MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Evaluating Inference-time Algorithms for Semantic Sound Scene Segmentation

Sridhar, Sripathi; Wichern, Gordon; Masuyama, Yoshiki; Cartwright, Mark; Le Roux, Jonathan

TR2026-164 October 29, 2026

Abstract

Spatial semantic segmentation of sound scenes (S5) involves the detection and extraction of target sound events in audio files. However, in typical detection-extraction pipelines, errors in the detection module propagate to the extraction module, degrading overall performance. In this work, we develop a multichannel detection-extraction model and evaluate inference-time algorithms to improve the class detection and target sound extraction in a multi-stage setup. We evaluate two fitness functions: binary cross entropy and mixture consistency; two relabeling strategies: naive and conditional confusion matrix relabeling; and three source estimate evaluation models: a multi-label classifier, a fine-tuned single-source classifier, and an off-the-shelf audio judge. We thoroughly evaluate the impact of these design choices on the DCASE 2025 Task 4 dataset, and validate entropy-based gating of inference-time updates on the held-out evaluation set, laying the groundwork for inference-time updates of S5 systems in real-world scenarios.

Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE) 2026

© 2026 MERL. This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Evaluating Inference-time Algorithms for Semantic Sound Scene Segmentation

Sripathi Sridhar[1,2], *Gordon Wichern*[2], *Yoshiki Masuyama*[2], *Mark Cartwright*[1], *Jonathan Le Roux*[2]

[1]New Jersey Institute of Technology, Newark NJ, USA
[2]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

***Abstract*—Spatial semantic segmentation of sound scenes (S5) involves the detection and extraction of target sound events in audio files. However, in typical detection-extraction pipelines, errors in the detection module propagate to the extraction module, degrading overall performance. In this work, we develop a multichannel detection-extraction model and evaluate inference-time algorithms to improve the class detection and target sound extraction in a multi-stage setup. We evaluate two fitness functions: binary cross entropy and mixture consistency; two relabeling strategies: naive and conditional confusion matrix relabeling; and three source estimate evaluation models: a multi-label classifier, a fine-tuned single-source classifier, and an off-the-shelf audio judge. We thoroughly evaluate the impact of these design choices on the DCASE 2025 Task 4 dataset, and validate entropy-based gating of inference-time updates on the held-out evaluation set, laying the groundwork for inference-time updates of S5 systems in real-world scenarios.**

***Index Terms*—inference-time algorithms, source separation, audio classification**

1. INTRODUCTION

Inference-time compute scaling, which focuses on improving the accuracy of responses by scaling computation at inference-time, has shown great success in LLM reasoning models [1], [2]. In audio, analogous gains have been pursued in applications such as source separation and speech enhancement, through the iterative refinement of signal representations [3] or previously extracted target signals via multiple passes through the model [4]. Some models use iterative refinement during both training and inference [4]–[6]. Spex++ used extracted speech from a previous stage to condition subsequent stages [5], and in Iterative Refined Adaptation (IRA), the speaker embedding and speech latents iteratively updated and refined each other [6]. Others such as Zang et al. [7] adapted a single-step source separation model for multi-step inference by blending the mixture with the previous stage output in different ratios, improving performance on speech and music source separation. The underlying assumption though is that the target is well-defined, such as the speaker id or instrument classes in *drums*, *bass*, *vocals*, and *other*. However, this may not hold in detection-extraction systems where the target class must be detected first before extracting the target source.

In such a framework, errors from the detection module degrade target source extraction [8], and such errors cannot be fixed through iterative refinement in the source separation module alone. This challenge naturally arises in the DCASE 2025 Task 4, titled "Spatial semantic segmentation of sound scenes (S5)". Prior work on this task favored an iterative refinement approach in a chain-of-inference style, where previously extracted labels and sources condition subsequent source extraction steps, with some using the same models in consequent steps [9], [10] while others trained multiple classifiers and separators in a two-stage setup [11]. While these methods refine the separated sources, they treat the predicted labels as reliable. Other approaches used label correction by comparing the predicted label for a source estimate against its input query [12], [13], or removed false positives and redid the source separation process when the number of predicted sources exceeded the maximum allowed [12]. Concurrent submissions to the follow-up DCASE 2026 Task 4 explore inference-time candidate selection [14] and re-tagging [15]. However, these approaches do not systematically explore more general label updates, fitness functions to score candidate labels, or label hypotheses beyond the model prediction. Evaluating these design choices could unlock a deeper understanding of inference-time algorithms for applications to general detection-extraction systems in real-world scenarios.

