When Matching Is Not Selection: Learning Criterion-Dependent Similarity in Speech Representations

Anonymous authors
Paper under double-blind review

Abstract

Speech carries multiple attributes such as speaker identity, emotion, and linguistic content, and different retrieval tasks require judging similarity based on different attributes. Existing instruction-conditioned methods can already condition speech representations or retrieval behavior on different criteria, but correct matching on the target attribute does not mean that candidate ranking is still determined by the current criterion when attributes compete. This paper distinguishes attribute matching from attribute selection, and reveals a consistent matching–selection gap across multiple instruction-conditioned representation models through attribute competition and criterion reversal evaluations. Conventional criterion-wise training does not explicitly organize the joint relevance status of the same candidate across criteria. We propose Joint-State Relevance Learning (JSRL), which structures training supervision using the four joint relevance states of a candidate under a criterion pair and performs criterion-wise ranking over the resulting relevance partitions, while allowing all criteria to share a single candidate index. On three corpora not used in training, JSRL outperforms the baselines on attribute-conflict ranking, criterion reversal, and multi-candidate retrieval. The learned criterion-dependent ranking behaviors can also be recombined for the speaker–content pair, which is not used in criterion-pair training, and transfer to the speaker/style instruction retrieval tasks constructed in INSPIRE Synthetic. These results show that organizing supervision around joint relevance states can improve how reliably speech similarity follows the current criterion.

1 Introduction

Speech simultaneously carries multiple types of information such as speaker identity, emotional state, and linguistic content (Chen et al., 2022; Li et al., 2026a). Two utterances can come from the same speaker yet express different emotions, or convey the same content yet come from different speakers. Therefore, similarity between speech utterances is not fixed but depends on the criterion that the current task focuses on. Traditional speech representations typically use criterion-agnostic similarity in a fixed representation space, making it difficult for the similarity of the same speech pair to change with the task criterion (Desplanques et al., 2020; Ma et al., 2024). To accommodate such criterion-dependent similarity, speech representations need to retain multiple types of information while emphasizing information relevant to the current criterion, so that the similarity relationships among the same set of utterances change accordingly. Recently, ALM2Vec (Lu et al., 2026), WAVE (Tang et al., 2026), and TP-CLAP (Li et al., 2026b) have begun learning instruction-conditioned audio representations, while INSPIRE (Li & Lee, 2026) further expresses retrieval criteria through natural language instructions and dynamically defines relevance in speech retrieval. These advances show that instructions can already change speech representations or retrieval behavior. However, the fact that representations change with natural language instructions does not mean that the criterion expressed by the instruction truly determines the similarity relationships between speech utterances.

Speech retrieval provides a direct way to examine this issue. Given the same query utterance and the same set of candidates, different retrieval criteria should lead to different candidate rankings. We focus on three types of attributes—speaker identity (S), emotional state (E), and linguistic content

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

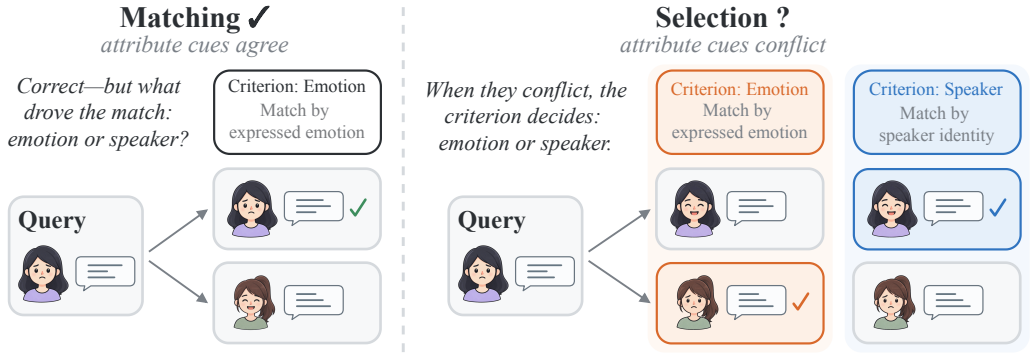

Figure 1: **Attribute matching and attribute selection.** When attribute cues agree, correct retrieval does not reveal which attribute drives the ranking; when cues conflict, switching the retrieval criterion should switch the preferred candidate.

(C)—which coexist in the same utterance. Each retrieval criterion specifies which attribute to use for judging relevance. For example, in emotion-based retrieval, if a candidate sharing the same emotion as the query also happens to come from the same speaker, a model that primarily relies on speaker information may also produce the correct result. In this case, the correct retrieval result does not indicate that the model relied on emotion information. We therefore distinguish attribute matching from attribute selection. The former requires the model to identify matching relationships on the target attribute; the latter further requires that candidate ranking follows the current criterion when multiple attributes provide competing evidence, as shown in Figure 1. We refer to the gap between attribute matching ability and attribute selection ability as the matching–selection gap.

We quantify attribute selection ability through attribute competition evaluation. For the same target attribute, we systematically vary the ranking evidence provided by non-target attributes, from being aligned with the target (Aligned), to providing no systematic directional advantage (Balanced), to directly competing with the target (Conflict). The criterion reversal evaluation fixes the same query and the same candidate pair, changes only the retrieval criterion, and tests whether the candidate order can correctly reverse. Experiments on three corpora show that multiple instruction-conditioned representation models can reliably match the target attribute under the Aligned condition, but when non-target attributes provide competing evidence, ranking accuracy drops substantially, and models often cannot satisfy both ranking directions after a criterion switch. This consistent matching–selection gap indicates that matching performance under a single criterion does not adequately reflect whether candidate ranking truly follows the current criterion.

This gap motivates organizing training supervision around how candidate relevance changes across criteria. Under criterion-wise training, relevance is supervised for the current criterion, while the joint relevance status of the same candidate across a criterion pair is not explicitly represented as a training structure. Because multiple attributes coexist in each speech candidate, a criterion pair naturally induces four joint states: relevant under both criteria, irrelevant under both, or relevant under only one of the two criteria. Based on this structure, we propose Joint-State Relevance Learning (JSRL), which constructs training units from these four states and performs criterion-wise set ranking over the relevance partition induced by the current criterion. These training units expose both candidate relationships that should persist and those that should switch when the criterion changes, while retaining a shared candidate index. Experiments show that JSRL outperforms the baselines on attribute conflict, criterion reversal, and multi-candidate retrieval. The learned criterion-dependent ranking behaviors can also be recombined for the S–C pair, which is not used in criterion-pair training, and transfer to the speaker/style instruction retrieval tasks constructed in INSPIRE Synthetic.

Our contributions are as follows:

(1) We distinguish attribute matching from attribute selection in selective speech representations, and reveal a consistent matching–selection gap across multiple instruction-conditioned representation models through attribute competition and criterion reversal evaluations.

(2) We propose Joint-State Relevance Learning (JSRL), a joint-state supervision framework that organizes candidates by four relevance states under a criterion pair, so that criterion-wise ranking covers both persistent and switching relationships while retaining a shared candidate index.

(3) Experiments across three corpora not used in training show that JSRL improves attribute-conflict ranking, criterion reversal, and multi-candidate retrieval. The learned criterion-dependent ranking behaviors can also be recombined for the untrained S–C criterion pair, and transfer to independently constructed instruction retrieval tasks in INSPIRE Synthetic.

2 Related Work

Traditional speech and audio representations produce a fixed representation for each input and support different downstream tasks with the same representation space (Baevski et al., 2020; Hsu et al., 2021; Chen et al., 2022; Wu et al., 2023). Recent work has begun specifying the current task through natural language, allowing the same audio to be represented differently depending on the task. WAVE (Tang et al., 2026) and ALM2Vec (Lu et al., 2026) learn prompt-aware and instruction-aware embeddings respectively, and TP-CLAP (Li et al., 2026b) incorporates text prompts into audio representations via cross-attention and fine-tunes with attribute supervision for attribute-oriented audio retrieval in music. INSPIRE (Li & Lee, 2026) extends this direction with an instruction-aware speech retrieval benchmark that dynamically defines relevance based on semantic content, speaker, speaking style, environmental sounds, and their combinations through natural language. AudioLens (Huang et al., 2026) proposes multi-perspective speech clustering, enabling the same set of speech utterances to form different valid partitions based on different natural language perspectives. These advances show that models can already adjust the information they attend to based on task requirements described in natural language, and change how speech collections are organized accordingly. However, whether the current criterion truly determines the similarity relationships between candidates and the final ranking when multiple coexisting attributes support different candidates has not been systematically characterized. We examine this issue through attribute competition and cross-criterion reversal, where the target and competing attributes support different candidates, and test whether the same candidate set is correctly reordered according to the retrieval criterion.

Research on adjusting similarity relationships according to task requirements also appears in vision and text retrieval. In vision, CSN (Veit et al., 2017) learns different similarity concepts through conditioned subspaces, and GeneCIS (Vaze et al., 2023) evaluates whether models can adjust similarity relationships between images based on text conditions. In text retrieval, recent work has begun directly examining relevance changes caused by instruction changes: FollowIR (Weller et al., 2025a) uses p-MRR to measure whether document rankings change in the correct direction as the relevance definition changes, and Promptriever (Weller et al., 2025b) trains models to distinguish topical relevance from instruction constraints through instruction negatives. Dual-View Training (Zeng et al., 2026) constructs complementary instructions for positives and instruction negatives, making them swap relevance roles under the same query, thereby directly supervising instruction-dependent relevance reversal. Multiple attributes in speech coexist in each candidate, so the same query–candidate pair naturally forms four joint relevance states under a pair of criteria. JSRL uses this four-state structure to organize training supervision, so that the same criterion-pair training framework covers both relevance persistence and relevance switching.

3 Method

JSRL learns criterion-dependent similarity between speech queries and candidates on a shared candidate index. As shown in Figure 2, each candidate is first assigned to one of four joint states according to its binary relevance under a pair of criteria. The same joint-state candidate set then induces different relevance partitions depending on the current criterion, from which criterion-wise ranking supervision is derived.

