An Incremental Framework for Sound Source Separation

Wonjun Park and Kenny Q. Zhu
Arlington Computational Linguistics Lab,
Department of Computer Science and Engineering,
University of Texas at Arlington, USA
wxp7177@mavs.uta.edu, kenny.zhu@uta.edu

Abstract

In this paper, we showcase a simple yet effective framework that iteratively utilizes the capacity of the pre-trained sound separation models without additional training. The framework gains the upper hand in terms of performance over the one-step inference method, generally adopted by the universal sound separation task. An online system provides an interactive demonstration of our framework, allowing users to experience the performance gain from the multi-step inference process. We also provide a superior result of the proposed framework than the one-step inference method on standard benchmarks like speech enhancement and dog vocal separation, supporting the potential of the multi-step inference process, which will ultimately enrich the research community of natural language processing.

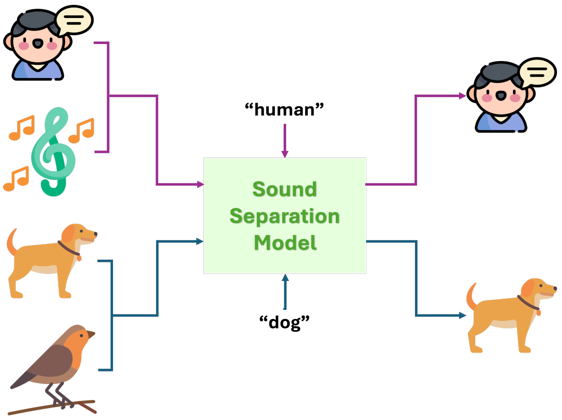


Figure 1: Overview of the sound source separation framework. The purple line represents an input mixture containing human speech and music, separating the human speech. The blue line shows dog and bird calls as an input, separating the dog call.

1 Introduction

Small-big Cycle is a well-known problem-solving strategy which progressively solves smaller and sub- problems, achieving the big and complex problem. Small pieces of the problem are comparatively manageable, tangible, and solvable, and the solution of the small problems can be a guidance to solve the big problem. An important assumption here is that nested small success can lead to the bigger success. This strategy is widely used in various fields, not only in life management and product development but also in the field of Natural Language Processing (NLP), which becomes an effective method to solve its complex problems.

Chain-of-Thought (CoT) (Wei et al., 2022; Kojima et al., 2022) improves the Large Language Model (LLM) reasoning performance by inducing the model to generate intermediate steps rather than directly going to the final output. The gains, for example, are well-documented from counting the letter 'r' in *strawberry* to multi-step mathematical reasoning. Notwithstanding such potential of the multi-step inference process, Universal Sound Separation (USS) (Kavalerov et al., 2019) pretrained models (Wisdom et al., 2020; Liu et al., 2024; Shi et al., 2025), schematized in Figure 1, undergo a massive training process to separate a sound source, facilitating the one-step inference in the end.

In order to fully draw out the capacity of the pretrained sound separation models, we present a multi-step incremental inference framework for USS, crucially utilizing existed checkpoints without any additional training cost. Transfering the principle of CoT to sound separation, to be specific, rather than utilizing a single large model to separate everything in one shot, the framework iteratively re-introduces a problem with a dynamic proportion from an original mixture which applies the number of steps that a model takes so far, representing the gradually solving small problems to achieve the complex original mixture. In other words, it decomposes the original mixture into a cycle of smaller sub-problems, each within the competence of an existing pre-trained model.

This work is of interest to researchers in both the NLP and audio communities. For the NLP community, it demonstrates that chain-of-thought-style structured inference is not specific to language – it transfers productively to a challenging task. For the audio community, it provides a practical, training-free path to improving universal sound separation by combining existing models. An online demonstration system allows users to interactively experience the framework on their own audio.

We evaluate the proposed framework against the one-step inference baseline on standard benchmarks, such as Speech Enhancement (SE) (Valentini-Botinhao, 2017) and Dog Vocal Separation (DVS) (Barkopedia, 2025), measuring both separation quality and computational cost: Albeit the training cost of the multi-step inference is zero, but the inference cost is imposed by the number of steps taken. Results confirm that structured multi-step inference consistently outperforms the single-step approach across all conditions.