To this end, we propose inference-time algorithms to update predicted labels and the corresponding separated sources, and evaluate them on different detection models for the S5 task. We propose and evaluate different combinations of relabelers and fitness functions with a baseline, fine-tuned, and off-the-shelf detection model. We thoroughly analyze performance on the DCASE 2025 Task 4 test set and show improvements under certain conditions, laying the groundwork for inference-time algorithms in detection and separation systems.

2. METHODS

2.1. Task and system description

The S5 task [16] in DCASE 2025 Challenge Task 4 requires systems to detect and separate individual sound events from multi-channel spatial audio mixtures. Each input mixture contains up to three simultaneous target sound events drawn from a vocabulary of $K = 18$ classes, along with optional interference events and non-directional background noise. Systems must output isolated single-channel source estimates, each paired with a predicted class label. The primary evaluation metric is the class-aware signal-to-distortion ratio improvement (CA-SDRi) [17], which jointly rewards accurate source separation and correct class assignment by computing SDR improvement only when the predicted label matches the ground truth. Each false positive and false negative prediction is penalized with 0 SDRi.

Inspired by the baseline S5 system [17], we use a two-stage model consisting of an audio classifier followed by a target source extraction model. The classifier predicts active sound event labels from the input mixture, and these labels are used as prompts to query the extraction model, which produces one isolated source estimate per predicted label. We then apply a relabeling procedure that explores alternative label combinations and generate corresponding source estimates. Each alternative label combination is then ranked by making predictions for the corresponding source estimates and scoring them with a fitness function. We pick the alternative labels and resulting source estimates with the top rank as our final model output.

We use M2D [18], a multi-label audio classifier model trained on Audioset [19], as our detection model. We use a publicly available checkpoint released by the challenge organizers [17] that is fine-tuned on the DCASE 2025 Task 4 data. While other competitive audio classifiers exist such as BEATs [20], EAT [21], and SSLAM [22], we choose M2D since it performs competitively on AudioSet [18] while allowing us to build on the baseline system and compare directly to challenge submissions. Moreover, although the magnitude

of performance gains may vary across different classifiers, the pipeline itself remains classifier-agnostic, since it relies only on the classifier's own probabilities and confusion matrix.

2.2. Target source extraction model

We build our target source extraction model based on the task-aware unified source separation (TUSS) architecture [23]. Unlike language-queried models such as AudioSep [24] that extract one source per text query, TUSS uses context from all the provided prompts to jointly extract target sources from the input mixture, which suits the closed-set S5 task. We refer the reader to [23] for further details. We base our model on the publicly available code[1].

TUSS uses a fixed prompt vocabulary during training and inference, but can extract a flexible number of sources during inference. We use the 18 target sound classes in DCASE 2025 Task 4 [17] as the prompt vocabulary, and add separate non-target 'interference' and 'noise' prompts, where the interference prompt is repeated to match the number of interference sources in the training example. At inference, the number of interference sources is unknown, whereas background noise is always present during training and inference. We develop a multichannel TUSS variant, Mc-TUSS, to take advantage of the four-channel input audio in this task.

We train Mc-TUSS from scratch on the DCASE 2025 Task 4 training data on a single GPU using batch size two for 150 epochs, with other training parameters following TUSS [23]. We evaluate the checkpoint with the best validation loss in all inference-time experiments. Mc-TUSS matches or outperforms non-iterative variants of the top two ranked submissions [9], [10], achieving 15.1 dB CA-SDRi with ground-truth labels on the test set.

2.3. Relabelers

We explore different algorithms to update the predicted labels that prompt the target sound extraction model. In this work, 'relabeling' refers to the selection of alternate labels different from the classifier predicted labels. Initial experiments with directly sampling from the sigmoid logits of the multilabel classifier prediction did not yield promising results due to the model's tendency to confuse certain classes with each other. Per-class decision thresholds could also be optimized [25], but do not exploit the structure of inter-class confusions. Thus, we explore two relabeling strategies based on the classifier's confusion matrix. In addition to alternate labels for target events, we also relabel the number of interference sources from the set {0, 1, 2} during target+non-target relabeling.

2.3.1. Naïve Confusion Matrix (CM) relabeling: We compute the confusion matrix from single-source examples and select the top 5 confusions per class. Given a set of predicted labels, we enumerate the combinations of all possible replacements for each predicted label as our set of alternative labels. Since there are a maximum of three target events per example, the maximum cardinality of the search space is 15.