3.1 Criterion-Conditioned Retrieval with a Shared Candidate Index

Given a query utterance q, a candidate utterance d, and a retrieval criterion $c \in \{\mathrm{S}, \mathrm{E}, \mathrm{C}\}$ (speaker, emotion, linguistic content), let $y_c(q, d) \in \{0, 1\}$ denote whether d is relevant to q under criterion

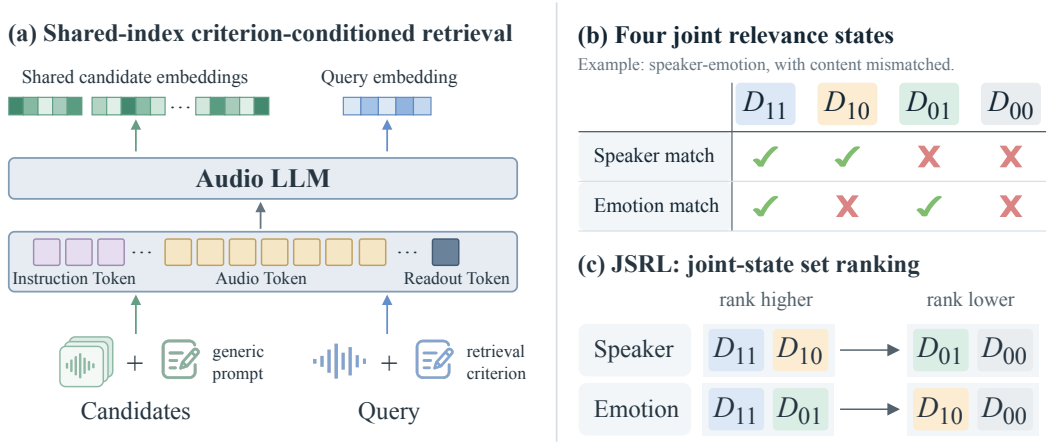

Figure 2: **Joint-State Relevance Learning (JSRL).** (a) Criterion-conditioned query encoding with a shared candidate index. (b) Four joint relevance states under a criterion pair. (c) The same joint-state candidate set induces criterion-wise ranking supervision that covers both relevance persistence and switching. The example shows speaker–emotion with content mismatched.

c. The query and candidate share the retrieval encoding function g_θ. The query side takes as input a natural language instruction z_c expressing criterion c, while the candidate side uniformly uses a fixed generic prompt z_0. The normalized representations and retrieval score are

$$\mathbf{u}_c(q) = \mathrm{norm}(g_\theta(q, z_c)), \qquad \mathbf{v}(d) = \mathrm{norm}(g_\theta(d, z_0)), \\ s_c(q, d) = \mathbf{u}_c(q)^\top \mathbf{v}(d). \tag{1}$$

where $\mathrm{norm}(\mathbf{h}) = \mathbf{h}/\|\mathbf{h}\|_2$. Since the candidate representation is independent of the criterion, switching criteria only requires re-encoding the query, and all criteria share the same candidate index.

3.2 Joint-State Relevance Learning

Joint relevance states. JSRL organizes training supervision with criterion pairs $e = (f, g)$, where each retrieval is still defined by one of the two criteria. For a query q, the candidate set $\mathcal{D}(q; e)$ for a training unit is sampled from the candidate pool. The same candidate has binary relevance under each of the two criteria, resulting in four joint states:

$$\mathcal{D}_{ab}(q; e) = \{d \in \mathcal{D}(q; e) : y_f(q, d) = a,\ y_g(q, d) = b\}, \quad a, b \in \{0, 1\}. \tag{2}$$

These four states cover all relevance combinations under the two criteria and define the relevant and irrelevant sets for each criterion:

$$\mathcal{P}_f = \mathcal{D}_{11} \cup \mathcal{D}_{10}, \qquad \mathcal{N}_f = \mathcal{D}_{01} \cup \mathcal{D}_{00}, \\ \mathcal{P}_g = \mathcal{D}_{11} \cup \mathcal{D}_{01}, \qquad \mathcal{N}_g = \mathcal{D}_{10} \cup \mathcal{D}_{00}. \tag{3}$$

The same candidate set thus forms different positive–negative partitions under the two criteria. $\mathcal{D}_{11}$ and $\mathcal{D}_{00}$ retain their positive and negative roles under both criteria, while $\mathcal{D}_{10}$ and $\mathcal{D}_{01}$ swap supervision roles. In the S/E/C setting used in this paper, the third attribute is required not to match the query when constructing training units, so that it does not provide additional alignment cues; the complete attribute correspondence for all three criterion pairs is given in Appendix Table 3.

Criterion-wise optimization with joint-state supervision. Given the current criterion c, Eq. (3) yields the relevant set $\mathcal{P}_c$ and the irrelevant set $\mathcal{N}_c$. We apply a standard softmax ranking loss to this criterion-specific partition:

$$\ell_{\text{set}}^{c|e}(q) = \frac{1}{|\mathcal{P}_c|} \sum_{d^+ \in \mathcal{P}_c} \log \left[1 + \sum_{d^- \in \mathcal{N}_c} \exp \left(\frac{s_c(q, d^-) - s_c(q, d^+)}{\tau} \right) \right], \tag{4}$$

where τ is the temperature. The ranking loss itself is criterion-wise; the joint-state structure enters through the construction of the candidate set in Eq. (2) and the criterion-dependent relevance partitions in Eq. (3). During training, f and g are selected as the current criterion in a balanced manner within each criterion pair, so that both criterion directions receive balanced supervision. Let $\mathcal{L}_{\text{set}}^{c|e}$ denote the mean of the ranking loss over queries under the current criterion. To avoid extra weighting of a criterion that appears in multiple criterion pairs, the aggregation operator $\mathcal{A}$ first averages across criterion pairs within the same criterion, then equally weights the three criteria of speaker, emotion, and content, giving

$$\mathcal{L}_{\text{JSRL}} = \mathcal{A}\left(\mathcal{L}_{\text{set}}^{c|e}\right). \tag{5}$$

3.3 Training Procedure

We organize training in three stages that progressively introduce single-criterion matching, attribute competition, and joint-state supervision.

Stage 1 (Attribute matching). Single-criterion retrieval is learned separately for speaker, emotion, and content, establishing basic matching ability for each attribute.

Stage 2 (Attribute competition). Candidates that match a non-target attribute but violate the current criterion are used as hard negatives, training the model to rank by the current criterion under competing cues.

Stage 3 (Joint-state learning). Single-criterion batches and joint-state batches alternate at a 1:1 ratio. The former maintains single-criterion retrieval ability. The latter constructs candidates according to the four joint relevance states in Eq. (2) and applies criterion-wise set ranking over the partition induced by the current criterion, thereby exposing both persistent and switching candidate relationships. Stages 2 and 3 train only on the S–E and E–C criterion pairs; S–C is not used in criterion-pair training and is reserved for evaluation as an untrained criterion pair in Section 4.5.

4 Experiments

4.1 Experimental Setup

Model and training. We build g_θ on MOSS-Audio-4B-Instruct (Yang et al., 2026), projecting the final-layer hidden state at the last assistant-prefix position to 512 dimensions through a shared linear layer. Training updates only the attention-layer LoRA (Hu et al., 2022) parameters and the projection layer. Following Section 3.3, Stage 1 uses ESD (Zhou et al., 2021), EARS (Richter et al., 2024), MSP-Podcast (Busso et al., 2026), and LibriTTS-R (Koizumi et al., 2023); Stages 2–3 use ESD and MSP-Podcast. Stage 1 and Stage 2 denote their respective final checkpoints, and JSRL denotes the final Stage 3 model. Training configurations are given in Appendix B.

Evaluation data. Attribute competition evaluation, criterion reversal evaluation, and multi-candidate retrieval are all conducted on CREMA-D (Cao et al., 2014), MSP-IMPROV (Busso et al., 2017), and RAVDESS (Livingstone & Russo, 2018), none of which are used in training. Candidates are partitioned according to the four joint states defined in Section 3.2. Consistent with training, the attribute competition evaluation requires that the third attribute does not match the query, retains only queries for which all four states are non-empty, and uses all candidates in each state. Content relevance in our evaluation is defined by matching sentence identities, capturing discrete matches in linguistic content. The evaluation protocol and panel statistics are given in Appendix A.

Baselines and retrieval interfaces. We compare WavLM-Large (Chen et al., 2022), LAION-CLAP (Wu et al., 2023), TP-CLAP (Li et al., 2026b), WAVE (Tang et al., 2026), ALM2Vec (Lu

et al., 2026) pre-trained (PT) and retrieval fine-tuned (FT) checkpoints, and Frozen MOSS-Audio without our retrieval training. We report three retrieval interfaces where supported: *Static* uses criterion-independent representations; *Fixed* conditions only the query and shares one candidate index across criteria; *Dynamic* conditions both query and candidates, requiring a separate index per criterion. Model-specific interfaces and representation extraction details are given in Appendix C.

Evaluation metrics. The attribute competition evaluation measures pairwise accuracy, assessing whether positives rank above negatives under the current criterion. In Aligned, the other attribute supports the same ranking direction as the current criterion; in Balanced, the other attribute's match status relative to the query is held fixed, and then positives and negatives under the current criterion are compared; in Conflict, the two attributes support opposite ranking directions. SwitchExact fixes the same query and the same $\mathcal{D}_{10}/\mathcal{D}_{01}$ candidate pair: success is counted only when both criteria yield the correct and opposite ranking. Results are aggregated with equal weight over queries and the three corpora; Aligned, Balanced, and Conflict are also averaged equally over the two criterion directions. Full definitions and aggregation rules are given in Appendix A.3.

4.2 Matching–Selection Gap

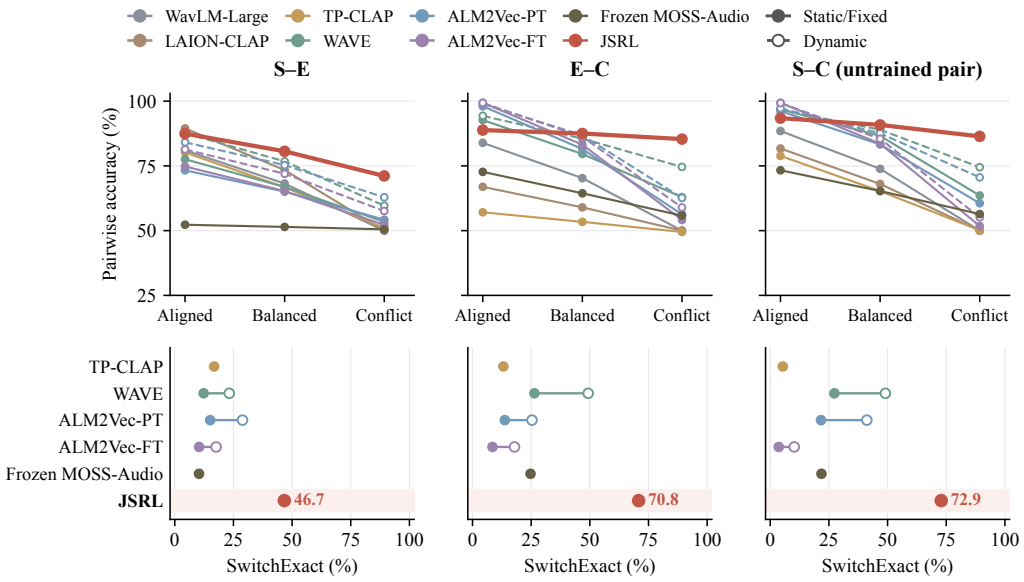

Figure 3: **Matching–Selection Gap under attribute competition.** Columns show S–E, E–C, and the untrained S–C pair. The top row reports pairwise accuracy from Aligned to Conflict; the bottom row reports SwitchExact on the same conflicting candidate pairs. Results are equal-weight macro-averages over three corpora.

