FROM LANGUAGE REASONING TO CROSS-MODAL SENTIMENT EVIDENCE:

RECURSIVE HARNESS SEARCH FOR MULTIMODAL SENTIMENT RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

The performance of multimodal sentiment analysis depends not only on the model itself, but also on how information is organized and on how the model is coordinated with external reasoning. Recursive self-improvement (RSI) provides an automated route to this by iteratively searching, executing, and modifying executable programs around a base model. However, existing RSI work has focused almost exclusively on text-based reasoning tasks. Unlike purely textual settings, sentiment cues in multimodal sentiment analysis are distributed across text, audio, and visual modalities, where they may complement or conflict with one another and vary from sample to sample. Consequently, the acquisition and utilization of sentiment cues must be searched jointly with the reasoning strategy and carried by the same executable program. To operationalize this view, we present Senti-Harness, a recursive Harness search framework that jointly edits sentiment-cue acquisition, the organization of modality-specific evidence across model calls, cross-modal evidence use, and reasoning within one executable program. With the inner-loop model fixed, a matched CH-SIMS run improves the validation frontier from 81.14% under Fixed-Exposure Reasoning Search to 89.0% when multimodal evidence handling is made searchable. We further find that Harness effectiveness depends strongly on the state of the inner-loop model: after Plain SFT, the marginal benefit of the same Harness can shrink substantially or even reverse, revealing a concrete model–Harness interaction. Motivated by this coupling, we further explore Harness-aware SFT, which improves retained-Harness Acc-2 from 88.18% with Plain SFT to 90.37% under a matched-budget CH-SIMS training setting. These results identify executable multimodal evidence handling and model–Harness interaction as two central dimensions of recursive Harness optimization for multimodal sentiment analysis.

1 INTRODUCTION

Multimodal sentiment analysis requires combining evidence whose meaning depends on both its source and its relation to other observations. Lexical content, vocal delivery, and facial behavior may agree, conflict, or provide little useful information, and their contribution varies across samples. This challenge has motivated modality-specific representation learning and adaptive fusion methods in multimodal sentiment analysis (MSA) (Yu et al., 2021; Feng et al., 2024).

Native multimodal foundation models change the surrounding design problem. A single model can process text, audio, and visual inputs (Xu et al., 2025b), but an external program still determines which materials are exposed to the model, how modality-specific evidence is elicited, how intermediate outputs are represented, and how those outputs are converted into a final prediction. We call this executable program a *Harness*. In multimodal sentiment analysis, Harness design therefore controls what sentiment cues are exposed to the model and how they are organized, not only the reasoning instructions.

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.

Recent work on prompt, program, and Harness optimization uses execution feedback to revise the computation surrounding language models (Agrawal et al., 2025; Lee et al., 2026). These methods demonstrate that model-external behavior can be optimized without changing the inner-loop model parameters. Their primary applications, however, are text-centered. Directly transferring text-domain recursive search leaves the acoustic and visual evidence outside the editable program space.

This paper studies three questions. **RQ1** asks whether recursive search should be restricted to reasoning over a fixed multimodal exposure, or should also optimize the acquisition, organization, and use of multimodal sentiment evidence. **RQ2** asks how the marginal utility of a fixed Harness changes when the inner-loop model state changes. **RQ3** then explores Harness-conditioned model adaptation as a practical response to the interaction observed in RQ2. We answer these questions with SentiHarness, which makes multimodal evidence handling and decision logic jointly editable in one executable search space, evaluates candidate Harnesses through real execution, and recursively revises them from accumulated feedback.

Our contributions are threefold. First, we formulate multimodal Harness search over sentiment-cue acquisition, modality organization, cross-modal evidence use, and decision logic. In a matched CH-SIMS comparison, full SentiHarness raises the validation frontier from 81.14% under Fixed-Exposure Reasoning Search to 89.0%; across three independent proposer-search trajectories, the validation-selected full-search Harnesses also lead on held-out test Acc-2 by 5.9, 4.6, and 5.9 points. Second, we characterize model–Harness interaction with a fixed Harness family: Plain SFT changes the marginal gain of H_{15} from +4.59 to -1.53 points and contracts the gains of H_{82} and H_{85} from +5.91/+5.47 to +0.88/+1.97 points. Third, we examine Harness-aware SFT through the first-stage interface retained by H_{85}; one matched-budget CH-SIMS training realization yields 90.37% retained-Harness Acc-2 versus 88.18% for Plain SFT.