2.3.2. Conditional Confusion Matrix (CCM) relabeling: The naïve confusion matrix relabeler preserves the number of predicted sources, which limits its ability to correct errors where M2D predicts the wrong number of active classes. We propose to address this limitation by combining the audio classifier probabilities with the confusion matrix, producing corrected class probabilities over all 18 classes that can shift probability mass to classes not originally predicted.

Let the baseline M2D sigmoid output vector be $\mathbf{p} \in [0,1]^K$ over $K = 18$ classes, and define $\mathbf{C} \in \mathbb{R}^{K\times K}$ as the column-normalized confusion matrix, where $C_{ij} = P(\text{true} = i \mid \text{predicted} = j)$. We compute corrected class probabilities as

$$\tilde{\mathbf{p}} = \mathbf{C}\mathbf{p}, \tag{1}$$

clipped to $[0,1]$. Each sigmoid probability p_j is redistributed according to the confusion pattern for class j. High-confidence predictions contribute proportionally more, and frequently confused classes receive probability mass from the classes they are mistaken for. Crucially, the corrected vector $\tilde{\mathbf{p}}$ can assign nonzero probability to classes absent from the original M2D prediction, enabling the relabeler to explore combinations with different numbers of active sources. This resembles the forward correction for label noise [26], applied here at inference time to generate label hypotheses rather than to correct a training loss.

We score candidate label multihot vectors $\mathbf{u} \in \{0,1\}^K$ by their log-probability

$$\log P(\mathbf{u} \mid \tilde{\mathbf{p}}) = \sum_{i=1}^{K} \left[u_i \log \tilde{p}_i + (1-u_i)\log(1-\tilde{p}_i)\right]. \tag{2}$$

Since the task constrains the number of active sources to between one and three, the set of valid class combinations has size $\binom{18}{1} + \binom{18}{2} + \binom{18}{3} = 987$. We enumerate all 987 combinations, score each using Eq. (2), sort by descending log-probability, and retain the top N candidates for evaluation by the fitness function. We set $N = 100$ unless specified otherwise. This deterministic enumeration avoids wasteful sampling and guarantees that no candidate is evaluated more than once.

2.4. Source estimate evaluation models

We evaluate several evaluation models that generate probability vectors or ratings for the source estimates generated from relabeled prompts. The outputs of the evaluation models are used by the fitness functions to rank the alternative labels. We evaluate M2D, which is used to detect the initial labels, also as an evaluation model.

While M2D performs well on mixtures, and even on reference single-class sources, we found that M2D exhibits more inter-class confusions when evaluated on Mc-TUSS source estimates from ground-truth labels as queries. We address this by finetuning M2D with Mc-TUSS source estimates using a cosine classifier head and ArcFace loss [27]. We extract source estimates from Mc-TUSS for 100k examples from the training set, select the source estimates with CA-SDRi $\geq$ 3dB, and finetune just the classifier head while keeping the rest of the model weights fixed. Similar to the winning team of DCASE 2025 Task 4 [9], we trained for 100 epochs with learning rate 1e-4 and use validation loss to pick the best checkpoint. We refer to this model trained on single-class source estimates as M2D-SC.

We also evaluate SAM Audio Judge [28], a judge model trained on human ratings of source separation quality in speech, music, and environmental sound data. Given a text prompt, input mixture, and the separated source, SAM Audio Judge rates the separated source along dimensions of precision, recall, faithfulness, and overall quality.

2.5. Fitness functions

We evaluate three fitness functions to score the quality of source estimates conditioned on alternate labels provided by the relabeler. For target sources, we use binary cross-entropy (BCE) when the evaluation model is M2D or M2D-SC, or a specific dimension (e.g., recall) when using SAM Audio Judge. We use mixture consistency to optimize the number of non-target prompts (interference and noise).

Let $\hat{\mathbf{u}} \in \{0,1\}^K$ be a candidate relabeled multi-hot vector with corresponding class set $\hat{\mathcal{T}} = \{t : \hat{u}_t = 1\}$, and $\hat{x}_t$ the separated source

[1] https://github.com/merlresearch/unified-source-separation

estimate for a target class t. Let $\hat{\mathbf{p}}_t \in [0,1]^K$ denote the sigmoid output of the evaluation model applied to the separated source estimate $\hat{x}_t$, and $\mathbf{v} \in \{0,1\}^K$ the initial M2D multi-hot prediction from the mixture.