Our contributions are as follows:

- We propose a novel framework for USS, a training-free multi-step inference process utilizing the existing pre-trained checkpoints, inspired by an idea from Small-big Cycle, a renowned problem-solving strategy.
- We demonstrate that our framework, so called Chain-of-Inference (CoI) – the transfer of chain-of-thought principles from language to audio – yields consistent gains over one-step inference in separation quality across all 5 different tasks.
- Publicly available online demonstration system [1] which allows users to compare the one-step and multi-step inference is released in the given link at the first page of the paper, enhancing an impact to the speech and NLP research community.

2 Methodology

While LLM inherently takes each problem stage within its context window (so-called next-token prediction), sound separation models lack this contextual input structure due to its basic design. In this section, we introduce a formula to address this issue, enabling the multi-step inference process for sound separation.

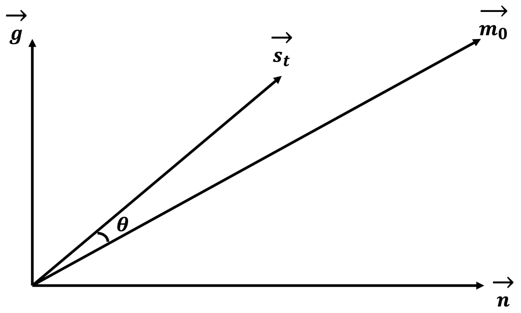


Figure 2: Vector space representation of the key variables in the multi-step inference process. The original mixture $\vec{m_0}$ is a sum of the target source $\vec{g}$ and the non-target sources $\vec{n}$. The separated source after t steps is $\vec{s_t}$, and the angle between $\vec{m_0}$ and $\vec{s_t}$ is θ.

2.1 Multi-step Inference Formulation

Equation 1 shows how the multi-step inference process is formulated;

$$m_t = r_t m_0 + (1 - r_t) s_{t-1} \tag{1}$$

where m_0 is the original mixture, s_{t-1} is the separated source from the previous step, r_t is the ratio in a range of $[0, 1]$ to control the weight between the original mixture and the separated source, and m_t is the input for the current step. The formulation enables a separation model $f(\cdot)$ to decipher m_0 in a gradual manner, as m_t is an auxiliary mixture that reintroduces a portion of the original problem at each step, preserving the previously solved part. Figure 2 illustrates the relationship among variables in a vectorial space, additionally introducing the angle θ between $\vec{m_0}$ and $\vec{s_t}$, the target source $\vec{g}$, and the non-target sources $\vec{n}$.

2.2 Ratio Schedule and Stopping Criterion

The most important variable is the ratio r_t, because it determines how much of the problem in the original mixture is restored for the current mixture input. Hinged on the vector representation in Figure 2, the separation task can be interpreted as a journey of $\vec{s_t}$ from $\vec{m_0}$ to $\vec{g}$ while increasing the angle θ. Theoretically, the range of θ is $(0, 90°)$ since $\vec{m_0}$ and $\vec{s_t}$ are never perpendicular to each other, which informatively means that the original mixture and the separated source do not have any common sound sources. This is aesthetically logical for defining the ratio as the cosine similarity between $\vec{m_0}$ and

[1] https://redgiant.uta.edu/sound_separation

Algorithm 1 Multi-step Inference

```
Require: mixture $m_0$, separation model $f$, text prompt $q$
$s_0 \leftarrow f(m_0, q)$
$r_0 \leftarrow 1$
for $t = 1, 2, \dots$ do
    $r_t \leftarrow \mathrm{cosim}(\vec{m_0}, \vec{s_{t-1}})$
    $m_t \leftarrow r_t\, m_0 + (1 - r_t)\, s_{t-1}$
    $s_t \leftarrow f(m_t, q)$
    if $r_t > r_{t-1}$ then
        break
    end if
end for
return $s_t$
```

$\vec{s_{t-1}}$:

$$
\begin{aligned}
r_t &= \mathrm{cosim}(\vec{m_0}, \vec{s_{t-1}}) \\
&= \frac{\vec{m_0} \cdot \vec{s_{t-1}}}{\|\vec{m_0}\| \, \|\vec{s_{t-1}}\|} = \cos\theta_{t-1},
\end{aligned}
\tag{2}
$$

which naturally maps r_t into the range $(0, 1)$. The less the angle θ is, the more problems are remained in the separated result so that the more problems should be reconsidered in the following step. Conversely, the more the angle θ is, the less problems are in the separated result, as well as the less problems should be reconsidered in the following step. Keeping this point of view, r_t also works as a stopping criterion for the multi-step inference process, as the process can be terminated when θ is no longer becoming larger, which means that the separation model performs its maximum capacity to solve the problem with the current mixture input.