Figure 3 shows that instruction-conditioned models still exhibit a clear matching–selection gap. As non-target attributes shift from aligned cues to competing cues, ranking accuracy decreases continuously from Aligned to Balanced to Conflict, and SwitchExact further shows that the same conflicting candidate pair often cannot satisfy both ranking directions after a criterion switch. For example, the strongest baseline on E–C, WAVE (Dynamic), achieves 94.34% under the Aligned condition but drops to 74.61% under Conflict, with SwitchExact at only 49.31%. This shows that being able to identify the attribute corresponding to the current criterion does not mean that this criterion can dominate candidate ranking when competing attributes are present.

Candidate-side conditioning can reduce this gap. In the macro-averaged results across the three criterion pairs, the Dynamic interfaces of WAVE and ALM2Vec outperform their Fixed counterparts on Balanced, Conflict, and SwitchExact, indicating that re-encoding candidates by criterion can strengthen the influence of the criterion on similarity relationships. However, even when both the query and candidate sides are conditioned, the drop from Aligned to Conflict remains substantial, and SwitchExact for all Dynamic baselines is below 50% across the three criterion pairs. Ap-

pendix F.1 further decomposes these failures: for many conflicting candidate pairs, the score difference has already changed in the correct direction after a criterion switch, yet it is still not enough to reverse the candidate order. In other words, the model can respond to the instruction, but the current criterion may still not determine the final ranking.

With criterion conditioning on the query side and a shared candidate index, JSRL achieves the highest Conflict and SwitchExact scores across all three criterion pairs, surpassing the Dynamic baselines that re-encode candidates for each criterion.

4.3 Multi-Candidate Retrieval Quality

To test whether the improvement in attribute selection translates into overall ranking gains on candidate sets, we construct 16-candidate pools for each of the three corpora not used in training, containing multiple relevant candidates as well as distractors from different joint relevance states. Unlike the attribute competition evaluation, this evaluation does not exclude candidates that match the query on the third attribute, so models must rank the full candidate set under more complex co-occurring attribute cues. All models use the same candidate sampling, and mAP, NDCG@5 (Järvelin & Kekäläinen, 2002), and Recall@5 are used to measure overall ranking, top-list ranking, and positive recall, respectively.

Table 1 shows that JSRL achieves 75.76% mAP, 73.47% NDCG@5, and 74.30% Recall@5, the highest on all three metrics. JSRL improves over the strongest Dynamic baseline, WAVE, by 4.11, 4.46, and 4.62 percentage points on mAP, NDCG@5, and Recall@5, respectively; its mAP is 11.20 percentage points higher than the strongest Fixed baseline, WAVE. These results show that the improvements in Conflict and SwitchExact translate into overall ranking gains when multiple relevant candidates and heterogeneous distractors coexist.

Table 1: **Multi-candidate retrieval quality (%).** All models use identical candidate pools. Results are averaged over ten samplings and queries, macro-averaged over three corpora, and aggregated with equal weight over the speaker, emotion, and content criteria. Bold indicates the best result in each column.

Model	Interface	mAP ↑	NDCG@5 ↑	Recall@5 ↑
WavLM-Large	Static	55.92	52.06	51.93
LAION-CLAP	Static	51.58	46.89	47.21
TP-CLAP	Fixed	45.94	39.45	41.92
WAVE	Fixed	64.56	60.98	61.74
WAVE	Dynamic	71.65	69.01	69.68
ALM2Vec-PT	Fixed	63.53	59.84	60.61
ALM2Vec-PT	Dynamic	69.29	66.34	67.37
ALM2Vec-FT	Fixed	63.30	59.34	60.07
ALM2Vec-FT	Dynamic	66.54	63.07	63.94
Frozen MOSS-Audio	Fixed	48.51	42.28	43.72
JSRL	Fixed	**75.76**	**73.47**	**74.30**

4.4 Effect of Relevance Switching and Persistence

To separate the effects of relevance switching and persistence supervision, we start from the same Stage 2 checkpoint and compare, under the same data schedule and training budget, Stage 2 continuation, JSRL w/o persistence (which only supervises $\mathcal{D}_{10}/\mathcal{D}_{01}$ switching), and full JSRL. SwitchExact measures criterion reversal, Persistence is the average of the persistently-relevant and persistently-irrelevant metrics P_{11} and P_{00}, and multi-candidate mAP measures overall retrieval quality. The definition of Persistence is given in Appendix A.4. The difference between JSRL w/o persistence and JSRL is whether candidate comparisons involving persistent states are supervised; these comparisons affect both Persistence and multi-candidate retrieval. The correspondence between supervision and metrics is detailed in Appendix A.4.

As shown in Figure 4, JSRL w/o persistence raises SwitchExact from 54.73% (Stage 2 continuation) to 58.79%, showing that supervising $\mathcal{D}_{10}/\mathcal{D}_{01}$ effectively strengthens criterion reversal. At the same

time, Persistence drops from 75.33% to 74.46%, and multi-candidate mAP only increases from 72.26% to 72.76%. This shows that supervising relevance switching alone can enhance ranking reversal but does not cover the candidate relationships that should still hold after a criterion switch.

After adding persistence supervision, JSRL achieves SwitchExact comparable to JSRL w/o persistence, raises Persistence to 76.66%—the only one of the three branches whose Persistence score exceeds that of the initial checkpoint—and improves multi-candidate mAP by 3.00 percentage points over JSRL w/o persistence, reaching 75.76%. Full joint-state supervision maintains criterion reversal ability while improving the ranking of persistent-state candidates and multi-candidate retrieval quality.

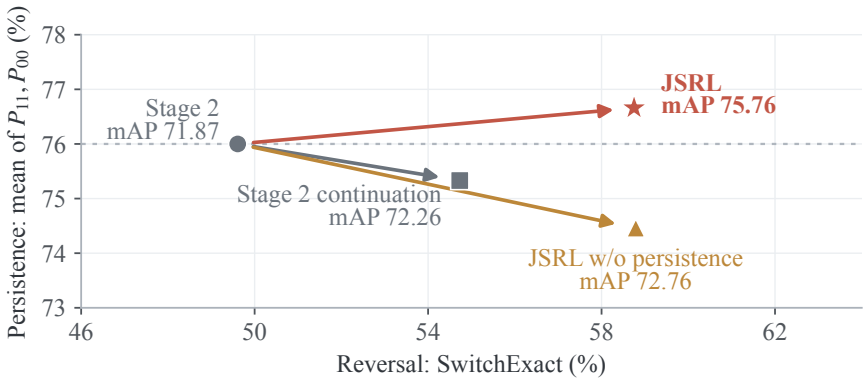

Figure 4: **Effects of relevance switching and persistence supervision.** All variants start from the same Stage 2 checkpoint. Axes show SwitchExact and Persistence averaged over S–E and E–C; labels report S/E/C mAP. The dashed line marks Stage 2 Persistence.

4.5 Ranking Recombination on the Untrained Criterion Pair

S–C tests whether two trained criteria can be combined in an untrained pairing. Both the speaker and content criteria appear during training, but the conflict between them has never been explicitly supervised. As shown in Figure 5(a), the S–C SwitchExact increases from 39.80% at Stage 1 to 59.67% at Stage 2 and 72.88% at JSRL, although criterion-pair training in Stages 2 and 3 uses only S–E and E–C. This shows that ranking behaviors learned on other criterion pairs can transfer to the untrained S–C pairing.

To examine whether the learned criterion-dependent ranking behaviors can be used for a new criterion pair, we compare, on the same queries, rankings under the same criterion across trained and untrained criterion pairs. As shown in Figure 5(b), the accuracy gap for JSRL's speaker criterion between S–E and S–C, and for the content criterion between E–C and S–C, is no more than 0.7 percentage points in both cases. This shows that the learned speaker and content ranking behaviors can be recombined for the untrained S–C criterion pair.

4.6 Transfer to an External Retrieval Protocol

To test whether the learned criterion-dependent ranking can generalize beyond this paper's evaluation protocol, we evaluate on two single-attribute instruction retrieval tasks in INSPIRE Synthetic (Li & Lee, 2026): same-speaker and same-speaking-style. The former directly corresponds to this paper's speaker criterion. The latter defines relevance by expressive speaking styles such as happiness, anger, and sadness, which is semantically close to this paper's emotion criterion. INSPIRE's data, candidate pool, and natural language instructions are all independently constructed, making this a cross-data and cross-protocol external transfer test.

We follow INSPIRE's official embedding evaluation setup: instruction-conditioned methods add the official natural language instruction only on the query side using the Fixed interface; static representations use the Static interface. In both tasks, each query has only 2 positives, while the valid candidate pools contain approximately 4,300 and 3,600 utterances respectively, constituting

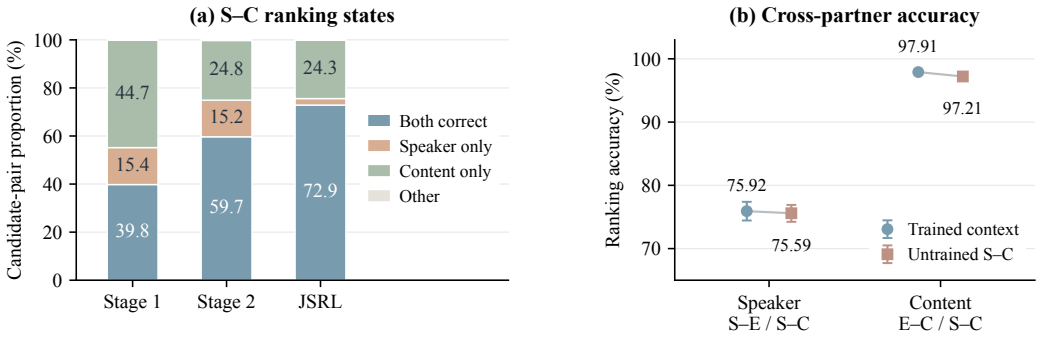

Figure 5: **Ranking recombination on the untrained S–C criterion pair.** (a) Ranking outcomes for the same S–C conflicting candidate pairs across Stage 1, Stage 2, and JSRL. (b) JSRL single-direction accuracy in trained criterion-pair contexts versus S–C on shared queries; error bars show query-speaker-clustered 95% confidence intervals. Results are macro-averaged over three corpora.

a highly sparse retrieval setting. We also include the Qwen3-Embedding-8B (Zhang et al., 2025) cascade, the strongest overall performer reported for INSPIRE Synthetic, as a reference, and report the expected Recall under uniform random ranking.

Table 2 shows that JSRL achieves 11.75% and 10.75% Recall@10 on same-speaker and same-speaking-style retrieval, respectively, compared with the best non-JSRL results of 1.00% and 4.75% on the two tasks. This shows that the criterion-dependent ranking learned during our training can transfer to external retrieval tasks with independently constructed data, candidate pools, and instructions, providing additional evidence for transfer in the instruction-aware retrieval setting.