2.5.1. Binary Cross-Entropy (BCE): For M2D, we assume that confusion patterns remain consistent between the mixture and the source estimates:

$$\mathcal{L}_{\text{M2D}}(\hat{\mathbf{u}}) = -\sum_{t\in\hat{\mathcal{T}}}\sum_{k=1}^{K}[v_k \log \hat{p}_{tk} + (1-v_k)\log(1-\hat{p}_{tk})] \quad (3)$$

For M2D-SC, which may not retain those confusion patterns, we compute BCE against the relabeled multi-hot vector $\hat{\mathbf{u}}$ instead, replacing v_k with $\hat{u}_k$ in Eq. (3). In both cases, we select the relabeled classes that minimize $\mathcal{L}(\hat{\mathbf{u}})$.

2.5.2. Mixture Consistency: Mixture consistency is defined as the mean squared error (MSE) between the mono input mixture y and the sum of all target ($\hat{x}_t$) and non-target ($\hat{x}_\text{n}$) source estimates, $\mathcal{L}_{\text{mc}} = \text{MSE}\left(y, \sum_t \hat{x}_t + \sum_n \hat{x}_\text{n}\right)$. When jointly evaluating BCE and mixture consistency, we scale the mixture consistency loss by 100, following the weighting strategy used during joint classification-separation training in prior work [29]. We pick the sample that minimizes $\mathcal{L}_{\text{bce}} + 100\mathcal{L}_{\text{mc}}$.

2.5.3. SAM Audio Judge score: SAM Audio Judge (SAJ) [28] predicts a score r given an input mixture, a text prompt, and a source estimate. We use the relabeled classes as text prompts and compute the mean score across all estimates within a clip:

$$\bar{r}(\hat{\mathbf{u}}) = \frac{1}{|\hat{\mathcal{T}}|}\sum_{t\in\hat{\mathcal{T}}} r(y, \hat{x}_t, t) \quad (4)$$

We define a joint score $S(\hat{\mathbf{u}}) = \bar{r}(\hat{\mathbf{u}}) - \alpha\,\mathcal{L}_{\text{mc}}(\hat{\mathbf{u}})$ combining SAJ with mixture consistency, and select the candidate that improves over the baseline:

$$\hat{\mathbf{u}}^* = \begin{cases} \arg\max_{\hat{\mathbf{u}}} S(\hat{\mathbf{u}}) & \text{if } \max_{\hat{\mathbf{u}}} S(\hat{\mathbf{u}}) > S(\mathbf{v}) \\ \mathbf{v} & \text{otherwise} \end{cases} \quad (5)$$

When using SAJ alone, we set $\alpha = 0$. When combining with mixture consistency, we set $\alpha = 1000$.

2.6. Gating functions for inference-time algorithms

Inference-time updates may not always be useful when applied indiscriminately, for instance on examples with high baseline CA-SDRi. We explore algorithmic approaches to automate this decision without access to ground-truth sources or labels. Specifically, we investigate SAJ scores and M2D binary entropy.

We compute M2D binary entropy for each class k of the $K = 18$ classes: $H_k = -p_k \log p_k - (1-p_k)\log(1-p_k)$ and aggregate to clip-level features $H_{\text{max}} = \max_k(H_k)$, $H_{\text{sum}} = \sum_{k=1}^{K} H_k$, and $H_{\text{norm}} = H_{\text{sum}}/n_{active}$, where n_{active} is the number of active targets.

We sweep 500 thresholds and evaluate performance on the test set for the SAJ scores and the binary entropy features. We pick an oracle threshold with maximum combined ΔCA-SDRi over gated and ungated examples, where gated examples are the ones inference-time updates are applied to. We assess generalization by applying them unchanged to the held-out DCASE 2025 Task 4 evaluation set containing 1620 examples.

3. RESULTS AND DISCUSSION

We evaluate all combinations of fitness function, relabeler, and evaluation models on the test set. We use the relabeled classes as the class predictions for the metric. We report results on the full test set, and subsets with examples that match certain conditions, in Table 1.