With this multi-step inference framework, as iterates, the sound separation model repeatedly processes the mixture input, progressively making the problem mixture easier to solve regardless of the complexity of the original mixture. Algorithm 1 summarizes the overall procedure.

3 Evaluation

3.1 Setup

Datasets. We evaluate on three tasks which have clear ground truths spanning the breadth of USS. **DVS** [2] (Barkopedia, 2025) is a Barkopedia challenge dataset held at IJCAI 2025, containing dog vocalizations mixed with environmental background sounds. Among speech separation (SE) benchmarks, **VCTK-DEMAND** [3] (Valentini-Botinhao, 2017) is utilized, evaluating clean speech recovery from noisy mixtures. **MUSDB18** [4] (Rafii et al., 2017) is a standard music source separation benchmark providing four-stem mixtures (vocals, drums, bass, other). We exclude the "other" stem from evaluation, since target source extraction models are mainly trained to disentangle the text-specified source from the rest of the mixture.

Models. We apply the multi-step inference framework to two publicly available and state-of-the-art pre-trained USS models: AudioSep (Liu et al., 2024), a language-queried generalist model trained on broad open-domain audio, and SAM-Audio (Shi et al., 2025), which extends the Segment Anything paradigm to audio. Both models are used as frozen checkpoints; no fine-tuning is performed.

Implementation. Two machines were used to conduct all experiments in the paper. One is equipped with two NVIDIA RTX 4090 GPUs which have 24GB of VRAM each, and the other is installed by one NVIDIA RTX PRO 6000 Blackwell Workstation Edition with 96 GB of VRAM. The demo system described in Section 4 is also hosted on one of these machines.

3.2 Result

Table 1 reports separation quality under one-step and multi-step inference for both models across all tasks. For AudioSep, we report SDR (Vincent et al., 2006) and SI-SDR (Le Roux et al., 2019) (dB, higher is better); for SAM-Audio, we report SAM-Audio Judge (SAJ) (Wang et al., 2026) (Overall, $\in [1, 5]$, higher is better) following the evaluation protocol of SAM Audio. While AudioSep is trained and evaluated for the traditional metrics, Audio-diffusion models like SAM-Audio are not correlated with these metrics despite their being favored from the human perceptual perspective. SAJ gives an alternative way to evaluate the separation quality of SAM-Audio, more aligned with human perception. Average inference steps are shown for both models to illustrate the computational cost of multi-step inference relative to the one-step baseline.

Since both AudioSep and SAM-Audio are language-queried models, the text prompt directly determines the separation target and serves as the

[2] https://huggingface.co/datasets/ArlingtonCL2/Dog-Vocal-Separation

[3] https://datashare.ed.ac.uk/handle/10283/2791

[4] https://sigsep.github.io/datasets/musdb.html

guidance signal throughout the iterative cycle. Following the findings of the NP/VP prompting effectiveness (Shi et al., 2025), we use the subsequent prompts to SAM Audio for each task: *"dog barking"* for DVS, *"human speaking"* for SE, *"singer delivering"* for MUSDB18 vocals, *"drums playing"* for MUSDB18 drums, and *"bass slapping"* for MUSDB18 bass. All prompts are fixed across both models and both inference methods to ensure a fair comparison.

Discussion. Multi-step inference consistently improves AudioSep across all five tasks (Table 1). The largest gain is on MUSDB18 bass (+2.14 SDR, +0.49 SI-SDR, 5.34 avg. steps), a highly out-of-distribution task that also demands the most iterations. Gains on DVS (+0.15 SDR) and SE (+0.07 SDR) are smaller but consistent, reflecting tasks nearer to AudioSep's training distribution. MUSDB18 drums shows the smallest improvement (+0.01 SDR) with the fewest steps (3.24), indicating the one-step baseline is already near capacity for this stem.