Table 2: **Recall@10 (%) on two single-attribute tasks from INSPIRE Synthetic.** Speaker and Style denote same-speaker and same-speaking-style retrieval. Random is the expected value under uniform ranking, and Cascade denotes transcription- and caption-based text retrieval. Bold indicates the best result in each column.

Model	Interface	Speaker Recall@10 ↑	Style Recall@10 ↑
Random (expected)	–	0.23	0.28
WavLM-Large	Static	0.25	4.75
LAION-CLAP	Static	0.50	1.00
TP-CLAP	Fixed	0.50	0.50
WAVE	Fixed	0.00	0.00
ALM2Vec-PT	Fixed	1.00	0.50
ALM2Vec-FT	Fixed	0.25	1.00
Qwen3-Embedding-8B cascade	Cascade	0.00	0.25
JSRL	Fixed	**11.75**	**10.75**

5 Conclusion

This paper characterizes selective speech representations as a criterion-dependent similarity learning problem and reveals a matching–selection gap through attribute competition and criterion reversal. We propose JSRL, which organizes training around four joint relevance states so that criterion-wise ranking supervision covers both relevance persistence and switching. Across three held-out corpora, JSRL improves attribute-conflict ranking, criterion reversal, and multi-candidate retrieval, while the learned ranking behaviors recombine on the untrained S–C pair and transfer to INSPIRE Synthetic. More broadly, our results suggest that selective speech representation should be evaluated and learned in terms of whether the current criterion actually reorganizes similarity relationships under competing attributes, rather than only whether the target attribute can be matched.

Reproducibility Statement

The appendix provides training details for this paper. Appendix A gives the evaluation corpus filtering and label processing, joint state construction rules, metric definitions, and candidate pool sampling and aggregation rules. Appendix B gives the input format, representation extraction position, training instructions, three-stage training configurations, and optimization hyperparameters. Appendix C gives the representation extraction methods and retrieval interface implementations for each baseline, and Appendix D gives the INSPIRE evaluation procedure and WavLM reproduction comparison. The training and evaluation corpora used in this paper are all available through their official channels. Code and evaluation panels will be released upon paper acceptance.

AI Use Statement

We used generative AI tools throughout the research and writing process. AI tools were used to assist with literature retrieval, discovery, and summarization; to provide feedback on research framing, methodology, and experimental design; and to assist with the interpretation of experimental results. They were also used to translate and polish portions of the manuscript, assist with code comments and formatting, and create or edit visual assets for figures. All AI-assisted suggestions and outputs were critically reviewed and verified by the authors.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, 2020.

Carlos Busso, Srinivas Parthasarathy, Alec Burmania, Mohammed AbdelWahab, Najmeh Sadoughi, and Emily Mower Provost. MSP-IMPROV: An Acted Corpus of Dyadic Interactions to Study Emotion Perception. *IEEE Transactions on Affective Computing*, 8(1):67–80, 2017. doi: 10.1109/taffc.2016.2515617.

Carlos Busso, Reza Lotfian, Kusha Sridhar, Ali N. Salman, Wei-Cheng Lin, Lucas Goncalves, Srinivas Parthasarathy, Abinay Reddy Naini, Seong-Gyun Leem, Luz Martinez-Lucas, et al. The MSP-Podcast Corpus. *IEEE Transactions on Affective Computing*, 17(3):3065–3083, 2026. doi: 10.1109/taffc.2026.3678489.

Houwei Cao, David G. Cooper, Michael K. Keutmann, Ruben C. Gur, Ani Nenkova, and Ragini Verma. CREMA-D: Crowd-Sourced Emotional Multimodal Actors Dataset. *IEEE Transactions on Affective Computing*, 5(4):377–390, 2014. doi: 10.1109/taffc.2014.2336244.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1505–1518, 2022. doi: 10.1109/jstsp.2022.3188113.

Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Interspeech*, 2020.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. doi: 10.1109/taslp.2021.3122291.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Wenjun Huang, Qiaosong Chu, Tiger Shao, Pengfei Zhang, Yutong Song, Hanning Chen, Yezi Liu, Weiyi Wu, SungHeon Jeong, Ryozo Masukawa, et al. AudioLens: Multi-Perspective Speech Clustering with Reasoning Audio-Language Models. *arXiv preprint arXiv:2608.25177*, 2026.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. doi: 10.1145/582415.582418.

Yuma Koizumi, Heiga Zen, Shigeki Karita, Yifan Ding, Kohei Yatabe, Nobuyuki Morioka, Michiel Bacchiani, Yu Zhang, Wei Han, and Ankur Bapna. LibriTTS-R: A Restored Multi-Speaker Text-to-Speech Corpus. In *Interspeech*, 2023.

Chen Li, Peiji Yang, Yicheng Zhong, Jianxing Yu, Zhisheng Wang, Zihao Gou, Wenqing Chen, and Jian Yin. HPSU: A Benchmark for Human-Level Perception in Real-World Spoken Speech Understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026a.

Chen-An Li and Hung-yi Lee. INSPIRE: A Benchmark for Instruction-Aware Speech Retrieval. In *Interspeech*, 2026.

Mohan Li, Rama Doddipatla, and Philip C. Woodland. Text-Prompted CLAP: Learning Text-Conditioned Audio Representations via Contrastive Learning. *arXiv preprint arXiv:2607.25085*, 2026b.

Steven R. Livingstone and Frank A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLOS ONE*, 13(5):e0196391, 2018. doi: 10.1371/journal.pone.0196391.

Fengjie Lu, Chenang Jiang, Jiarui Hai, Helin Wang, and Aaron Yee. ALM2Vec: Learning Audio Embeddings for Universal Audio Retrieval with Large Audio-Language Models. *arXiv preprint arXiv:2606.30682*, 2026.

Ziyang Ma, Zhisheng Zheng, Jiaxin Ye, Jinchao Li, Zhifu Gao, ShiLiang Zhang, and Xie Chen. emotion2vec: Self-Supervised Pre-Training for Speech Emotion Representation. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Julius Richter, Yi-Chiao Wu, Steven Krenn, Simon Welker, Bunlong Lay, Shinji Watanabe, Alexander Richard, and Timo Gerkmann. EARS: An Anechoic Fullband Speech Dataset Benchmarked for Speech Enhancement and Dereverberation. In *Interspeech*, 2024.

Changli Tang, Qinfan Xiao, Ke Mei, Tianyi Wang, Fengyun Rao, and Chao Zhang. WAVE: Learning Unified & Versatile Audio-Visual Embeddings with Multimodal LLM. In *International Conference on Learning Representations*, 2026.

Sagar Vaze, Nicolas Carion, and Ishan Misra. GeneCIS: A Benchmark for General Conditional Image Similarity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional Similarity Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn Lawrie, and Luca Soldaini. FollowIR: Evaluating and Teaching Information Retrieval Models to Follow Instructions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025a.

Orion Weller, Benjamin Van Durme, Dawn Lawrie, Ashwin Paranjape, Yuhao Zhang, and Jack Hessel. Promptriever: Instruction-Trained Retrievers Can Be Prompted Like Language Models. In *International Conference on Learning Representations*, 2025b.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-Scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-Omni Technical Report. *arXiv preprint arXiv:2509.17765*, 2025.

Chen Yang, Chufan Yu, Hanfu Chen, Jie Zhu, Jingqi Chen, Ke Chen, Wenxuan Wang, Yang Wang, Yaozhou Jiang, Yi Jiang, et al. MOSS-Audio Technical Report. *arXiv preprint arXiv:2606.01802*, 2026.

Qingcheng Zeng, Puxuan Yu, Aman Mehta, Fuheng Zhao, and Rajhans Samdani. Dual-View Training for Instruction-Following Information Retrieval. *arXiv preprint arXiv:2604.18845*, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176*, 2025.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. Seen and Unseen Emotional Style Transfer for Voice Conversion with A New Emotional Speech Dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021.

A Evaluation Protocol and Panel Construction

A.1 Evaluation Corpora and Label Processing

We evaluate speech retrieval under the speaker identity (S), emotion (E), and linguistic content (C) criteria on CREMA-D (Cao et al., 2014), MSP-IMPROV (Busso et al., 2017), and RAVDESS (Livingstone & Russo, 2018). None of the three corpora are used in training. Within each corpus, speaker match is defined by speaker identity, emotion match by discrete emotion labels, and content match by sentence identity. The corpus subsets and label processing used for evaluation are as follows.

CREMA-D. We use the subset of recordings where the listener-perceived emotion agrees with the actor-intended emotion, totaling 3,100 utterances. Based on listener emotion votes under the audio-only condition, a one-sided binomial test is performed against a chance probability of $1/6$ for the highest vote count, and recordings are retained where $p < 0.05$ and the actor-intended emotion is among the set of highest-voted labels. Evaluation uses the original speaker, sentence, and intended emotion labels from the file names, where emotions include anger, disgust, fear, happiness, neutral, and sadness.

MSP-IMPROV. From the 640 recordings in the Target-Improvised section, we retain the 476 recordings where the listener consensus emotion matches the actor-intended emotion, applying the same filtering to both queries and candidate sets. Emotion labels use the listener consensus annotation, covering anger, happiness, sadness, and neutral. Speaker and target sentence identifiers remain unchanged.

RAVDESS. We use the 1,438 recordings from the speech section after removing two duplicate recordings, using the corpus's speaker, emotion, and sentence identifiers. Emotion labels retain the original categories, with calm and neutral treated separately. Content labels correspond to two different sentences.

The above subsets form the common corpus basis for all evaluation panels. Specific panels determine query eligibility and candidate sets on this basis; their joint state construction and sampling rules are described in subsequent subsections.

A.2 Joint State Construction

For a query q and candidate d from the same corpus, we determine whether they match on each attribute according to the label definitions in Appendix A.1. Given an ordered criterion pair (f, g), following the joint state definition in the main text, candidates are partitioned into $\mathcal{D}_{11}$, $\mathcal{D}_{10}$, $\mathcal{D}_{01}$, and $\mathcal{D}_{00}$. The first subscript indicates whether the candidate matches the query under criterion f; the second indicates whether it matches under criterion g. The specific correspondences for the three criterion pairs are as follows.

Table 3: **Joint state definitions for three criterion pairs.** All attribute matches are determined relative to the query.

Candidate set	S–E	E–C	S–C
$\mathcal{D}_{11}$	Same speaker same emotion	Same emotion same content	Same speaker same content
$\mathcal{D}_{10}$	Same speaker diff. emotion	Same emotion diff. content	Same speaker diff. content
$\mathcal{D}_{01}$	Diff. speaker same emotion	Diff. emotion same content	Diff. speaker same content
$\mathcal{D}_{00}$	Diff. speaker diff. emotion	Diff. emotion diff. content	Diff. speaker diff. content

In the attribute competition evaluation, we further require that the candidate does not match the query on the third attribute: S–E excludes same-content candidates, E–C excludes same-speaker

candidates, and S–C excludes same-emotion candidates. This ensures that the third attribute's match status is consistent across the four joint states. All candidates come from the same corpus as the query, excluding the query itself.

