectrogram of the mixed audio injected with noise.

A. *Quantitative results*

To visually compare the effects of different noise injection strategies, two heatmaps (Figure 15) were generated for two experimental scenarios: noise injection to both mixture and source tracks (**Scenario-to_both**, blue heatmap) and noise injection solely to the mixture track (**Scenario-to_mix**,



(a) bass (b) drums

(c) vocals (d) other

Fig 13: Spectrograms of the separated stems.

green heatmap). These heatmaps report Global SDR across different noise parameter combinations. In the proposed LLM-facing interpretation, these scores are used as a proxy for the stability of stem-level musical evidence under noisy input conditions. Analysis of the heatmaps reveals that changes caused by parameter variations are generally smooth, without obvious anomalies or discontinuities. This suggests that the robustness effect of noise injection is not random, but is systematically related to the injection probability, variance, and target of perturbation.

Baseline Interface Robustness The baseline model's Global SDR exhibits a clear downward trend as test noise intensity increases, indicating that the quality of extracted stem-level evidence degrades under noisy input conditions. This degradation matters for an LLM-facing perception interface because residual interference in vocals, drums, bass, or accompaniment can affect the

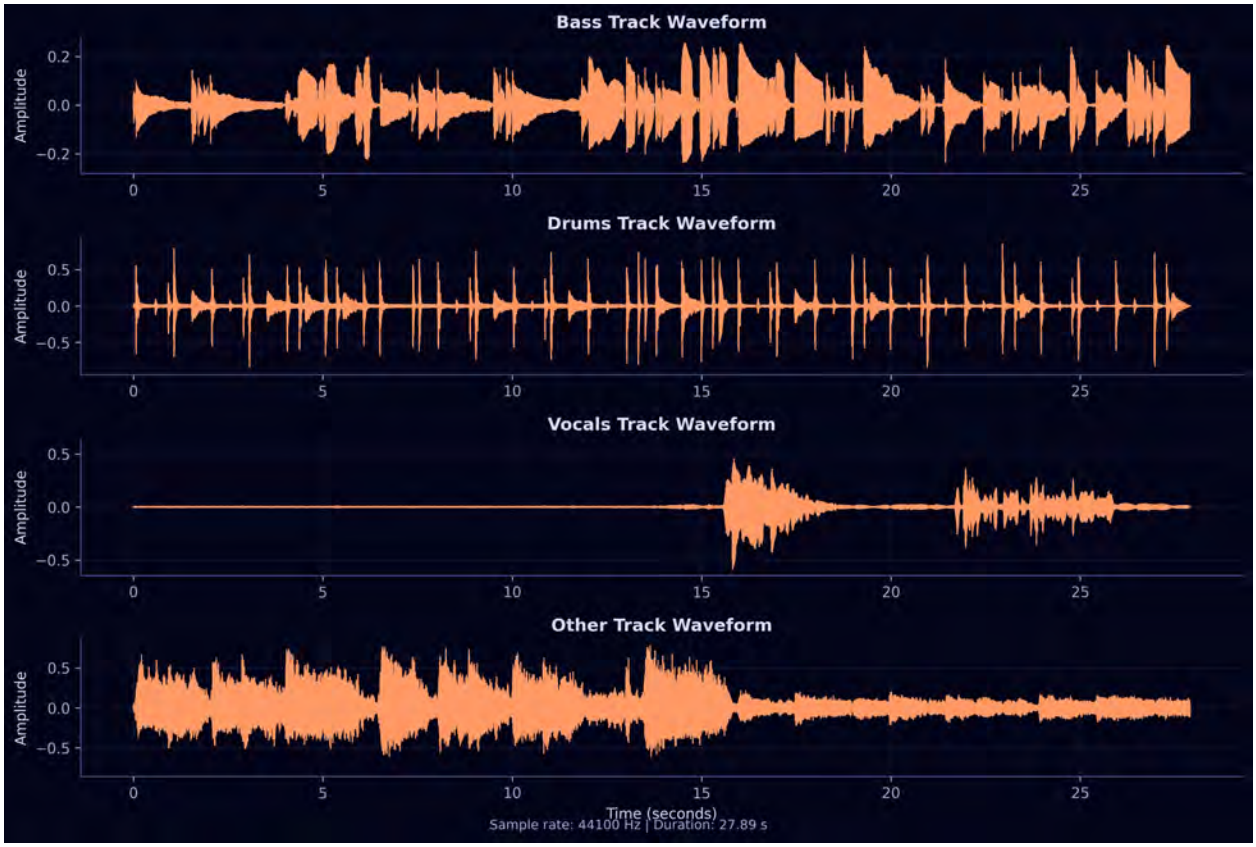
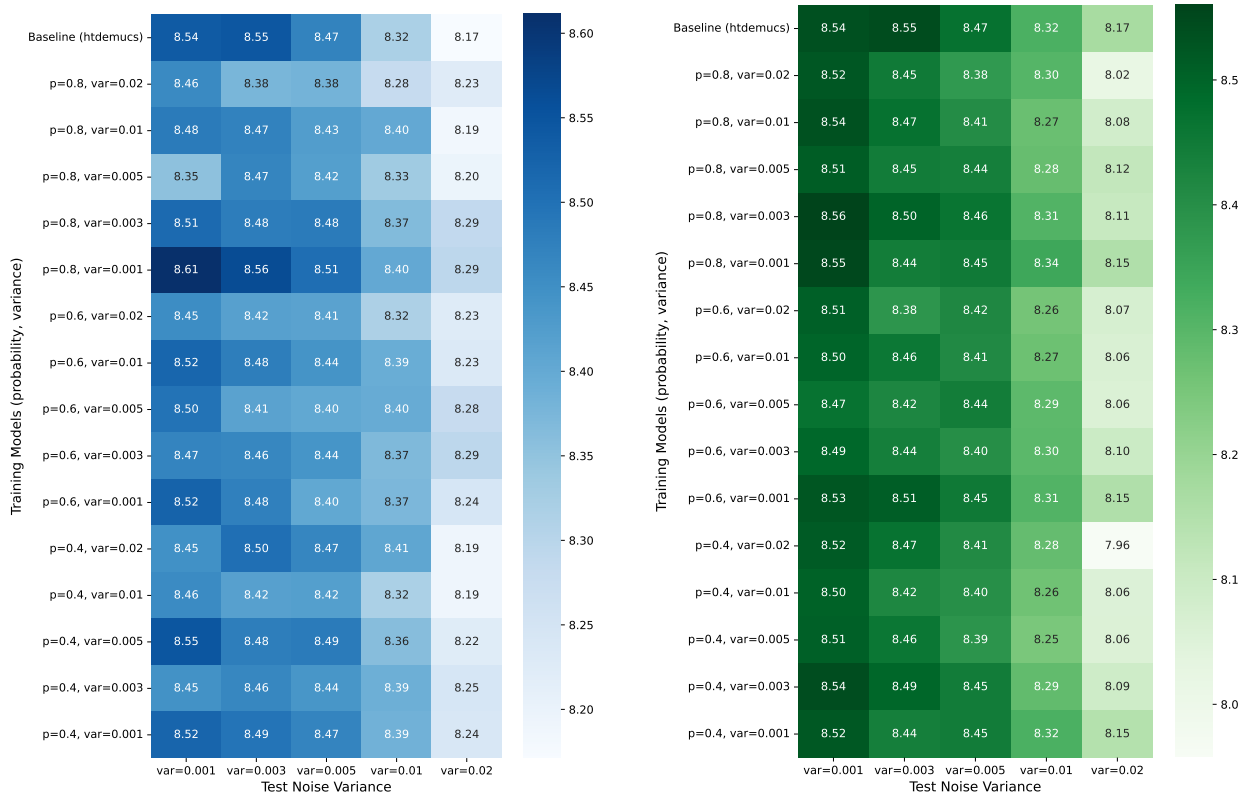


Fig 14: Waveforms of the separated stems.

reliability of downstream music understanding. The two injection settings also reveal different robustness patterns: in Scenario-to_both, the baseline model shows a performance decline of 0.37 dB from the lowest to the highest noise intensity, whereas in Scenario-to_mix, the baseline outperforms most fine-tuned models.