2 Related Work

Task-specific multimodal sentiment methods learn modality representations and fusion rules (Yu et al., 2021; Feng et al., 2024; Jiang et al., 2026); multimodal program systems coordinate predefined tools and modules (Gupta & Kembhavi, 2023; Surís et al., 2023). SentiHarness instead searches the executable acquisition, organization, and use of sentiment cues together with reasoning.

Prompt and program optimizers use task feedback to improve model-external computation (Khattab et al., 2023; Agrawal et al., 2025), while recursive Harness systems revise executable programs from accumulated execution histories (Lee et al., 2026; Zhang et al., 2025). Our study examines both the multimodal search space and its dependence on the inner-loop model state. Work coupling model updates with external structures (Tiwari et al., 2026; Chen et al., 2026a; Kim et al., 2026) further motivates training through a retained Harness interface. Appendix A provides the extended comparison.

3 SentiHarness

Let $x = (x^{\mathrm{t}}, x^{\mathrm{a}}, x^{\mathrm{v}})$ denote textual, acoustic, and visual observations and let y denote the target label. An inner-loop model M_θ is accessed only through an executable Harness H:

$$(\hat{y}, \tau) = \mathrm{Exec}(H, M_\theta, x), \tag{1}$$

where τ records model requests, intermediate responses, and program decisions. Let $P_{\mathcal{D}}(H, M_\theta)$ denote performance on dataset $\mathcal{D}$. Under search budget B, the completed candidate set $\mathcal{H}_B$ is selected by validation performance,

$$H^\star \in \arg\max_{H \in \mathcal{H}_B} P_{\mathcal{D}_{\mathrm{val}}}(H, M_\theta). \tag{2}$$

During search, the inner-loop model is fixed. The proposer receives training-split execution traces and aggregate training and validation scores, but never validation examples or validation traces; the test split remains held out until search and validation-based selection are complete.

Table 1: Editable decisions in Fixed-Exposure Reasoning Search and full multimodal Harness search.

Decision	Fixed-Exposure	SentiHarness
Reasoning prompt / output schema	✓	✓
Post-hoc decision logic	✓	✓
T/A/V materials exposed to the model	fixed	editable
Number / order of model calls	fixed (one)	editable
Modality-specific request construction		✓
Conditional evidence acquisition		✓
Cross-call evidence organization		✓

Each round proposes three candidates. The controller retains the highest-validation Harness found so far and returns it after B rounds. Appendix J defines the round-best and maintained-frontier trajectories used in the figures.

3.1 Searchable Multimodal Harness

We use *Fixed-Exposure Reasoning Search* as the controlled RQ1 baseline. Every candidate receives the same transcript, full utterance audio, and full video through exactly one joint T/A/V model call. Search may revise the reasoning prompt, serialized output schema, and deterministic decision rule, but it cannot alter the exposed media, separate modalities into additional requests, change the number or order of model calls, or conditionally reacquire evidence. SentiHarness removes these restrictions and makes multimodal evidence acquisition, organization across calls, and cross-modal integration editable. Table 1 summarizes this controlled difference, and Appendix D gives the implementation-level specification.

3.2 Recursive Search from Execution Feedback

The multimodal reference H_0 is a strong single-call T/A/V Harness whose reasoning prompt follows the CoT design of C2F-Thinker (Luo et al., 2026). At each round, the proposer uses the accumulated candidate programs, aggregate train/validation scores, and training-split traces to generate executable revisions for the fixed inner-loop model. Candidates are then executed and appended to the search history, allowing later programs to revise earlier request construction, response schemas, evidence gates, thresholds, or decision rules instead of restarting independently. The final Harness is selected by Eq. 2.

A concrete searched evolution. Search-1 illustrates what the broader search space changes in practice. The reference H_0 makes one fused T/A/V request and directly uses the model-written score. Frontier revisions first introduce structured modality evidence and programmatic gating, then add conditional text-free audio–visual verification for unresolved cases, and later tighten the rule under which that evidence can change the prediction. The final H_{85} retains a structured first stage and invokes its additional delivery-pair check only for unresolved zero/flat cases. This lineage changes not only how the model reasons about a fixed observation, but also what evidence is exposed independently, when additional evidence is requested, and how multiple reads are composed.