Query eligibility is determined independently for each criterion pair. A query enters the corresponding attribute competition panel only when all four joint states are non-empty after the third-attribute filtering. For eligible queries, all candidates in each state are used in evaluation, preserving the actual sizes of each state. The Aligned, Balanced, Conflict, and SwitchExact metrics for the same query are all computed based on this candidate set, and all models share the same queries and candidate sets.

Multi-candidate retrieval uses the same joint state labels but excludes only the query itself when constructing candidate pools, allowing the third attribute to match the query. Query eligibility is determined by the number of candidates required for each state; specific sampling rules are given in Appendix A.5.

A.3 Attribute Competition and Criterion Reversal Evaluation Protocol

The attribute competition evaluation is based on the four joint states constructed in Appendix A.2 and characterizes performance under three conditions: the other attribute supports the current criterion's ranking, has the same match status relative to the query, or supports the opposite ranking. For a query q, current criterion c, and two candidate sets U, V, define the strict pairwise ranking accuracy

$$R_c(U, V \mid q) = \frac{1}{|U||V|} \sum_{u \in U} \sum_{v \in V} \mathbf{1}[s_c(q, u) > s_c(q, v)], \tag{6}$$

where $s_c(q, d)$ is the retrieval score under criterion c. This metric enumerates all candidate pairs between the two sets; ties are not counted as correct. Below, the dependence of sets and metrics on query q is omitted.

Aligned. Compares $\mathcal{D}_{11}$ (matching both attributes) with $\mathcal{D}_{00}$ (matching neither). Here, the two attributes support the same ranking direction:

$$A_f = R_f(\mathcal{D}_{11}, \mathcal{D}_{00}), \qquad A_g = R_g(\mathcal{D}_{11}, \mathcal{D}_{00}). \tag{7}$$

Balanced. We separately compare positives and negatives under the current criterion when the other attribute matches the query and when it does not, then average the two pairwise accuracies with equal weight:

$$\begin{aligned} B_f &= \tfrac{1}{2}[R_f(\mathcal{D}_{11}, \mathcal{D}_{01}) + R_f(\mathcal{D}_{10}, \mathcal{D}_{00})], \\ B_g &= \tfrac{1}{2}[R_g(\mathcal{D}_{11}, \mathcal{D}_{10}) + R_g(\mathcal{D}_{01}, \mathcal{D}_{00})]. \end{aligned} \tag{8}$$

Equal averaging ensures both conditions contribute equally to the metric, regardless of the number of candidates in each state.

Conflict. Compares $\mathcal{D}_{10}$ and $\mathcal{D}_{01}$, each matching only one attribute. The two criteria require opposite ranking directions:

$$C_f = R_f(\mathcal{D}_{10}, \mathcal{D}_{01}), \qquad C_g = R_g(\mathcal{D}_{01}, \mathcal{D}_{10}). \tag{9}$$

For each query, Aligned, Balanced, and Conflict are each averaged over the two criterion directions: $A = (A_f + A_g)/2$, $B = (B_f + B_g)/2$, and $C = (C_f + C_g)/2$.

SwitchExact. We fix the same query and the same conflicting candidate pair $(u, v) \in \mathcal{D}_{10} \times \mathcal{D}_{01}$, and compute scores under both criteria. A candidate pair counts as successful only when u is strictly preferred over v under criterion f, and v is strictly preferred over u under criterion g:

$$\mathrm{SwitchExact}(q) = \frac{\sum_{u \in \mathcal{D}_{10}} \sum_{v \in \mathcal{D}_{01}} \mathbf{1}[s_f(q, u) > s_f(q, v)]\, \mathbf{1}[s_g(q, v) > s_g(q, u)]}{|\mathcal{D}_{10}|\,|\mathcal{D}_{01}|}. \tag{10}$$

Thus, Conflict measures the average ranking accuracy across the two criterion directions, while SwitchExact measures the proportion of candidate pairs that are simultaneously correct in both directions. A tie in either direction does not satisfy the SwitchExact success condition.

Aggregation. Each metric is computed within a query as defined above; aggregation across queries, corpora, and criterion pairs follows Appendix A.7.

A.4 Persistence Metrics

When criteria switch, candidates in $\mathcal{D}_{11}$ remain relevant under both criteria, while candidates in $\mathcal{D}_{00}$ remain irrelevant under both. We measure whether the ranking relationships of these two types of candidates are preserved. The evaluation uses the attribute competition candidate sets and query eligibility from Appendix A.2.

For a query q, criterion c, and candidate sets U, V, define

$$\mathrm{AUC}_c(U, V \mid q) = \frac{1}{|U||V|} \sum_{u \in U} \sum_{v \in V} \left(\mathbf{1}[s_c(q,u) > s_c(q,v)] + \frac{1}{2}\mathbf{1}[s_c(q,u) = s_c(q,v)] \right). \quad (11)$$

This metric averages over all cross-set candidate pairs, counting ties as 0.5. When computing over unions of multiple states, candidates are merged directly and pairwise comparisons preserve the weights corresponding to actual counts in each state.

Persistent relevance. For persistently relevant $\mathcal{D}_{11}$, we measure how well it ranks above all negatives under each criterion, and take the lower value between the two criteria:

$$P_{11}(q) = \min_{c \in \{f,g\}} \mathrm{AUC}_c(\mathcal{D}_{11}, \mathcal{N}_c \mid q), \quad (12)$$

where

$$\mathcal{N}_f = \mathcal{D}_{01} \cup \mathcal{D}_{00}, \qquad \mathcal{N}_g = \mathcal{D}_{10} \cup \mathcal{D}_{00}. \quad (13)$$

This metric captures how well persistently relevant candidates rank above negatives under both criteria.

Persistent irrelevance. For persistently irrelevant $\mathcal{D}_{00}$, we measure how well all positives rank above it, again taking the lower value between the two criteria:

$$P_{00}(q) = \min_{c \in \{f,g\}} \mathrm{AUC}_c(\mathcal{P}_c, \mathcal{D}_{00} \mid q), \quad (14)$$

where

$$\mathcal{P}_f = \mathcal{D}_{11} \cup \mathcal{D}_{10}, \qquad \mathcal{P}_g = \mathcal{D}_{11} \cup \mathcal{D}_{01}. \quad (15)$$

This metric captures how well persistently irrelevant candidates remain ranked below positives under both criteria. The minimum in both metrics is computed within a single query, so each query's result is determined by the weaker criterion direction.

Aggregation. P_{11} and P_{00} for each query are aggregated following Appendix A.7, and the aggregated values are still denoted P_{11} and P_{00}. The Persistence shown in Figure 4 in the main text is the equal-weight average over these two metrics and the S–E and E–C criterion pairs. Persistence measures the ranking relationships corresponding to relevant and irrelevant identities after a criterion switch, not whether candidate scores remain unchanged.

Correspondence between supervision and metrics. This set of metrics corresponds to the categories of candidate relationships involved in Eq. (4). Under criterion f, Eq. (4) compares four state combinations between $\mathcal{P}_f = \mathcal{D}_{11} \cup \mathcal{D}_{10}$ and $\mathcal{N}_f = \mathcal{D}_{01} \cup \mathcal{D}_{00}$, where only the comparison between $\mathcal{D}_{10}$ and $\mathcal{D}_{01}$ belongs to the conflicting pairs examined by Conflict and SwitchExact; the other three combinations involve persistent-state candidates and constitute the comparisons examined by P_{11} and P_{00}. Criterion g is symmetric. JSRL w/o persistence and JSRL differ only in whether the latter category of comparisons is supervised. We use this to test whether persistence supervision improves Persistence and multi-candidate retrieval while retaining criterion reversal ability. Multi-candidate retrieval particularly depends on the latter category: among the 48 positive–negative candidate pairs formed by 4 positives and 12 negatives under each criterion, only 9 are conflicting pairs.

A.5 Multi-Candidate Retrieval Construction and Metrics

We evaluate multi-candidate retrieval quality using fixed-size candidate pools. For a query q and criterion pair (f, g), candidates from the same corpus (excluding the query itself) are partitioned according to the joint state definitions in Appendix A.2. Here, candidates are allowed to match the query on the third attribute.

Candidate sampling. Each candidate pool samples 1, 3, 3, and 9 utterances from $\mathcal{D}_{11}$, $\mathcal{D}_{10}$, $\mathcal{D}_{01}$, and $\mathcal{D}_{00}$ respectively, totaling 16 candidates. Under either criterion, the candidate pool contains 4 positives and 12 negatives (Table 4).

Table 4: **Joint state composition and criterion relevance of candidate pools in multi-candidate retrieval.**

Joint state	Count	Under criterion f	Under criterion g
$\mathcal{D}_{11}$	1	Positive	Positive
$\mathcal{D}_{10}$	3	Positive	Negative
$\mathcal{D}_{01}$	3	Negative	Positive
$\mathcal{D}_{00}$	9	Negative	Negative

A query enters the retrieval evaluation for the corresponding criterion pair only when the number of candidates in all four states meets the above sampling requirements. Each sampling is performed uniformly without replacement within each state, without selecting the same candidate twice. For each eligible query, ten candidate pools are generated using ten fixed random seeds; candidates may overlap across different samplings. The candidate pools are pre-generated, and all models and both criterion directions within the same criterion pair share the same candidate sets. These ten repetitions correspond to candidate sampling, not independent training repetitions.

Ranking and relevance. The model ranks the 16 candidates in descending order of the score under the current criterion; ties are broken by lexicographic order of the candidate's unique identifier. Whether a candidate is relevant is determined by label match under the current criterion. Let $r_k \in \{0, 1\}$ denote the relevance of the k-th ranked candidate; each retrieval satisfies $\sum_{k=1}^{16} r_k = 4$.

Retrieval metrics. We report mAP, NDCG@5, and Recall@5. The average precision for a single retrieval is

$$\mathrm{AP} = \frac{1}{4} \sum_{k=1}^{16} r_k \left(\frac{1}{k} \sum_{j=1}^{k} r_j \right), \tag{16}$$

covering all 16 ranking positions. NDCG@5 uses binary relevance gain and logarithmic position discount:

$$\mathrm{NDCG@5} = \frac{\displaystyle\sum_{k=1}^{5} \frac{r_k}{\log_2(k+1)}}{\displaystyle\sum_{k=1}^{4} \frac{1}{\log_2(k+1)}}. \tag{17}$$

The denominator corresponds to the ideal ranking where four positives occupy the top four positions. Recall@5 is the proportion of positives retrieved in the top five positions:

$$\mathrm{Recall@5} = \frac{1}{4} \sum_{k=1}^{5} r_k. \tag{18}$$

Aggregation. Each metric is first averaged over the ten candidate samplings for the same query, then aggregated across queries, corpora, and criterion contexts following Appendix A.7. AP aggregated in this manner yields mAP. S–C retrieval uses the same candidate proportions, repetition count, and metric definitions; its aggregated results are reported separately.