Fine-tuned Interface Robustness The fine-tuned models are also affected by variations in test noise intensity, but the mixture-only strategy shows a clearer robustness benefit. Results indicate that models trained with higher injection probabilities (0.8) typically outperform those trained with lower probabilities (0.4), suggesting that frequent exposure to noisy mixtures can improve the stability of the separation interface. Figure 16 also shows that, when the injection probability is fixed, changing the noise variance does not produce a simple linear effect. This indicates that

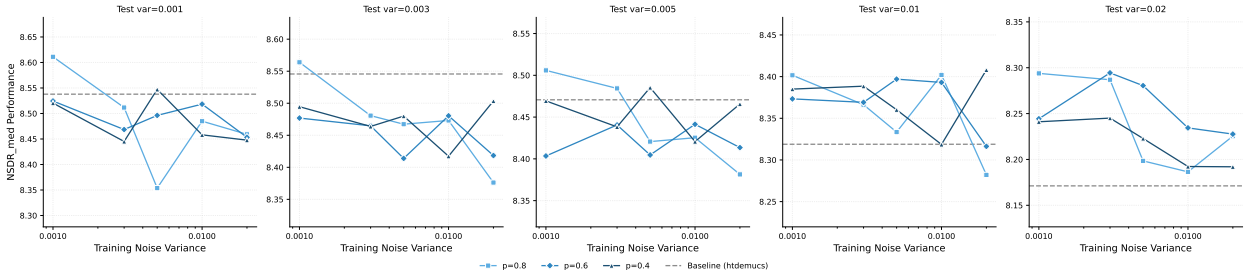


(a) Scenario-to_both (b) Scenario-to_mix

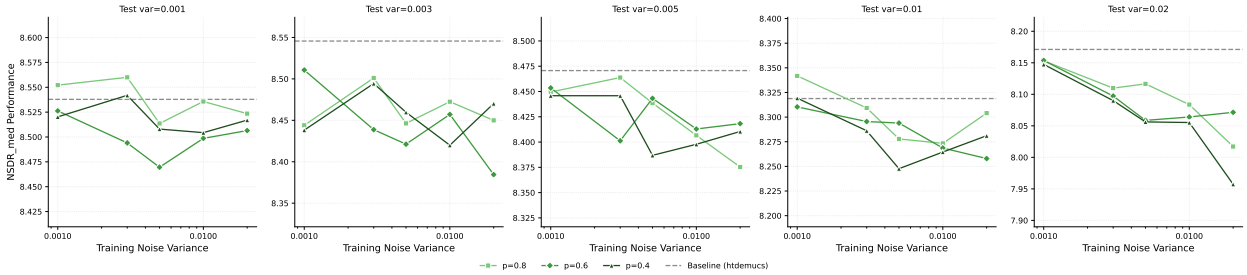
Fig 15: Heatmaps of Signal-to-Distortion Ratio performance metrics for different noise injection strategies.

robustness adaptation depends on both the amount and consistency of training-time perturbation. Among all experimental configurations, the model fine-tuned with a noise injection probability of 0.8 and variance of 0.001 achieves the highest Global SDR values across the tested noise conditions. Therefore, this configuration is selected as the *top-performing interface configuration* within the evaluated parameter grid.

Best Interface Configuration For this configuration, robustness exceeds the baseline across the tested noise intensity levels. Under high-intensity noise conditions, the improvement in Global SDR compared with the baseline reaches 0.12 dB. The absolute gain is modest, but it is obtained without changing the model architecture or adding inference-time computation. From the perspec-



(a) Scenario-to_both



(b) Scenario-to_mix

Fig 16: Performance comparison of the baseline and fine-tuned models under different noise conditions.

tive of an LLM-facing music perception interface, this result suggests that training-time mixture perturbation can make stem-level evidence slightly more stable when the input mixture is noisy.

B. Analysis of the Results

B-1. Human Auditory Perception

Since the quality of music separation is highly subjective, this section discusses audible phenomena observed during inspection rather than presenting a full perceptual listening study.