For SAM-Audio, multi-step inference improves four of the five tasks, with the largest gains on MUSDB18 bass (+0.46 SAJ) and drums (+0.19 SAJ). The exception is SE, where the one-step baseline (3.620) outperforms multi-step (3.369). Such anomaly in this benchmark was constantly reported in the literatures (Yu et al., 2022; Zang et al., 2025), appearing as a smaller model gives better results than a larger model and multi-step inference gives worse results than one-step inference. Hinged on these reports, we attribute this to SAM-Audio's strong alignment with speech in its training distribution: the first-pass output is already near-optimal, and reintroducing the original mixture in subsequent steps introduces interference rather than refinement.

The average step counts reflect task difficulty across both models: AudioSep requires 3.24–5.34 steps and SAM-Audio 2.40–3.13 steps, with bass consistently demanding the most iterations. This confirms that the cosine-similarity stopping criterion in Equation 2 adapts the number of iterations to the difficulty of each mixture, incurring no unnecessary computation on easier tasks.

4 System Demonstration

The online system is built entirely in Python, using Reflex (Reflex Development Team, 2022) (v0.8.26) as the full-stack web framework that compiles the UI to the browser without requiring a separate JavaScript codebase. As shown in Figure 3, the interface offers two modes of interaction. The sample page (*Left*) presents curated audio examples with pre-computed one-step and multi-step outputs available for immediate listening. The upload page (*Right*) allows users to supply their own mixture audio file and specify a separation target via a free-form text query; the server runs both the one-step baseline and the full iterative cycle on demand and returns both outputs for side-by-side perceptual comparison.

The audio processing pipeline is built on `librosa` (McFee et al., 2015) for loading, resampling, and computing the inter-vector angle θ that drives the ratio schedule (Equation 2), and `soundfile` (Bechtold, 2025) for reading uploaded files and encoding separated outputs. Waveform and spectrogram visualizations displayed in the interface are rendered with `matplotlib` (Matplotlib Development Team, 2012), while `numpy` (NumPy Developers, 2006) underpins all signal mixing and cosine similarity computations.

The video demonstration of the system is available at https://youtu.be/JKSdAZI7SmM.

5 Related Work

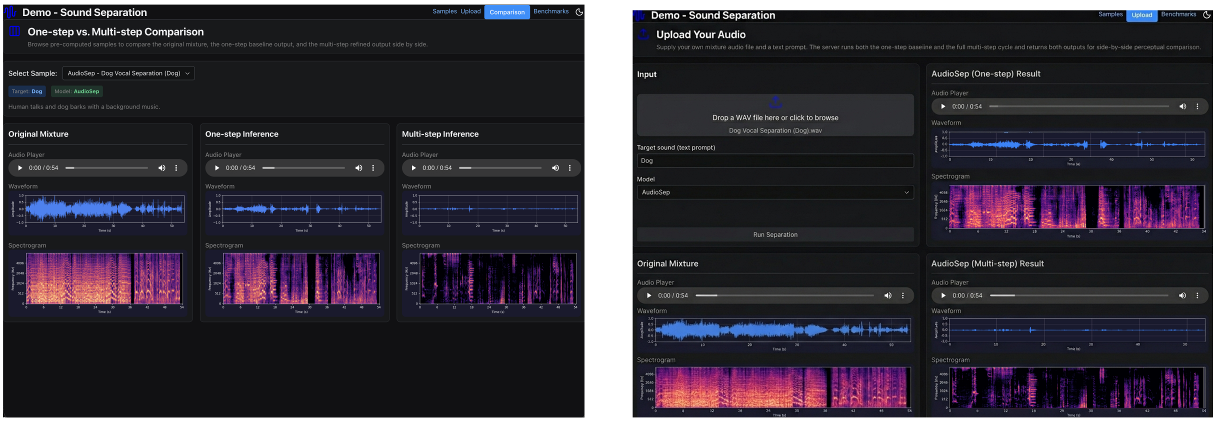
5.1 Universal Sound Separation

The goal of USS (Kavalerov et al., 2019) is to isolate an arbitrary sound source from an acoustic mixture without prior knowledge of which source categories are present. Early deep learning approaches, epitomized by Conv-TasNet (Luo and Mesgarani, 2019), established a one-step inference paradigm: a single neural network is trained end-to-end on large-scale mixed data and simultaneously estimates all sources in a single forward pass. This paradigm has remained dominant, with subsequent work focusing on scaling training data and improving network architecture rather than rethinking the inference procedure.