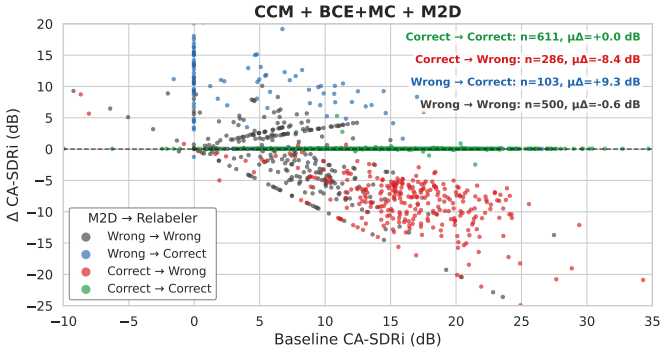

Fig. 1: Delta CA-SDRi vs. baseline CA-SDRi on the full test set for M2D with conditional CM relabeler and BCE+MC fitness function.

3.1. Performance breakdown on the full test set

Our algorithms do not improve performance when applied indiscriminately. This trend holds across relabeler, fitness, and evaluation model types. We note that the relabeler does present useful hypotheses for evaluation, as evidenced by the positive Δ_{oracle} values for all relabeler configurations, which evaluates relabeled classes by computing CA-SDRi against the ground truth. Moreover, the conditional confusion matrix relabeler, which can alter the number of sources in the relabeled classes, presents better hypotheses, as seen in larger Δ_{oracle} values. This suggests that the fitness function is the bottleneck.

To investigate this further, we qualitatively analyze the best performing model, M2D with CCM relabeling and BCE+MC fitness function. We depict separation improvements as a function of the baseline performance, CA-SDRi by using the initial M2D prompts, in Fig. 1. We notice that the model generally improves performance on examples with poor baseline performance, e.g., CA-SDRi $\leq$ 10dB, either by correcting the baseline predictions to the correct labels (blue dots, Wrong $\rightarrow$ Correct), or updating them to other wrong labels (grey dots, Wrong $\rightarrow$ Wrong) but which better isolate the true positive target sources. Based on this observation, we analyze the performance of our algorithms on specific subsets of the test set where maximum gains may be achieved.

3.2. Performance breakdown on specific subsets

In Table 1, we present results for examples in the test set with baseline CA-SDRi $\leq$ 10 dB. We also show results on oracle gap examples, which have baseline CA-SDRi $<$ 0dB but oracle CA-SDRi $>$ 5 dB, i.e., where finding the correct labels is likely to improve performance. M2D is best overall, and improves CA-SDRi by 1.56 dB on the $\leq$ 10 dB examples, which constitutes more than a third of the test set. M2D-SC matches or beats SAJ across relabeler types. M2D-SC improves performance over M2D when using the naive CM relabeler, but seems incompatible with CCM as seen by its degraded performance. This is likely due to the addition of false positives and negatives by CCM, leading to penalties and big swings in CA-SDRi.

Combining mixture consistency with other fitness functions such as BCE and SAJ recall improves performance. Mixture consistency adds a useful mixture constraint on the relabeler, and adding non-target prompts can reduce leakage into target source estimates. We also find that the choice of relabeler is conditional on the evaluation model. While the CCM relabeler significantly improves performance for M2D over the naive CM, especially in the combined BCE+MC fitness function case, it degrades performance for M2D-SC and SAJ.

However, it is clear that the proposed algorithms are useful for examples with poor baseline CA-SDRi performance. While we

Table 1: Performance of various prompt relabeling strategies, fitness functions, and evaluation models on the DCASE 2025 Task 4 test set. CM: Naïve Confusion matrix, CCM: Conditional Confusion matrix, BCE: Binary cross entropy, MC: Mixture consistency, M2D: Multilabel classifier, M2D-SC: Single-class classifier, SAJ: SAM Audio Judge. Δ refers to improvement over the baseline CA-SDRi, while Oracle is the theoretical maximum improvement from the hypotheses evaluated given an ideal fitness function, i.e., computing CA-SDRi against ground truth. Bold indicates best performance; underline indicates second best.