Although Global SDR improves slightly, the perceptual experience still differs across stems. Drums and bass generally maintain clearer auditory structure, which is important because these stems provide rhythm, groove, and low-frequency evidence for music understanding. In contrast, noticeable flaws remain in vocals and other instrument tracks, such as drum attack sounds appearing in the vocal track or vocal fragments leaking into the accompaniment track. For an LLM-facing interface, these errors are meaningful because vocals are closely related to lyrics and singing voice

understanding, while accompaniment leakage can distort harmonic and timbral evidence. When the noise intensity increases beyond 0.3, these flaws become more pronounced and the separated stems gradually trend toward a muddled sound. This suggests that noise disrupts both transient and long-term musical relationships, which in turn affects the reliability of the extracted evidence.

B-2. Impact of Noise on Inference Phase

While numerous studies on noise injection have been conducted in the image domain [57, 47], Gaussian noise in audio exhibits a series of distinctive characteristics. First, audio signals typically demonstrate **non-stationary** characteristics, meaning that noise effects may vary considerably across different time segments. Second, real-world audio noise usually demonstrate obvious frequency-dependent characteristics, such as noise from electrical interference or friction. Furthermore, audio signals are extremely sensitive to **phase perturbations**. A small phase shift can mean the difference between a clear and blurry sound, or the position of a sound source in stereo cases. Gaussian noise simultaneously affects both amplitude and phase information. From a perceptual perspective, noise in different frequency bands can have different impacts on auditory quality, with noise in critical bands such as human voice frequencies often having a stronger effect on perceived quality.

Concerning the Hybrid Transformer Demucs model itself, some noise-resistant capabilities are inherently in its design. As discussed in Section B., Demucs processes time and frequency domain features simultaneously, allowing temporal disturbances and noise to potentially be smoothed or filtered in the frequency domain representation, vice versa. More significantly, the Transformer architecture at the model's core is proved to have, to some extent, noise resistance capabilities [5, 10]. Transformers utilise a self-attention mechanism to capture long-range dependencies, thus enables

the model to focus on important features while filtering out noise interference when processing noisy inputs, and therefore enhancing model robustness. This is one of the reasons why only **hybrid** models are considered in Section D-3..

The core advantage of Transformer architecture lies in its distinctive self-attention mechanism, which captures long-range dependencies within input sequences. This enables the model to dynamically focus on truly important features while disregarding noise interference when processing noisy inputs, fundamentally enhancing model robustness. In audio separation tasks, this characteristic is particularly valuable, as musical signals typically contain complex patterns and structures spanning multiple time steps, and the Transformer architecture excels at capturing these long-range correlations.

B-3. Analysis on Noise Injection Strategy

The experimental results revealed an intriguing phenomenon requiring deeper analysis: in the Scenario-to_both condition (where noise was injected into both mix and source tracks), the fine-tuned models unexpectedly exhibited reduced robustness. This strategy was initially considered because fine-tuning was expected to help the model recover "*clean stems*" from noisy mixtures. The counterintuitive outcome can be analysed as follows:

The main reason is the **task inconsistency** introduced by different noise targets. In the mixture-only setting, the model receives a noisy mixture while the reference sources remain clean. This is aligned with the LLM-facing interface objective: the input may be noisy, but the desired evidence for downstream reasoning should be as clean and source-specific as possible. In contrast, the mixture-and-source setting also corrupts the target stems. This weakens the clean reconstruction objective and can teach the model to reproduce noisy evidence rather than recover clean musical

cues.

This interpretation is related to the idea of noise stability regularization, where robustness depends on controlling the discrepancy between perturbed inputs and desired outputs [27]. It is also consistent with findings that mismatches between training and testing noise patterns can reduce model adaptability [68]. In this study, the key issue is not only whether noise is present, but whether the training target represents the clean stem-level evidence expected by the downstream interface.

The mixture-and-source setting also introduces a clear **target distribution shift**. If the target stems are noisy during fine-tuning, the model is optimized toward a different evidence distribution from the clean stems used as the desired interface output. For music-aware multimodal systems, this matters because language-level reasoning benefits from stable and interpretable evidence: vocals should preserve lyrical content, drums and bass should preserve rhythmic structure, and accompaniment should preserve harmonic and timbral context. The mixture-only strategy better matches this requirement by simulating noisy input while preserving clean target evidence.