A.6 Evaluation Panel Statistics

Table 5 lists the number of eligible queries and candidate pools for each corpus and criterion pair. The attribute competition evaluation and the criterion reversal evaluation use the same set of eligible queries; multi-candidate retrieval determines query eligibility independently based on the sampling count requirements for each state. Each retrieval query corresponds to ten candidate samplings, and the same candidate pool is shared across both criterion directions and all compared models.

Table 5: **Evaluation panel sizes.** The candidate pool count reflects pool instances generated across ten samplings, without deduplication of identical candidate sets.

Corpus	Utterances	Criterion pair	Attr. comp. queries	Retrieval queries	Candidate pools
CREMA-D	3,100	S–E	3,034	3,044	30,440
CREMA-D	3,100	E–C	3,100	3,100	31,000
CREMA-D	3,100	S–C	2,925	2,925	29,250
MSP-IMPROV	476	S–E	475	475	4,750
MSP-IMPROV	476	E–C	476	476	4,760
MSP-IMPROV	476	S–C	465	465	4,650
RAVDESS	1,438	S–E	1,438	1,438	14,380
RAVDESS	1,438	E–C	1,438	1,438	14,380
RAVDESS	1,438	S–C	1,438	1,438	14,380

The candidate pool count is computed as the number of eligible queries multiplied by ten samplings, without multiplying by the number of models or criterion directions. Candidate pools from different samplings may overlap. Query eligibility is determined independently for each criterion pair, and the same utterance may participate in evaluations for multiple criterion pairs.

The number of eligible queries may differ between attribute competition evaluation and multi-candidate retrieval, because they use different candidate filtering conditions. In the S–E panel of CREMA-D, attribute competition evaluation retains 3,034 queries, while multi-candidate retrieval (which allows third-attribute matching) retains 3,044 queries. All metrics are still aggregated with equal weight over queries and corpora within their respective eligible query sets.

A.7 Aggregation Rules

We aggregate evaluation results hierarchically across queries, corpora, and criterion contexts. For corpus d and criterion pair e, let the set of eligible queries be $\mathcal{Q}_{d,e}$ and the metric for a single query be $m(q; e)$. The within-corpus result is

$$M_{d,e} = \frac{1}{|\mathcal{Q}_{d,e}|} \sum_{q \in \mathcal{Q}_{d,e}} m(q; e), \tag{19}$$

and the three-corpus macro-average is

$$M_e = \frac{1}{3} \sum_{d=1}^{3} M_{d,e}. \tag{20}$$

Each query has equal weight within its corpus, and the three corpora have equal weight in the final result.

Within-query computation. Attribute competition and criterion reversal evaluations use all candidate pairs as defined in Appendix A.3, with the accuracies under the two Balanced conditions averaged equally. Persistence first takes the lower value of the two criterion directions within a single query (Appendix A.4), then averages across queries and corpora. Multi-candidate retrieval first

averages over the ten samplings for the same query:

$$m(q; e, c) = \frac{1}{10} \sum_{t=1}^{10} m^{(t)}(q; e, c), \tag{21}$$

where c is the current retrieval criterion and t is the candidate sampling index. The result is then aggregated across queries and corpora as described above.

Criterion pair aggregation. Results for S–E, E–C, and S–C are reported separately. When aggregating Aligned, Balanced, Conflict, SwitchExact, or Persistence over trained criterion pairs, S–E and E–C are equally averaged:

$$M_{\text{trained}} = \frac{1}{2} \left(M_{\text{S–E}} + M_{\text{E–C}} \right). \tag{22}$$

Here, Aligned, Balanced, and Conflict have already been averaged over the two criterion directions within each query, while SwitchExact directly measures the proportion of candidate pairs that are simultaneously correct in both directions.

S/E/C retrieval aggregation. Let $M_{e,c}$ denote the three-corpus macro-averaged result for retrieval under criterion c within criterion pair e. The main retrieval table first forms three single-criterion results:

$$\begin{aligned} M_{\text{S}} &= M_{\text{S–E,S}}, \\ M_{\text{E}} &= \tfrac{1}{2} \left(M_{\text{S–E,E}} + M_{\text{E–C,E}} \right), \\ M_{\text{C}} &= M_{\text{E–C,C}}, \end{aligned} \tag{23}$$

then computes

$$M_{\text{SEC}} = \frac{1}{3} \left(M_{\text{S}} + M_{\text{E}} + M_{\text{C}} \right). \tag{24}$$

Thus, the two emotion contexts together receive the same total weight as one criterion. S–C retrieval equally averages its S and C directions and is reported separately, not included in this S/E/C aggregate.

A.8 Random Seed Stability and Evaluation Uncertainty

To test the sensitivity of Stage 3 training conclusions to randomness, we repeat subsequent training from the same Stage 2 checkpoint using 3 random seeds for each of Stage 2 continuation, JSRL w/o persistence, and JSRL. The main comparisons are consistent across the three seeds: JSRL w/o persistence improves SwitchExact relative to Stage 2 continuation in all seeds; after adding persistence supervision, JSRL further improves Persistence and multi-candidate mAP across all three seeds, while SwitchExact does not show a consistent additional change. The cross-seed sample standard deviations of these metrics are all within 0.76 percentage points, indicating that the main conclusions in the main text about Stage 3 relevance switching, persistence, and overall retrieval gains are stable to training randomness. The main text and other appendices use the same seed-0 configuration; the multi-seed repetitions are used to verify the stability of the main conclusions, not to select the reported seed based on multi-seed results.

Evaluation uncertainty is estimated by paired cluster bootstrap with 10,000 resamples using the query speaker as the clustering unit; all models share the same resampling, preserving the per-query pairing between models. Error bars in figures use this method. For the core Stage 3 model comparisons under the same paired bootstrap, the SwitchExact improvement of JSRL w/o persistence over Stage 2 continuation, and the Persistence and multi-candidate mAP improvements of JSRL over JSRL w/o persistence, all have 95% confidence intervals that do not cross zero, consistent with the comparisons observed in the multi-seed experiments. On the common query set for the candidate pool size sensitivity experiment, JSRL's S/E/C mAP advantage over WAVE (Dynamic) at pool sizes of 16, 32, and 64 candidates is 4.06, 4.15, and 4.23 percentage points, with corresponding 95% confidence intervals of [3.46, 4.67], [3.51, 4.79], and [3.54, 4.90]. These intervals characterize the uncertainty from finite evaluation samples and do not treat different pool sizes as independent hypothesis tests. Training randomness and evaluation uncertainty are reported separately and not combined into a single interval.

B Model and Training Details

B.1 Full Input Format and Representation Extraction

We use MOSS-Audio-4B-Instruct as the shared speech encoder. The natural language instruction is placed in the system message, speech in the user message, and the input ends with an assistant prefix. The query and candidate use the same message structure, differing only in the instruction content:

```
<|im_start|>system
{instruction}<|im_end|>
<|im_start|>user
<|audio_bos|><|AUDIO|><|audio_eos|><|im_end|>
<|im_start|>assistant
```

where `<|AUDIO|>` denotes the speech input expanded by the model's audio processor, and the assistant prefix retains a trailing newline. The model directly extracts the final-layer hidden state at the last position of this prefix, without generating a text response. This hidden state is 2,560-dimensional and is projected to 512 dimensions through a shared linear projection, followed by L_2 normalization. The inner product of the query and candidate representations serves as the retrieval score.

Table 6 lists some examples of training prompts.

Table 6: **Training prompt examples.**

Criterion	Instruction
Speaker	Retrieve speech from the same speaker. Find the recording spoken by the same person. Compare stable vocal characteristics that identify the speaker. Retrieve by speaker identity.
Emotion	Retrieve speech expressing the same emotion. Find the recording with the most similar vocal emotion. Compare the affective state conveyed by the voice. Retrieve by expressed emotion.
Content	Retrieve speech with the same spoken content. Find the recording with the same spoken content. Compare the linguistic message conveyed by the utterance. Retrieve by speech content.

The generic candidate instruction for Stage 1 is:

```
Represent this speech recording without selecting a particular
content, speaker, or emotion criterion.
```

Stages 2 and 3 uniformly use the following candidate instruction:

```
Represent this speech recording for retrieval.
```

B.2 Three-Stage Training Details

The training configuration for each stage is given in Table 7.

Table 7: **Three-stage training details.**

Stage	Setting	Value
Stage 1	Training datasets	ESD, EARS, MSP-Podcast, LibriTTS-R
	Global batch	64 training units
	Micro-batch	4
	LoRA peak LR	5×10^{-5}
	Projection peak LR	1×10^{-4}
Stage 2	Training datasets	ESD, MSP-Podcast
	Global batch	16 competition units
	Micro-batch	4
	LoRA peak LR	1×10^{-5}
	Projection peak LR	2×10^{-5}
Stage 3	Training datasets	ESD, MSP-Podcast
	Global batch	Regular steps: 24; structured steps: 24 per criterion pair
	Micro-batch	8
	LoRA peak LR	1×10^{-5}
	Projection peak LR	2×10^{-5}

The attributes and criterion pairs covered by the three stages are shown in Table 8.

Table 8: **Attribute and criterion pair coverage across training stages.**

Stage	S	E	C	S–E pair training	E–C pair training	S–C pair training
Stage 1	Yes	Yes	Yes	No	No	No
Stage 2	Yes	Yes	Yes	Yes	Yes	No
Stage 3	Yes	Yes	Yes	Yes	Yes	No

Both attributes in S–C appear during training, but their pairing is not used in Stage 2 or Stage 3 criterion-pair training.

B.3 Optimization and Compute Settings

Training updates the LoRA parameters in attention layers and the shared projection layer; all other base model parameters are frozen. The main configuration is given in Table 9.

Table 9: **Optimization and compute settings.**

Item	Configuration
Base model	MOSS-Audio-4B-Instruct
Audio sample rate	16 kHz
Decoder layers / hidden dim	36 / 2,560
LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`
LoRA rank / alpha	16 / 32
LoRA dropout	0
Projection dim / dropout	512 / 0
Optimizer	AdamW
Weight decay	10^{-3}
Gradient clip norm	1.0
Compute precision	BF16

B.4 Training Stages and Ablation Setup

We compare Stage 2 and its subsequent training branches to analyze the effects of additional training, relevance switching supervision, and full joint-state supervision. The training signals and data schedules for each branch are given in Table 10.