				All (1500 ex.) Baseline: 12.24 dB			**$\leq$10 dB (563 ex.)** Baseline: 4.79 dB			**Oracle Gap (51 ex.)** Baseline: -0.21 dB		
Relabeler	**Non-Target?**	**Fitness**	**Evaluator**	Δ	**CA-SDRi**	Δ_{oracle}	Δ	**CA-SDRi**	Δ_{oracle}	Δ	**CA-SDRi**	Δ_{oracle}
CM	✗	BCE	M2D	-4.10	8.15	1.24	-0.43	4.37	3.08	2.61	2.40	10.04
CM	✓	BCE+MC	M2D	-4.02	8.22	1.42	-0.28	4.52	3.49	2.06	1.85	10.35
CM	✗	BCE	M2D-SC	-2.68	9.56	1.24	0.47	5.26	3.08	5.34	5.13	10.04
CM	✓	BCE+MC	M2D-SC	-2.48	9.77	1.42	<u>0.84</u>	<u>5.64</u>	3.49	5.22	5.01	10.35
CM	✗	Recall	SAJ	-2.45	9.79	1.24	0.24	5.03	3.08	4.01	3.80	10.04
CM	✓	Recall+MC	SAJ	<u>-1.90</u>	<u>10.35</u>	1.42	0.67	5.47	3.49	4.52	4.31	10.35
CCM	✗	BCE	M2D	-3.85	8.40	3.48	0.78	5.57	7.71	4.83	4.62	14.84
CCM	✓	BCE+MC	M2D	**-1.16**	**11.08**	3.48	**1.56**	**6.36**	7.71	4.85	4.64	14.84
CCM	✗	BCE	M2D-SC	-4.39	7.86	3.48	0.42	5.21	7.71	**5.43**	**5.22**	14.84
CCM	✓	BCE+MC	M2D-SC	-3.78	8.46	3.48	0.42	5.22	7.71	<u>5.40</u>	<u>5.19</u>	14.84
CCM	✗	Recall	SAJ	-7.82	4.43	3.52	-1.89	2.91	7.72	2.03	1.82	14.84
CCM	✓	Recall+MC	SAJ	-7.10	5.15	3.52	-1.43	3.36	7.72	2.49	2.21	14.84

manually selected these examples by computing CA-SDRi against ground-truth sources here, we view this oracle analysis as a first step towards human-in-the-loop iterative inference, where a listener flags clips with unsatisfactory separation for inference-time updates. An alternate approach is to automatically identify potential candidates for inference-time updates to develop a more end-to-end system.

3.3. Automatically detecting candidate examples for inference-time updates

We present gating results for our best performing model so far, i.e., M2D with conditional confusion matrix relabeling on target and non-target prompts using BCE+MC as the fitness function, in Tables 2 and 3. Together with Δ_{oracle} in Table 1, which bounds the improvement given an ideal fitness function, Table 2 provides upper bounds for gating functions since their thresholds are oracle-selected on the test set itself.

On the test set, M2D entropy features provide a better gating signal. SAJ score selects examples with already strong baseline performance; while there are some performance gains on the gated examples, the relative improvement is less significant and the support set is also small. On the other hand, H_{sum} conservatively picks 24 examples with low baseline CA-SDRi of 2.85 dB and boosts performance by 1.96 dB on those examples, a relative improvement of 69%. H_{norm} is less conservative, picking 165 examples ($\approx$10% of the test set) but improving performance by a smaller margin. This suggests a tradeoff between candidate selection and performance gains which could be decided based on the application scenario.

Table 3 shows what the frozen thresholds achieve in practice on the held-out evaluation set. Similar to Fig. 1, SAJ gates select clips with baseline CA-SDRi near or above 10 dB, where updates are likely to be less useful. The entropy gates on the other hand select low-baseline CA-SDRi clips. However only H_{norm} remains useful, gating 298 clips ($\approx$18% of the eval set) with 4.92 dB baseline and improving them by +0.18 dB (+0.032 dB overall). Though small, these gains suggest that entropy-based gating could identify suitable candidates on unseen data without re-tuning gating thresholds, with headroom available towards the oracle bounds presented in Table 2.

Table 2: Gating threshold sweep results on the test set. SAM Audio Judge scores (R=Recall, P=Precision, O=Overall quality, F=Faithfulness) gate clips with score $\leq \tau$; M2D entropy features gate clips with score $> \tau$. Thresholds τ are oracle-selected by the sweep. N: number of gated clips. Δ: change in overall CA-SDRi vs. baseline (12.24 dB). Δ_{g}: mean CA-SDRi change for gated clips. BL_{g}, BL_{r}: mean baseline CA-SDRi for gated and remaining clips.