C. Limitations

As shown in Table 1, besides MUSDB-18, Hybrid Transformer Demucs is trained with an extra 800 songs. These songs are obtained from a meticulous process to ensure data diversity and quality [55]. However, because the extra data is not publicly available, and copyright-free music data is very rare, this study only uses the MUSDB-18 dataset for training and testing. This process may not fully perform the model's capabilities, because Transformers are known to be "data-hungry" architectures [41]. The original model seen a significant drop in Signal-to-Distortion Ratio if not trained with the extra data [55].

During noise addition, the noise is evenly distributed across all four stems, which is a simplified assumption. In reality, as shown in 13, the separated results indicate that noise is primarily directed to the "other" category. The question of how to appropriately distribute noise requires further research. Some papers have explored more sophisticated noise allocation methods, such as the masking approach implemented by Microsoft in [9].

Furthermore, due to computational resource and budget constraints, the granularity chosen for training and evaluation processes was not sufficiently detailed. With additional resources, a more specific analysis of impact trends might have been possible.

This section evaluates the effectiveness of noise injection training methods for music source separation through quantitative analysis and qualitative assessment. Heat maps visualize experimental results from two primary noise injection strategies: injecting noise solely into mixture tracks and injecting noise simultaneously into mixture and source tracks. The analysis shows clear differences between the two strategies. In the "Scenario-to_both" setting, the baseline model outperforms most fine-tuned models. In the "Scenario-to_mix" context, a model fine-tuned with 0.8 probability and 0.001 variance parameters achieves the highest Global SDR among the tested configurations, with improvements particularly visible under high-intensity noise conditions.

Perceptual analysis indicates that despite improvements in Signal-to-Distortion Ratio metrics, noticeable imperfections persist in vocal and other instrument tracks. The discussion explores the distinctive characteristics of audio-domain noise and highlights how the inherent noise-resistant capabilities of the Transformer architecture may contribute to robust performance. The underperformance of the "Scenario-to_both" strategy is attributed primarily to inconsistencies between training and testing objectives, as well as data distribution shifts.

VI. Conclusions and Future Work

This section summarizes the findings and outlines future directions.

A. Conclusions

This paper studies robust music source separation as an LLM-facing perception interface for music-aware multimodal systems. The central motivation is that language-level reasoning about lyrics, vocals, instrumentation, rhythm, and performance characteristics depends on stable stem-level musical evidence extracted from noisy and polyphonic music inputs.

Using Hybrid Transformer Demucs as the baseline separation interface, Gaussian noise injection was evaluated as a lightweight training-time adaptation strategy. The experiments show that injecting noise only into mixture tracks is more effective than injecting noise into both mixture and source tracks. The best mixture-only configuration, with noise probability 0.8 and variance 0.001, provides a small but consistent improvement in source-to-distortion ratio under high-intensity noisy test conditions.

These results suggest that when music source separation is used as an evidence extraction interface, preserving clean target stems during training is important. Noise injected into both the input mixture and the target stems can weaken the clean reconstruction objective and reduce robustness under noisy inference conditions. The proposed approach does not alter the Hybrid Transformer Demucs architecture and does not introduce additional inference cost, making it suitable as an input-side robustness adaptation strategy for future music-aware multimodal large language model pipelines.

This study does not claim to directly train or improve a language model. Instead, it improves the robustness of the music perception component that can provide stem-level evidence to downstream

audio encoders and language reasoning modules.

B. Future Work

Realistic Noise and Acoustic Conditions While this study demonstrates the usefulness of Gaussian noise injection, real music recordings often contain more complex perturbations, including audience noise, room reverberation, microphone artefacts, compression distortion, and frequency-dependent coloured noise. Future work should evaluate whether Gaussian-noise adaptation generalizes to these conditions. A useful next step would be to construct noisy MUSDB18-HQ variants with real environmental noise and multiple signal-to-noise ratio levels, then compare clean-test and noisy-test performance across the same mixture-only and mixture-and-source strategies.