Table 10: **Training stages and subsequent training branches.**

Model	Starting point	Training signal	Data schedule
Stage 2	Stage 1	Hard-negative InfoNCE under attribute competition	Original attribute competition schedule
Stage 2 continuation	Stage 2	Continuing with original attribute competition objective	Same as JSRL
JSRL w/o persistence	Stage 2	Only supervising $\mathcal{D}_{10}/\mathcal{D}_{01}$ relevance switching	Same as JSRL
JSRL	Stage 2	Set-based ranking over four joint states	MSP-Podcast/ESD mixed S–E, ESD E–C

Stage 2 continuation tests the effect of additional attribute competition training. The only difference between JSRL w/o persistence and JSRL is whether persistence is supervised; their comparison tests the effect of persistence supervision. All models use the common evaluation queries, candidate sets, and aggregation rules defined in Appendix A.

C Baseline Models and Retrieval Interfaces

C.1 Baseline Models and Representation Extraction

We compare static speech representations, instruction-conditioned representations, and the base model without our retrieval training. Each baseline uses its published model weights for inference; representation extraction methods are given in Table 11. Retrieval representations are L_2-normalized, and cosine similarity is computed via inner product.

Table 11: **Representation extraction and retrieval interfaces of baseline models.**

Model	Representation extraction	Interface
WavLM-Large	Mean pooling over valid audio frames in the last layer, excluding padding	Static
LAION-CLAP	Audio embedding with native fusion processing	Static
TP-CLAP	Query: text-instruction-conditioned audio embedding; candidate: unconditional audio embedding	Fixed
WAVE	Audio–instruction joint representation from the model's native embedding head	Fixed, Dynamic
ALM2Vec-PT/FT	Representations via the model's query/document encoding interface	Fixed, Dynamic
Frozen MOSS-Audio	Final-layer hidden state at the last position of the assistant prefix, directly normalized, 2,560-dim	Fixed

Frozen MOSS-Audio and JSRL use the same base model and hidden state extraction position. Frozen MOSS-Audio freezes all parameters and directly uses the base model hidden states. JSRL updates LoRA parameters and the shared projection layer through three-stage retrieval training, mapping hidden states to 512 dimensions. This comparison examines the effect of our retrieval training on criterion-conditioned ranking.

C.2 Fixed and Dynamic Candidate Index Implementation

This paper classifies retrieval interfaces as Fixed or Dynamic based on whether the candidate representation changes with the retrieval criterion, and implements the corresponding settings based on each model's encoding interface.

The Static interface uses criterion-independent audio representations on both the query and candidate sides. WavLM-Large and LAION-CLAP extract a single representation for each audio and reuse the same scores across the Speaker, Emotion, and Content criteria. WavLM-Large uses the mean of valid frames from the last layer of the official model; LAION-CLAP resamples input audio to 48 kHz and uses its native audio processor and fusion configuration.

The Fixed interface introduces the current criterion only on the query side; candidate representations remain unchanged across criteria. Specific implementations for each model are as follows.

TP-CLAP. A shared audio encoder feature is extracted for each audio. The query side feeds the current criterion instruction along with this feature into the conditioned audio embedding interface; the candidate side uses the unconditional audio embedding interface. Speaker, Emotion, and Content are specified through corresponding natural language instructions, and the candidate index is shared across the three criteria. Inference uses single-audio batches, preserving the native pooling method.

WAVE. The query side inputs the audio and the current criterion instruction through the model's native chat template and extracts the embedding head output. The Fixed candidate side uniformly uses the generic instruction `Represent this audio for retrieval.` The Dynamic candidate side uses the same criterion instruction as the query, so each candidate audio produces separate Speaker, Emotion, and Content representations, and the index corresponding to the current criterion is selected.

ALM2Vec. PT and FT checkpoints use the same interface setup. The query side calls the encoding interface with audio and criterion text using `task=query`. The Fixed candidate side calls `task=document` with audio only; the Dynamic candidate side calls `task=query` with both audio and the same criterion text as the query. The two settings correspond to Q→D and Q→Q respectively.

Frozen MOSS-Audio and JSRL. Both use the Fixed interface: the query side inputs the current criterion instruction, and the candidate side uniformly uses `Represent this speech recording for retrieval.` The full message structure and representation extraction position are given in Appendix B.1.

The Fixed and Dynamic comparisons for the same model use the same checkpoint, query instructions, evaluation audio, and candidate sets, differing only in the candidate-side encoding method. Both interfaces compute metrics following the protocol in Appendix A.

C.3 Evaluation Instructions

Evaluation instructions differ in wording and sentence structure from training instructions. We first pre-sample one evaluation instruction for each query–criterion pair using a fixed random seed, and reuse the same instruction assignment across all instruction-conditioned models and both Fixed and Dynamic interfaces. Static models do not use instructions and directly use their audio representations.

Table 12: **Evaluation instruction examples.**

Criterion	Instruction
Speaker	Match the speaker identity.
Emotion	Match the expressed emotion.
Content	Match what is being said.

D INSPIRE IMPLEMENTATION AND EXTERNAL EVALUATION

D.1 WAVLM REPRODUCTION ON FULL SYNTHETIC

To verify the INSPIRE Synthetic evaluation pipeline, we reproduce results using WavLM-Large on all 15 retrieval tasks, covering 3,000 query–instruction pairs composed of 200 query audios and 5,400 candidate recordings, and retaining the official positive and candidate exclusion rules. Table 13 compares the reproduction results with INSPIRE's original Table 4 (Li & Lee, 2026). The reproduced Recall@10, Recall@50, and Recall@100 are 1.78%, 4.47%, and 6.88%, with absolute differences of 0.08, 0.01, and 0.34 percentage points from the original. The close agreement with the original results on the full Synthetic benchmark provides a numerical check of the INSPIRE evaluation pipeline used in this paper.

Table 13: **WavLM-Large reproduction on full INSPIRE Synthetic.** Recall in %; the difference is reproduction minus original, in percentage points.

Source	Recall@10 ↑	Recall@50 ↑	Recall@100 ↑
INSPIRE original	1.70	4.48	7.22
Our reproduction	1.78	4.47	6.88
Difference	+0.08	−0.01	−0.34

D.2 EVALUATION TASKS AND CANDIDATE SETS

We select two single-attribute tasks from INSPIRE Synthetic that correspond to this paper's speaker and emotion criteria for external evaluation: same-speaker defines relevance by speaker identity; same-speaking-style defines relevance by expressive speaking style, providing an external test of emotional expression. Both tasks retain the official query instructions and share 5,400 candidate recordings. Each task includes 200 queries, and each query has two positives.

Table 14: **Evaluation sizes of two retrieval tasks in INSPIRE Synthetic.** The valid candidate count is the range after applying official per-query exclusion lists.

Task	Queries	Pool size	Positives per query	Valid candidates
Same-speaker	200	5,400	2	4,276–4,357
Same-speaking-style	200	5,400	2	3,593–3,616

All instruction-conditioned methods input the official instruction only on the query side, use criterion-independent representations on the candidate side, and adopt the Fixed interface. WavLM-Large and LAION-CLAP use the Static interface. All methods share the same queries, positives, and valid candidate sets. Retrieval is performed over the full valid candidate pool, without using the multi-candidate retrieval panels from Appendix A.5.

Let $\mathcal{P}_q$ be the official positive set for query q, and let $\mathcal{T}_K(q)$ be the top K items in the ranked valid candidates. Then

$$\text{Recall@}K(q) = \frac{|\mathcal{T}_K(q) \cap \mathcal{P}_q|}{|\mathcal{P}_q|}. \tag{25}$$

Each task equally averages over its 200 queries, reporting $K \in \{10, 50, 100\}$. Random is the exact expected value under uniform random ranking. For a query with N_q valid candidates, the expected Recall@K is $\min(K, N_q)/N_q$, then averaged over queries. Therefore, the random reference values for the two tasks differ according to their respective valid candidate pool sizes.

D.3 TEXT CASCADE BASELINE

The Qwen3-Embedding-8B cascade first converts audio to transcriptions and descriptions, then performs text retrieval. Whisper large-v3 (Radford et al., 2023) generates the transcription, and Qwen3-Omni-30B-A3B-Captioner (Xu et al., 2025) generates the description from the audio. The captioner

does not receive the retrieval criterion; it uses the full audio, temperature 0, seed 42, and a maximum of 1,024 generated tokens, with inference through a vLLM backend.

Both queries and candidates concatenate the transcription and description into the following text:

```
Transcription: {transcription} Caption: {caption}
```

The query side further prepends the official instruction:

```
Instruct: {instruction}
Query: Transcription: {transcription} Caption: {caption}
```

The candidate side directly uses the concatenated text without an instruction prefix. Qwen3-Embedding-8B has a text input limit of 8,192 tokens; the hidden state of the last valid token is used as the text representation, and retrieval is performed by cosine similarity. Both tasks share the same candidate text index.

D.4 Full Retrieval Results

Table 15 summarizes Recall@10, Recall@50, and Recall@100 for the two tasks in the main text. JSRL uses the same final checkpoint as in the main text; all values are computed by within-task query averaging as described above.

Table 15: **Full Recall on INSPIRE Synthetic (%).** R@K denotes Recall@K. Speaker and Style denote same-speaker and same-speaking-style respectively.

Model	Speaker			Style		
	R@10 ↑	R@50 ↑	R@100 ↑	R@10 ↑	R@50 ↑	R@100 ↑
Random (expected)	0.23	1.16	2.31	0.28	1.39	2.78
WavLM-Large	0.25	1.00	2.50	4.75	8.50	12.75
LAION-CLAP	0.50	1.50	2.25	1.00	3.75	6.00
TP-CLAP	0.50	3.25	5.00	0.50	2.00	3.25
WAVE	0.00	0.75	2.50	0.00	0.50	2.25
ALM2Vec-PT	1.00	2.25	4.25	0.50	2.00	3.75
ALM2Vec-FT	0.25	2.00	3.25	1.00	2.50	4.50
Qwen3-Embedding-8B cascade	0.00	1.00	3.00	0.25	2.00	3.00
JSRL	**11.75**	**20.00**	**25.25**	**10.75**	**17.50**	**22.25**

E Supplementary Experimental Results

This section supplements candidate pool size sensitivity, per-corpus attribute competition results, and training stage and ablation results. Evaluation panels and metric definitions are given in Appendix A; baseline interface implementations are given in Appendix C.

E.1 Candidate Pool Size Sensitivity

We enlarge the structured candidate pool from 16 to 32 and 64 on S–E and E–C, maintaining the ratio $\mathcal{D}_{11} : \mathcal{D}_{10} : \mathcal{D}_{01} : \mathcal{D}_{00} = 1 : 3 : 3 : 9$. For each corpus and criterion pair, a common query set that can support 64 candidates is used, retaining the original ten 16-candidate samplings and expanding each pool through nested sampling without replacement within each joint state. All models and both criterion directions share the candidate sets. The common query counts for CREMA-D, MSP-IMPROV, and RAVDESS are 2,485/3,097, 455/457, and 1,342/1,438 (S–E/E–C). All pool sizes reuse fixed checkpoints and their scores; S/E/C mAP is aggregated following Appendix A.7.