3.3 Harness-aware Model Adaptation

A Harness defines a model-facing interface through its request format and the structured response schema consumed by downstream program logic. Harness-aware SFT adapts the inner-loop model through such a Harness-compatible interface rather than a standalone task interface. In the controlled CH-SIMS instantiation, we reproduce the first-stage interface inherited by the final H_{85} Harness; this stage corresponds to the H_{15} request–response format. Training targets are constructed deterministically from the original overall sentiment annotation and transcript, without executing the Base model or any external teacher. Appendix G gives the exact target construction, serialization, and loss mask.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and metrics. We evaluate multimodal sentiment recognition on CH-SIMS (Yu et al., 2020) and CMU-MOSI (Zadeh et al., 2016), and use MELD (Poria et al., 2019) as a related seven-class multimodal emotion-recognition benchmark. CH-SIMS reports Acc-2, support-weighted binary F1, and Acc-3; CMU-MOSI reports Non0 Acc-2 and support-weighted binary F1; MELD reports weighted F1. Validation Acc-2, Non0 Acc-2, and weighted F1 select the Harness on CH-SIMS, CMU-MOSI, and MELD, respectively. Acc-3 is used only for final CH-SIMS reporting.

Search protocol. The frozen inner-loop model is Qwen3-Omni-30B-A3B-Instruct and the proposer is `deepseek-v4-flash`. Search uses up to 50 rounds with three candidates per round and the full official splits. Training traces inform proposals, validation scores select the maintained frontier, and test data remain held out until selection is complete. Search-1/2/3 denote independent proposer trajectories with fixed inner-loop decoding and splits. Appendix J gives the execution and evaluation details.

Model states and adaptation. We evaluate Base, Plain SFT, and Harness-aware SFT with the fixed CH-SIMS family H_0, H_{15}, H_{82}, and H_{85}. Here H_k indexes evaluated candidates, not proposal rounds. Harness-aware SFT uses the H_{15} request–response interface retained as the first stage of H_{85}; the conditional second stage remains external to training. Both SFT variants use the same 1,368 training examples, 91 updates, effective batch size 4, cutoff 8192, and learning rate 10^{-4}. Harness-aware targets derive from the overall gold label and transcript, with voice/face verdicts fixed to `none`. Appendix G specifies targets and loss masks; Appendix H documents packing and budget accounting.

For the interaction study, H_0 is the manually constructed single-call T/A/V Harness following the CoT design of C2F-Thinker. H_{15}, H_{82}, and H_{85} are evaluated candidates from the same CH-SIMS search lineage. H_{15} is used because its structured text/voice/face request–response contract is the first stage inherited by H_{85}; it is not an independently selected training champion. H_{82} adds text-free voice/face verification with an agreement gate, while H_{85} executes the inherited first stage and conditionally issues a delivery-pair query for unresolved zero/flat cases. This fixed family lets us measure model-state effects without changing the searched program.

The Harness-aware conversion writes one training record per CH-SIMS training example. Each request contains the transcript, full utterance audio, full video, and the 2×2 visual mosaic expected by the retained first stage. The assistant response is deterministically serialized from the overall gold sentiment label and transcript. It contains no Base-model trajectory, teacher rationale, modality pseudo-label, audio/visual gold label, or delivery-pair target; voice and face verdicts are fixed to `none`. The contrast therefore changes the model-facing interface and structured response contract while retaining the same supervised sentiment target.

The CH-SIMS split contains 1,368 examples, yielding 1,368 Harness-aware records. With an 8192-token cutoff, these records pack into 365 sequences with average packed length about 7,413 tokens; the shorter Plain-SFT records form 115 packed sequences. Both runs use 91 optimization steps, effective batch size 4, cutoff 8192, and learning rate 10^{-4}, corresponding to approximately 3.0M nominal token slots. Plain SFT resamples its packed sequences to fill the same update budget. The exact response serialization and response-only loss mask are given in Appendix G.

Table 2: Main results on CH-SIMS (%). The Qwen3-Omni + H_0 row uses the strong single-call T/A/V initialization described in Section 4.1, rather than bare direct inference. Dashes indicate metrics not retained for the matched Plain-SFT checkpoint in this audit.

Method	Acc-2	F1	Acc-3
Task-specific MSA			
Self-MM (Yu et al., AAAI'21)	80.04	80.44	65.47
KuDA (Feng et al., Findings EMNLP'24)	80.74	80.71	66.52
CACR (Wang et al., CVPR'26)	81.62	80.99	–
ECA (Han et al., PR'26)	85.65	85.90	76.54
CICA (Jiang et al., CVPR'26)	86.00	85.68	76.37
MLLM-based MSA			
HumanOmni (Zhao et al., arXiv'25)	85.10	85.00	72.90
Qwen2.5-Omni (Xu et al., arXiv'25)	87.50	87.00	73.30
SentiLLM (Chen et al., ACM MM'26)	80.96	80.22	67.18
C2F-Thinker† (Luo et al., arXiv'26)	87.50	87.80	**77.20**
Ours			
Qwen3-Omni + H_0	81.84	82.37	72.21
Qwen3-Omni + Searched Harness	87.31	87.01	73.96
Plain SFT + Searched Harness	88.18	–	–
Harness-aware SFT + Searched Harness	**90.37**	**90.17**	76.15

† Comparison note: C2F-Thinker uses teacher-distilled chain-of-thought supervision on the original training samples.