Stem-Aware Noise Distribution Strategies This study shows that different noise allocation methods affect the quality of extracted stem-level evidence. Future research could therefore explore stem-aware or frequency-aware noise injection strategies. Vocals may require stronger protection because they carry lyrical and singing voice evidence, whereas drums and bass may require preservation of transient and low-frequency rhythmic cues. Human auditory perception models could also be used to prioritize frequency bands that are important for downstream music understanding.

LLM-Facing Downstream Proxy Evaluation The present work uses Global SDR as a proxy for stem-level musical evidence quality. Future work should connect the separation interface to downstream music-understanding tasks more directly. Possible proxy evaluations include lyrics or singing voice recognition on separated vocal stems, music tagging consistency on separated stems, audio-language retrieval using contrastive audio-language models, and multimodal music question

answering. These evaluations would test whether more robust source separation improves not only acoustic reconstruction quality, but also the usefulness of extracted evidence for music-aware multimodal large language model pipelines.

Integration with Multimodal Music Systems Future studies could integrate the robust separation interface with audio encoders and language reasoning modules in an end-to-end music understanding pipeline. Such systems could compare raw noisy mixtures, baseline separated stems, and noise-adapted separated stems as inputs to the same downstream model. This would allow a clearer assessment of when front-end robustness improves language-level reasoning, while keeping the boundary clear between improving the perception interface and improving the language model itself.

References

[1] Andrea Agostinelli, Timo I. Denk, Zalan Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text. *arXiv*, abs/2301.11325, 2023.

[2] Guozhong An. The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. *Neural Computation*, 8(3):643–674, April 1996.

[3] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[4] Ashwin Bellur and Mounya Elhilali. Bio-mimetic attentional feedback in music source sep-

aration. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8718–8722. IEEE, 2020.

[5] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding Robustness of Transformers for Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10231–10241, 2021.

[6] Chris M. Bishop. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation*, 7(1):108–116, January 1995.

[7] Zalan Borsos, Raphael Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matthew Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533, 2022.

[8] J.F. Cardoso and A. Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proceedings F Radar and Signal Processing*, 140(6):362, 1993.

[9] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, October 2022.

[10] Chen Cheng, Xinzhi Yu, Haodong Wen, Jingsong Sun, Guanzhang Yue, Yihao Zhang, and Zeming Wei. Exploring the Robustness of In-Context Learning with Noisy Labels. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, April 2025.

[11] Seungjin Choi, Andrzej Cichocki, Hyung-Min Park, and Soo-Young Lee. Blind source separation and independent component analysis: A review. *Neural Information Processing-Letters and Reviews*, 6(1):1–57, 2005.

[12] Woosung Choi, Minseok Kim, Jaehwa Chung, and Soonyoung Jung. LaSAFT: Latent source attentive frequency transformation for conditioned source separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 171–175. IEEE, 2021.

[13] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 933–941. PMLR, July 2017.

[14] Alexandre Défossez. Hybrid Spectrogram and Waveform Source Separation, August 2022.

[15] Alexandre Défossez. Dora: An experiment management framework. GitHub repository, `https://github.com/facebookresearch/dora`, 2023. Version 0.1.13a5, commit 8e5d905, MIT License.

[16] Alexandre Défossez. Demucs: Music source separation. GitHub repository, `https://github.com/adefossez/demucs`, 2024. Version 4.1.0a3, commit b9ab48c, MIT License.

[17] Alexandre Défossez, Yossi Adi, and Gabriel Synnaeve. Differentiable Model Compression via Pseudo Quantization Noise, October 2022.

[18] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music Source Separation in the Waveform Domain, April 2021.

[19] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. *ICASSP 2023 - 2023 IEEE Interna-*

tional Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5, 2023.

[20] Giorgio Fabbro, Stefan Uhlich, Chieh-Hsin Lai, Woosung Choi, Marco Martínez-Ramírez, Weihsiang Liao, Igor Gadelha, Geraldo Ramos, Eddie Hsu, Hugo Rodrigues, Fabian-Robert Stöter, Alexandre Défossez, Yi Luo, Jianwei Yu, Dipam Chakraborty, Sharada Mohanty, Roman Solovyev, Alexander Stempkovskiy, Tatiana Habruseva, Nabarun Goswami, Tatsuya Harada, Minseok Kim, Jun Hyung Lee, Yuanliang Dong, Xinran Zhang, Jiafeng Liu, and Yuki Mitsufuji. The Sound Demixing Challenge 2023 - Music Demixing Track, January 2024.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[22] Yves Grandvalet, Stéphane Canu, and Stéphane Boucheron. Noise Injection: Theoretical Prospects. *Neural Computation*, 9(5):1093–1108, July 1997.