Table 16: **Candidate pool size sensitivity on common query sets.** mAP in %; Δ is the gain of JSRL over WAVE (Dynamic) in percentage points, computed from unrounded values. JSRL uses the Fixed interface.

Pool size	JSRL mAP	WAVE (Dynamic) mAP	Δ mAP
16	**75.71**	71.65	+4.06
32	**73.21**	69.06	+4.15
64	**71.57**	67.35	+4.23

Table 16 shows that as the candidate pool grows from 16 to 64, JSRL's overall mAP advantage over WAVE (Dynamic) remains at 4.06–4.23 percentage points. The 16-candidate results on the common query set are close to the full query set results in the main text, where the common queries for S–E on CREMA-D cover 81.64% of the original 16-candidate eligible queries. In per-attribute comparisons, the overall gain mainly comes from speaker and emotion; on content, the two methods are comparable, with the gap within 1 percentage point at all three pool sizes. This experiment shows that in structured candidate pools with a fixed common query set and joint state proportions, JSRL's advantage remains stable as the pool grows, slightly increasing from 4.06 to 4.23 percentage points.

E.2 Per-Corpus Attribute Competition Results

Below we report per-corpus Conflict (C) and SwitchExact (Sw.) for Figure 3 in the main text, retaining all eleven model and interface configurations. Each corpus computes query averages separately; the main text then aggregates with equal weight over corpora.

Table 17: **Attribute competition results on CREMA-D (%).** C is Conflict; Sw. is SwitchExact.

		S–E		E–C		S–C	
Model	Interface	C ↑	Sw. ↑	C ↑	Sw. ↑	C ↑	Sw. ↑
WavLM-Large	Static	50.00	0.00	50.00	0.00	50.00	0.00
LAION-CLAP	Static	50.00	0.00	50.00	0.00	50.00	0.00
TP-CLAP	Fixed	50.63	15.98	49.28	11.42	50.59	6.10
WAVE	Fixed	53.79	12.51	68.71	38.54	70.07	40.60
WAVE	Dynamic	62.14	26.31	85.88	71.90	80.59	61.39
ALM2Vec-PT	Fixed	53.94	13.42	60.49	22.36	61.53	23.41
ALM2Vec-PT	Dynamic	64.96	31.55	71.20	42.42	72.72	45.44
ALM2Vec-FT	Fixed	51.90	8.72	56.96	13.99	52.36	4.72
ALM2Vec-FT	Dynamic	59.36	19.97	65.11	30.22	57.51	15.03
Frozen MOSS-Audio	Fixed	50.58	10.46	56.65	27.58	58.10	24.50
JSRL	Fixed	74.24	52.00	90.01	80.07	84.32	68.68

Table 18: **Attribute competition results on MSP-IMPROV (%).** C is Conflict; Sw. is SwitchExact.

		S–E		E–C		S–C	
Model	Interface	C ↑	Sw. ↑	C ↑	Sw. ↑	C ↑	Sw. ↑
WavLM-Large	Static	50.00	0.00	50.00	0.00	50.00	0.00
LAION-CLAP	Static	50.00	0.00	50.00	0.00	50.00	0.00
TP-CLAP	Fixed	50.59	15.13	50.04	13.16	49.82	5.11
WAVE	Fixed	52.79	12.27	58.03	16.40	64.82	29.80
WAVE	Dynamic	55.86	20.02	55.22	10.56	73.32	46.69
ALM2Vec-PT	Fixed	53.27	15.92	51.28	3.16	59.61	19.59
ALM2Vec-PT	Dynamic	57.57	21.13	50.97	2.04	66.37	32.76
ALM2Vec-FT	Fixed	51.34	11.54	50.28	0.59	50.97	1.95
ALM2Vec-FT	Dynamic	53.36	11.34	50.41	0.83	52.11	4.22
Frozen MOSS-Audio	Fixed	50.44	10.14	57.19	28.73	56.71	23.94
JSRL	Fixed	66.46	38.16	80.98	62.12	85.94	72.04

Table 19: **Attribute competition results on RAVDESS (%).** C is Conflict; Sw. is SwitchExact.

		S–E		E–C		S–C	
Model	Interface	C ↑	Sw. ↑	C ↑	Sw. ↑	C ↑	Sw. ↑
WavLM-Large	Static	50.00	0.00	50.00	0.00	50.00	0.00
LAION-CLAP	Static	50.00	0.00	50.00	0.00	50.00	0.00
TP-CLAP	Fixed	51.84	19.19	49.17	14.87	50.04	4.83
WAVE	Fixed	53.62	12.12	62.08	24.18	55.82	11.67
WAVE	Dynamic	60.87	23.36	82.73	65.47	69.56	39.12
ALM2Vec-PT	Fixed	55.48	15.94	57.14	15.82	60.62	21.91
ALM2Vec-PT	Dynamic	66.06	33.92	65.78	31.64	72.64	45.30
ALM2Vec-FT	Fixed	53.46	10.92	55.13	10.95	51.97	4.22
ALM2Vec-FT	Dynamic	60.03	21.53	61.34	22.69	55.74	11.48
Frozen MOSS-Audio	Fixed	50.48	10.40	53.53	17.85	54.40	17.06
JSRL	Fixed	72.65	49.93	85.04	70.22	88.90	77.90

E.3 Training Stages and Ablation Results

Table 20 summarizes the relevance switching, relevance persistence, and multi-candidate retrieval results for Stage 2 continuation, JSRL w/o persistence, and JSRL, corresponding to the comparison in Figure 4 in the main text. Table 21 supplements per-criterion-pair results from Frozen MOSS-Audio through each training stage. Training setup is given in Appendix B.4; Persistence definition and aggregation are given in Appendix A.4.

Table 20: **Change and persistence analysis in criterion-dependent ranking (%).** The first four columns are equally averaged over S–E and E–C. mAP uses the S/E/C aggregate from Table 1.

Training variant	Conflict ↑	SwitchExact ↑	P_{11} ↑	P_{00} ↑	mAP ↑
Stage 2 continuation	75.96	54.73	78.67	71.98	72.26
JSRL w/o persistence	**78.40**	**58.79**	76.51	72.42	72.76
JSRL	78.24	58.75	**79.63**	**73.70**	**75.76**

Table 21: **Supplementary results across training stages (%).** Each column is a three-corpus equal-weight macro-average. S/E/C mAP is the multi-candidate retrieval aggregate from the main text; S–C mAP is equally averaged over S and C. JSRL is the main model; Stage 2 continuation is the same-budget control; JSRL w/o persistence only supervises $\mathcal{D}_{10}/\mathcal{D}_{01}$ relevance switching.

	S–E		E–C	
Model	Conflict ↑	SwitchExact ↑	Conflict ↑	SwitchExact ↑
Frozen MOSS-Audio	50.50	10.34	55.79	24.72
Stage 1	60.04	23.64	70.12	40.52
Stage 2	67.14	39.26	79.74	59.96
Stage 2 continuation	67.97	41.40	83.94	68.06
JSRL w/o persistence	71.51	46.81	85.29	70.76
JSRL	71.12	46.70	85.35	70.80

	S/E/C	S–C	
Model	mAP ↑	SwitchExact ↑	mAP ↑
Frozen MOSS-Audio	48.51	21.83	53.90
Stage 1	68.53	39.80	72.72
Stage 2	71.87	59.67	77.07
Stage 2 continuation	72.26	67.36	77.19
JSRL w/o persistence	72.76	72.93	76.22
JSRL	75.76	72.88	81.76

F Supplementary Analysis on Criterion Response and Ranking Control

This section further distinguishes the score changes brought by criterion switching from actual ranking reversal. The evaluation uses the same queries and candidate sets as the main text's attribute competition evaluation (Appendix A.2); all statistics are first averaged over conflicting candidate pairs within each query, then aggregated with equal weight over queries and the three corpora.

F.1 Reversal Results and Failure Types

For the same query and a pair of candidates $d_{10} \in \mathcal{D}_{10}$ and $d_{01} \in \mathcal{D}_{01}$, let m_f and m_g denote the standardized score margin $s_c(q, d_{10}) - s_c(q, d_{01})$ under $c = f$ and $c = g$ respectively. For each query and criterion, the candidate scores are normalized by their standard deviation over $\mathcal{D}_{10} \cup \mathcal{D}_{01}$. Define

$$\beta = \frac{m_f + m_g}{2}, \qquad \delta = \frac{m_f - m_g}{2}. \tag{26}$$

Here, β is the score margin component shared across the two criteria, and δ is the component that changes sign with the criterion switch, i.e., $m_f = \beta + \delta$ and $m_g = \beta - \delta$. Correct reversal requires $m_f > 0$ and $m_g < 0$, equivalent to $\delta > |\beta|$. When $0 < \delta \leq |\beta|$, the score changes in the target direction but is not yet sufficient for correct reversal. The remaining candidate pairs are classified as other failures.

Figure 6 compares the three outcome types for WAVE, ALM2Vec-PT, and JSRL; the correct reversal proportion corresponds to SwitchExact in the main text. A substantial fraction of candidate pairs for WAVE (Dynamic) and ALM2Vec-PT (Dynamic) have already changed in the correct direction but failed to complete the reversal, indicating that responding to the criterion is not the same as letting the criterion determine the candidate order. JSRL has the highest correct reversal proportion on both criterion pairs.

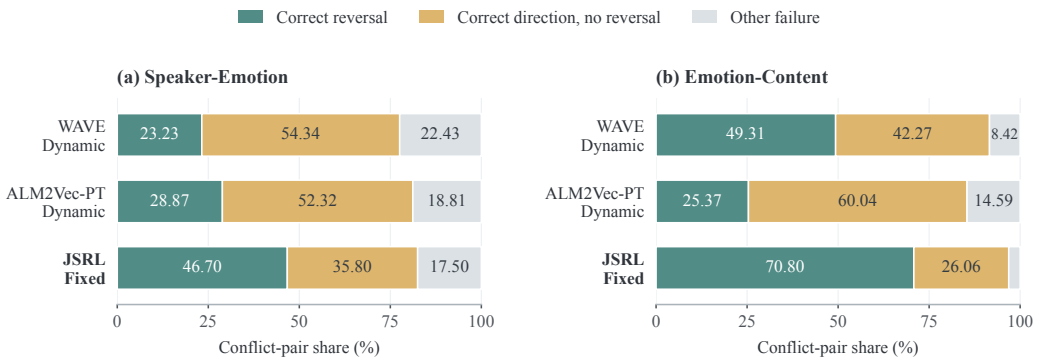

Figure 6: **Reversal results and failure types after criterion switching.** WAVE and ALM2Vec-PT use the Dynamic interface; JSRL uses the Fixed interface. The three categories are: correct reversal, correct direction but not reversed, and other failures. Proportions are aggregated with equal weight over queries and corpora.