Two pre-trained target extraction models have emerged as the primary publicly available checkpoints for USS at scale. AudioSep (Liu et al., 2024) is trained on a broad collection of open-domain audio datasets and separates sources specified by free-form natural language queries, making it the most widely accessible generalist model. SAM-Audio (Shi et al., 2025) extends the Segment Anything paradigm to the audio domain, supporting flexible, prompt-driven separation across a simi-

Task	Method	AudioSep SDR	 SI-SDR	 Avg. Steps	SAM-Audio SAJ Ovr.	 Avg. Steps
DVS	One-step	10.075	8.713	1.00	3.742	1.00
	Multi-step	**10.226**	**8.785**	3.31	**3.764**	2.40
SE	One-step	17.320	17.237	1.00	**3.620**	1.00
	Multi-step	**17.393**	**17.294**	3.43	3.369	2.65
MUSDB18 vocals	One-step	−75.578	−77.166	1.00	3.216	1.00
	Multi-step	**−75.422**	**−76.863**	4.68	**3.342**	2.63
MUSDB18 drums	One-step	−27.760	−29.590	1.00	3.789	1.00
	Multi-step	**−27.748**	**−29.555**	3.24	**3.975**	2.80
MUSDB18 bass	One-step	−56.473	−59.226	1.00	2.944	1.00
	Multi-step	**−54.338**	**−58.735**	5.34	**3.407**	3.13

Table 1: Separation quality under one-step and multi-step inference. Bold denotes the better result per task and model. AudioSep metrics are SDR and SI-SDR (dB); SAM-Audio uses SAJ Overall, all higher is better. The negative AudioSep SDR and SI-SDR values on MUSDB18 reflect out-of-distribution evaluation (AudioSep was not trained on MUSDB18), but the consistent gains show the framework still helps.



Demo - Sound Separation

One-step vs. Multi-step Comparison

Browse pre-computed samples to compare the original mixture, the one-step baseline output, and the multi-step refined output side by side.

Select Sample: AudioSep - Dog Vocal Separation (Dog)

Target: Dog

Model: AudioSep

Human talks and dog barks with a background music.

Demo - Sound Separation

Upload Your Audio

Supply your own mixture audio file and a text prompt. The server runs both the one-step baseline and the full multi-step cycle and returns both outputs for side-by-side perceptual comparison.

Drop a WAV file here or click to browse

Dog Vocal Separation (Dog).wav

Target sound (text prompt): Dog

Model: AudioSep

Figure 3: Screenshots of the online demonstration system. (*Left*) The sample page presents prepared audio examples with one-step and multi-step outputs available for direct listening. (*Right*) Users upload their own mixture audio file and specify a separation target via a free-form text query, receiving both outputs side by side for direct perceptual comparison.

larly wide range of sound categories. Both models, however, operate under the one-step inference: given a mixture and a query, each produces a separated output in a single forward pass without recalling the original problem. Our work departs from a gap that multi-step inference like CoT provides a powerful tool for solving complex problems, treating these pre-trained checkpoints as components in a multi-step cycle rather than as standalone solvers.

5.2 Training-free Sound Separation

Recent work has begun to explore training-free approaches to sound separation. Zang et al. (2025) first demonstrated that multi-step inference can yield a "free lunch" in separation quality without any additional training. Their method, however, relies on access to the ground-truth clean sources to construct the intermediate inputs at each step, an oracle assumption that is unavailable in practical deployment scenarios where only the mixture is observed.

ZeroSep (Huang et al., 2025) proposes a genuinely training-free separation pipeline by leveraging the iterative denoising process of diffusion-based generative models. While effective within its setting, the approach is architecturally specific to diffusion models and cannot be applied to discriminative pre-trained models such as AudioSep, which constitute the dominant paradigm for large-scale USS.

Our framework addresses both limitations simultaneously. It requires no ground-truth sources and places no architectural constraint on the underlying models, making it directly applicable to the existing publicly available USS checkpoints.

6 Conclusion

We have presented a training-free multi-step inference framework for USS that transfers the chain-of-inference principle – the audio analogue of CoT

– to sound separation. Rather than demanding a single pre-trained model solve the full separation problem at once, the framework iteratively reintroduces the original mixture at a cosine-derived ratio r_t, decomposing the problem into a sequence of sub-problems each within the reach of an existing frozen checkpoint. No additional training is required.