[23] Limin Gui, Bang Huang, and Wen-Qin Wang. Robust Detector Designing for FDA-MIMO Radar With Training Data in Gaussian Noise. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–17, 2025.

[24] Richard C. Hendriks, Timo Gerkmann, and Jesper Jensen. *DFT-domain Based Single-microphone Noise Reduction for Speech Enhancement: A Survey of the State-of-the-art*. Morgan & Claypool Publishers, 2013.

[25] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs), June 2023.

[26] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: A fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software*, 5:2154, June 2020.

[27] Hang Hua, Xingjian Li, Dejing Dou, Cheng-Zhong Xu, and Jiebo Luo. Improving Pretrained Language Model Fine-Tuning With Noise Stability Regularization. *IEEE Transactions on*

Neural Networks and Learning Systems, 36(1):1898–1910, January 2025.

[28] Jing Huang and Brian Kingsbury. Audio-visual deep learning for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7596–7599, May 2013.

[29] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4):411–430, June 2000.

[30] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural computation*, 9(7):1483–1492, 1997.

[31] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde. Singing voice separation with deep U-Net convolutional networks, October 2017.

[32] Minseok Kim, Woosung Choi, Jaehwa Chung, Daewon Lee, and Soonyoung Jung. KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing, November 2021.

[33] Rasmus Knabäck. *Impact of Gaussian Noise on Lora Training for Flux.1*. Bachelor's Thesis, Mälardalen University, School of Innovation, Design and Engineering, 2025.

[34] Qiuqiang Kong, Yin Cao, Haohe Liu, Keunwoo Choi, and Yuxuan Wang. Decoupling Magnitude and Phase Estimation with Deep ResUNet for Music Source Separation, September 2021.

[35] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[36] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999.

[37] Qiang Li, Deyu Chen, and Xin Guan. Music source separation method based on time and channel dual-dimensional sequential attention. *Acta Acustica*, 48(3):588–598, 2023.

[38] Yi Luo and Nima Mesgarani. Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266, August 2019.

[39] Yi Luo and Jianwei Yu. Music source separation with band-split RNN. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1893–1901, 2023.

[40] Ethan Manilow, Prem Seetharman, and Justin Salamon. *Open Source Tools & Data for Music Source Separation*. https://source-separation.github.io/tutorial, October 2020.

[41] Jiawei Mao, Honggu Zhou, Xuesong Yin, and Yuanqi Chang Binling Nie Rui Xu. Masked autoencoders are effective solution to transformer data-hungry, January 2023.

[42] K. Matsuoka. Noise injection into inputs in back-propagation learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):436–440, May 1992.

[43] Jingjing Meng and Yabin Xu. Separation method of music and voice based on ensemble learning. *Journal of Beijing Information Science & Technology University*, 38(3):27–34, 2023.

[44] Gabriel Mersy. Efficient Robust Music Genre Classification with Depthwise Separable Convolutions and Source Separation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(18):15972–15973, May 2021.

[45] Daniel Michelsanti, Zheng-Hua Tan, Shi-Xiong Zhang, Yong Xu, Meng Yu, Dong Yu, and Jesper Jensen. An Overview of Deep-Learning-Based Audio-Visual Speech Enhancement and Separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1368–1396, 2021.

[46] Yuki Mitsufuji, Giorgio Fabbro, Stefan Uhlich, Fabian-Robert Stöter, Alexandre Défossez, Minseok Kim, Woosung Choi, Chin-Yun Yu, and Kin-Wai Cheuk. Music Demixing Challenge 2021. *Frontiers in Signal Processing*, 1, January 2022.