Feature	τ	N	CA-SDRi	$\Delta\uparrow$	$\Delta_{\text{g}}\uparrow$	$\text{BL}_{\text{g}}\downarrow$	$\text{BL}_{\text{r}}\uparrow$
SAJ_R	0.97	5	12.25	+0.003	+0.90	12.21	12.24
SAJ_P	1.77	57	12.28	+0.031	+0.82	9.68	12.35
SAJ_O	0.99	17	12.25	+0.004	+0.35	11.62	12.25
SAJ_F	1.01	55	12.27	+0.024	+0.67	12.04	12.25
H_{sum}	3.20	24	12.28	+0.031	**+1.96**	**2.85**	12.40
H_{max}	0.69	89	12.27	+0.023	+0.39	8.54	12.48
H_{norm}	1.61	165	**12.29**	**+0.046**	+0.42	6.22	**12.99**

Table 3: Gating on the held-out DCASE 2025 Task 4 evaluation set (1620 clips) with thresholds τ frozen from Table 2. Δ: change in overall CA-SDRi vs. baseline (8.61 dB); other columns as in Table 2.

Feature	τ	N	CA-SDRi	$\Delta\uparrow$	$\Delta_{\text{g}}\uparrow$	$\text{BL}_{\text{g}}\downarrow$	$\text{BL}_{\text{r}}\uparrow$
SAJ_R	0.97	4	8.61	0.000	-0.10	11.59	8.60
SAJ_P	1.77	81	8.50	-0.106	-2.13	10.52	8.51
SAJ_O	0.99	35	8.58	-0.030	-1.41	9.56	8.59
SAJ_F	1.01	73	8.55	-0.057	-1.27	10.69	8.51
H_{sum}	3.20	37	8.60	-0.005	-0.24	**4.62**	8.70
H_{max}	0.69	95	8.54	-0.070	-1.20	7.58	8.67
H_{norm}	1.61	298	**8.64**	**+0.032**	**+0.18**	4.92	**9.44**

4. CONCLUSION

In this work, we systematically evaluate inference-time updates for the S5 task in a classifier-separator pipeline. We study different relabelers, fitness functions, and source estimate evaluation models. We find that our best inference-time algorithms improve performance on examples where the separation model performs poorly to begin with. Applied indiscriminately, however, these updates degrade overall performance. We explore automatic gating functions that select examples likely to benefit from our algorithms. Entropy-based gating achieves a small positive gain on the held-out evaluation set without threshold re-tuning. Our oracle analyses point to higher upper bounds, which leaves headroom for stronger fitness and gating functions. Overall, our work charts the design space of these inference-time algorithms and lays a foundation for their use in detection-separation systems. A limitation of our setup is that we do not evaluate on examples with more than one source per target class or no target sources, like in DCASE 2026 Task 4 [30]. This could be addressed for instance by source-counting [31] or relaxing the label hypothesis space, which we leave to future work.

REFERENCES

[1] C. Snell, J. Lee, K. Xu, and A. Kumar, "Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning," in *Proc. ICLR*, 2025.

[2] A. Setlur, M. Y. R. Yang, C. V. Snell, J. Greer, I. Wu, V. Smith, M. Simchowitz, and A. Kumar, "e3: Learning to explore enables extrapolation of test-time compute for LLMs," in *Proc. ICLR*, 2026.

[3] D. Bralios, E. Tzinis, G. Wichern, P. Smaragdis, and J. Le Roux, "Latent iterative refinement for modular source separation," in *Proc. ICASSP*, 2023.

[4] E. Manilow, C. Hawthorne, C.-Z. A. Huang, B. Pardo, and J. Engel, "Improving source separation by explicitly modeling dependencies between sources," in *Proc. ICASSP*, 2022.

[5] M. Ge, C. Xu, L. Wang, E. S. Chng, J. Dang, and H. Li, "Multi-stage speaker extraction with utterance and frame-level reference signals," in *Proc. ICASSP*, 2021, pp. 6109–6113.

[6] C. Deng, S. Ma, Y. Sha, Y. Zhang, H. Zhang, H. Song, and F. Wang, "Robust speaker extraction network based on iterative refined adaptation," in *Proc. Interspeech*, 2021, pp. 3530–3534.

[7] Y. Zang, J. Li, and Q. Kong, "Training-free multi-step audio source separation," *arXiv preprint arXiv:2505.19534*, 2025.

[8] Y. Kwon and J.-W. Choi, "Sound separation and classification with object and semantic guidance," *arXiv preprint arXiv:2509.15899*, 2025.

[9] Y. Kwon, D. Lee, D. Kim, and J.-W. Choi, "Self-guided target sound extraction and classification through universal sound separation model and multiple clues," in *Proc. DCASE*, 2025, pp. 235–239.