The key technical contribution is the formulation $m_t = r_t m_0 + (1 - r_t)s_{t-1}$, where the ratio $r_t = \text{cosim}(\vec{m_0}, \vec{s_{t-1}})$ is grounded in the vector geometry of the separation problem. This provides not only a principled schedule for difficulty annealing but also a natural stopping criterion: iteration terminates when the cosine similarity between the mixture and the separated output ceases to decrease, indicating that the model has reached its capacity on the current input. We also note that a precise text prompt is increasingly consequential in this setting, as multi-step inference amplifies each model's capacity in the direction of the guidance.

Experiments across three tasks – DVS, SE, and MUSDB18 – show that multi-step inference consistently improves AudioSep across all conditions, and improves SAM-Audio on the majority of tasks. The identified failure mode – iterative degradation when the one-step output is already near-optimal – points to adaptive stopping as a productive direction for future work. Beyond automatic metrics, the framework is deployed as an interactive online demonstration system open to the research community.

Our work supports the view that chain-of-inference is not language-specific. We invite the community to explore its application to other audio understanding tasks, such as audio captioning and sound event detection, and more broadly to perception domains where pre-trained models are abundant but compositionally difficult problems remain unsolved.

References

Barkopedia. 2025. Dog vocal separation. https://huggingface.co/datasets/ArlingtonCL2/Dog-Vocal-Separation. Hugging Face Datasets.

Bastian Bechtold. 2025. python-soundfile. https://github.com/bastibe/python-soundfile. Audio library based on libsndfile, CFFI, and NumPy.

Chao Huang, Yuesheng Ma, Junxuan Huang, Susan Liang, Yunlong Tang, Jing Bi, Wenqiang Liu, Nima Mesgarani, and Chenliang Xu. 2025. Zerosep: Separate anything in audio with zero training. *arXiv preprint arXiv:2505.23625*.

Ilya Kavalerov, Scott Wisdom, Hakan Erdogan, Brian Patton, Kevin Wilson, Jonathan Le Roux, and John R Hershey. 2019. Universal sound separation. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 175–179. IEEE.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. 2019. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630. IEEE.

Xubo Liu, Qiuqiang Kong, Yan Zhao, Haohe Liu, Yi Yuan, Yuzhuo Liu, Rui Xia, Yuxuan Wang, Mark D Plumbley, and Wenwu Wang. 2024. Separate anything you describe. *IEEE Transactions on Audio, Speech and Language Processing*, 33:458–471.

Yi Luo and Nima Mesgarani. 2019. Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8):1256–1266.

Matplotlib Development Team. 2012. Matplotlib: Visualization with python. https://github.com/matplotlib/matplotlib.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. *SciPy 2015*.

NumPy Developers. 2006. Numpy: The fundamental package for array computing in python. https://github.com/numpy/numpy.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. 2017. Musdb18 - a corpus for music separation.

Reflex Development Team. 2022. Reflex: Web apps in pure python. https://github.com/reflex-dev/reflex. Accessed: 2026-02-27.

Bowen Shi, Andros Tjandra, John Hoffman, Helin Wang, Yi-Chiao Wu, Luya Gao, Julius Richter, Matt Le, Apoorv Vyas, Sanyuan Chen, and 1 others. 2025. Sam audio: Segment anything in audio. *arXiv preprint arXiv:2512.18099*.

Cassia Valentini-Botinhao. 2017. Noisy reverberant speech database for training speech enhancement algorithms and tts models. *(No Title)*.

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

Helin Wang, Bowen Shi, Andros Tjandra, John Hoffman, Yi-Chiao Wu, Apoorv Vyas, Najim Dehak, Ann Lee, and Wei-Ning Hsu. 2026. Sam audio judge: A unified multimodal framework for perceptual evaluation of audio separation. *arXiv preprint arXiv:2601.19702*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Scott Wisdom, Efthymios Tzinis, Hakan Erdogan, Ron Weiss, Kevin Wilson, and John Hershey. 2020. Unsupervised sound separation using mixture invariant training. *Advances in neural information processing systems*, 33:3846–3857.

Jianwei Yu, Yi Luo, Hangting Chen, Rongzhi Gu, and Chao Weng. 2022. High fidelity speech enhancement with band-split rnn. *arXiv preprint arXiv:2212.00406*.

Yongyi Zang, Jingyi Li, and Qiuqiang Kong. 2025. Training-free multi-step audio source separation. *arXiv preprint arXiv:2505.19534*.