[47] Tiago S. Nazaré, Gabriel B. Paranhos Da Costa, Welinton A. Contato, and Moacir Ponti. Deep Convolutional Neural Networks and Noisy Images. In Marcelo Mendoza and Sergio Velastín, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 10657, pages 416–424. Springer International Publishing, Cham, 2018.

[48] Nobutaka Ono, Zafar Rafii, Daichi Kitamura, Nobutaka Ito, and Antoine Liutkus. The 2015 Signal Separation Evaluation Campaign. In *Latent Variable Analysis and Signal Separation*, pages 387–395. Springer, Cham, 2015.

[49] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[50] Igor Pereira, Felipe Araújo, Filip Korzeniowski, and Richard Vogl. Moisesdb: A dataset for source separation beyond 4-stems, July 2023.

[51] Prof. K. G. Jagtap, Vivek Deshmukh, Maviya Mahagami, Himanshu Lohokane, and Rashi Kacchwah. An In-depth Review on Music Source Separation. *International Journal of Advanced Research in Science, Communication and Technology*, pages 256–260, March 2024.

[52] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December 2017.

[53] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. MUSDB18-HQ - an uncompressed version of MUSDB18, August 2019.

[54] Russell Reed, Seho Oh, and Robert J. Marks. Regularization using jittered training data. In

International Joint Conference on Neural Networks, volume 3, pages 147–152, 1992.

[55] Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid Transformers for Music Source Separation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, June 2023.

[56] Jangwon Seo, Hyo-Seok Hwang, Minhyeok Lee, and Junhee Seok. Stabilized GAN models training with kernel-histogram transformation and probability mass function distance. *Applied Soft Computing*, 164:112003, October 2024.

[57] Chaohua Shi, Kexin Huang, Lu Gan, Hongqing Liu, Mingrui Zhu, Nannan Wang, and Xinbo Gao. On the Analysis of GAN-based Image-to-Image Translation with Gaussian Noise Injection. In *The Twelfth International Conference on Learning Representations*, April 2024.

[58] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation, June 2018.

[59] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 Signal Separation Evaluation Campaign. In Yannick Deville, Sharon Gannot, Russell Mason, Mark D. Plumbley, and Dominic Ward, editors, *Latent Variable Analysis and Signal Separation*, pages 293–305, Cham, 2018. Springer International Publishing.

[60] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. Open-Unmix - A Reference Implementation for Music Source Separation. *Journal of Open Source Software*, 4(41):1667, September 2019.

[61] Yiyun Sun. Dugdale-GAN: Physical Dugdale Model Integrated Generative Adversarial Network for High-quality Crack Image Generation. In *2025 IEEE 5th International Conference on Power, Electronics and Computer Applications (ICPECA)*, pages 286–297, January 2025.

[62] Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji. MMDenseLSTM: An efficient

combination of convolutional and recurrent neural networks for audio source separation. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 106–110. IEEE, 2018.

[63] Naoya Takahashi and Yuki Mitsufuji. D3Net: Densely connected multidilated DenseNet for music source separation, March 2021.

[64] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. *arXiv*, abs/2310.13289, 2023.

[65] Emmanuel Vincent, Hiroshi Sawada, Pau Bofill, Shoji Makino, and Justinian P. Rosca. First Stereo Audio Source Separation Evaluation Campaign: Data, Algorithms and Results. In *Independent Component Analysis and Signal Separation*, pages 552–559. Springer, Berlin, Heidelberg, 2007.

[66] Tuomas Virtanen. Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1066–1074, March 2007.

[67] Bin Wang and Ning Chen. An End-to-End Singing Voice Separation Model Based on Residual Attention U-Net. *Journal of East China University of Science and Technology*, 47(5):619–626, 2021.

[68] Song Wang, Zhen Tan, Ruocheng Guo, and Jundong Li. Noise-Robust Fine-Tuning of Pretrained Language Models via External Guidance, November 2023.

[69] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. WHAM!: Extending Speech Separation to Noisy Environments, July 2019.

[70] Xueqiong Yuan, Jipeng Li, and Ercan Engin Kuruoglu. Robustness enhancement in neural networks with alpha-stable training noise. *Digital Signal Processing*, 156:104778, January 2025.

[71] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the Robustness of Deep Neural Networks via Stability Training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4488, 2016.