[10] T. Morocutti, J. Greif, P. Primus, F. Schmid, and G. Widmer, "On temporal guidance and iterative refinement in audio source separation," in *Proc. DCASE*, 2025, pp. 225–229.

[11] X. Zhou, H. Wang, C. Li, B. Han, X. Zheng, and Y. Qian, "SJTU-AUDIOCC system for DCASE 2025 Challenge Task 4: Spatial semantic segmentation of sound scenes," DCASE Challenge, Tech. Rep., 2025.

[12] J. Park, J. Lee, D.-H. Lim, H. K. Kim, H. Geum, and J. E. Lim, "Performance improvement of spatial semantic segmentation with enriched audio features and agent-based error correction for DCASE 2025 Challenge Task 4," *arXiv preprint arXiv:2506.21174*, 2025.

[13] V. Stergioulis and G. Potamianos, "Redux: An iterative strategy for semantic source separation," DCASE2025 Challenge, Tech. Rep., 2025.

[14] Y. You, "DCASE 2026 task 4 submission: Duplicate-label-aware source selection and TUSS overlay grafting," DCASE2026 Challenge, Tech. Rep., 2026.

[15] S. Jeong and S.-E. Kim, "Tagging-driven inference refinement and dual-separator selection for spatial semantic segmentation of sound scenes," DCASE2026 Challenge, Tech. Rep., 2026.

[16] M. Yasuda, B. T. Nguyen, N. Harada, R. Serizel, M. Mishra, M. Delcroix, S. Araki, D. Takeuchi, D. Niizumi, Y. Ohishi, T. Nakatani, T. Kawamura, and N. Ono, "Description and discussion on DCASE 2025 Challenge Task 4: Spatial semantic segmentation of sound scenes," *arXiv preprint arXiv:2506.10676*, 2025.

[17] B. T. Nguyen, M. Yasuda, D. Takeuchi, D. Niizumi, Y. Ohishi, and N. Harada, "Baseline systems and evaluation metrics for spatial semantic segmentation of sound scenes," *arXiv preprint arXiv:2503.22088*, 2025.

[18] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked Modeling Duo: Towards a universal audio pre-training framework," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 2391–2406, 2024.

[19] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "AudioSet: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, 2017, pp. 776–780.

[20] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. ICML*, 2023.

[21] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: Self-supervised pre-training with efficient audio transformer," in *Proc. IJCAI*, 2024.

[22] T. Alex, S. Atito, A. Mustafa, M. Awais, and P. J. B. Jackson, "SSLAM: Enhancing self-supervised models with audio mixtures for polyphonic soundscapes," in *Proc. ICLR*, 2025.

[23] K. Saijo, J. Ebbers, F. G. Germain, G. Wichern, and J. Le Roux, "Task-aware unified source separation," in *Proc. ICASSP*, 2025.

[24] X. Liu, Q. Kong, Y. Zhao, H. Liu, Y. Yuan, Y. Liu, R. Xia, Y. Wang, M. D. Plumbley, and W. Wang, "Separate anything you describe," *arXiv preprint arXiv:2308.05037*, 2023.

[25] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, "Optimal thresholding of classifiers to maximize F1 measure," in *Proc. ECML PKDD*, 2014.

[26] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proc. CVPR*, 2017, pp. 1944–1952.

[27] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019, pp. 4690–4699.

[28] H. Wang, B. Shi, A. Tjandra, J. Hoffman, Y.-C. Wu, A. Vyas, N. Dehak, A. Lee, and W.-N. Hsu, "SAM Audio Judge: A unified multimodal framework for perceptual evaluation of audio separation," in *Proc. ICML*, 2026.

[29] F. Pishdadian, G. Wichern, and J. Le Roux, "Learning to separate sounds from weakly labeled scenes," in *Proc. ICASSP*, 2020, pp. 91–95.

[30] M. Yasuda, B. T. Nguyen, N. Harada, R. Serizel, M. Mishra, M. Delcroix, C. Hernandez-Olivan, S. Araki, D. Takeuchi, T. Nakatani *et al.*, "Description and discussion on DCASE 2026 challenge task 4: Spatial semantic segmentation of sound scenes," *arXiv preprint arXiv:2604.00776*, 2026.

[31] K. Saijo, Y. Masuyama, C. Boeddeker, G. Wichern, J. Richter, T. Edo, and J. Le Roux, "The MERL systems for DCASE 2026 challenge task 4," DCASE2026 Challenge, Tech. Rep., 2026.