Table 3: Main results on CMU-MOSI under the standard Non0 binary protocol (%).

Method	Non0 Acc-2	F1
Task-specific MSA		
Self-MM (Yu et al., AAAI'21)	85.46	85.43
KuDA (Feng et al., Findings EMNLP'24)	86.43	86.46
EBMC (He et al., CVPR'26)	87.84	87.79
CICA (Jiang et al., CVPR'26)	**90.24**	90.23
LLM-based MSA		
MSE-Adapter† (Yang et al., AAAI'25)	88.63	88.54
MGSI (Lin et al., arXiv'26)	89.60	89.57
MEERA (Niu et al., PR'26)	88.96	88.37
Ours		
Qwen3-Omni	85.67	85.50
+ Searched Harness	**90.24**	90.24
Harness-aware SFT + Searched Harness	**90.24**	**90.27**

† Comparison note: the CMU-MOSI result of MSE-Adapter is reproduced by Lin et al. (2026) using the ChatGLM3-6B backbone under the same evaluation setting.

4.2 Main Results

Tables 2–4 report recognition performance. With the inner-loop model frozen, searched Harnesses improve CH-SIMS Acc-2 from 81.84 to 87.31, CMU-MOSI Non0 Acc-2 from 85.67 to 90.24, and MELD WF1 from 61.04 to 66.31. The next experiments examine the search-space contribution, changes in Harness utility across model states, and adaptation through a retained Harness interface.

4.3 RQ1: What Should Be Searchable in Multimodal Sentiment Analysis?

We compare full search with Fixed-Exposure Reasoning Search using the same Base model, proposer, validation split, and 50-round budget. The control exposes T/A/V through one joint call and searches its reasoning prompt, output schema, and decision rule; full search additionally edits sentiment-cue acquisition and cross-call organization.

Table 4: Representative comparisons on MELD using weighted F1 (WF1, %).

Method	**WF1**
Task-specific MER	
CFN-ESA (Li et al., TAFFC'24)	66.70
Cross-modal Gated Enhancement (Zhao et al., Sci. Rep.'25)	66.20
TIAWFM (Lu et al., Speech Commun.'26)	66.83
DiGemo (Trang et al., Inf. Sci.'26)	65.92
LLM-based MER	
MSE-Adapter (Yang et al., AAAI'25)	65.13
PaP (Zhao et al., CVPR'26)	65.97
Ours	
Qwen3-Omni	61.04
+ Searched Harness	66.31
Harness-aware SFT + Searched Harness	**66.81**

Table 5: RQ1: held-out CH-SIMS test Acc-2 (%) across three proposer-search trajectories; all Harnesses are selected on validation.

Trajectory	Fixed-Exposure	Full search	Difference
1	81.4	87.3	+5.9
2	81.2	85.8	+4.6
3	79.2	85.1	+5.9

Across all three proposer trajectories, validation-selected full-search Harnesses lead on held-out test Acc-2 by 5.9, 4.6, and 5.9 points (Table 5). Figure 1 illustrates the matched Search-2 trajectory: the validation frontier reaches 89.0%, versus 81.14% (370/456) for Fixed-Exposure RS-2. The repeated held-out advantage supports searching evidence handling beyond reasoning over a fixed joint exposure. The control is designed to hold multimodal exposure constant while changing what the outer-loop search can edit. Both arms receive the same transcript, full audio, and full video; the difference is that the control must compress all three into one model response, whereas full search can request modality-specific evidence, serialize intermediate observations, and apply conditional follow-up. The held-out table therefore tests whether this broader program space reaches a more reliable frontier across proposer trajectories. It does not identify a single winning primitive: the selected programs use different combinations of evidence reads, gates, and decision rules. The repeated separation is consequently interpreted at the level of the editable search space.

The selected H_{85} uses its additional evidence check on 19 of 457 test examples (1.04 calls per example). This illustrates selective follow-up within the broader search space. Appendix B analyzes the searched programs. The same audit also clarifies what is and is not being compared. Search-1, Search-2, and Search-3 use the same proposal-round budget and evaluation protocol, but their executed calls can differ because a Harness may branch or request additional evidence. Thus the result is a comparison of editable search spaces under matched proposal opportunities, rather than a claim of equal per-candidate compute. The H_{85} trace is useful precisely because most examples terminate after the first stage and only a small subset receives the delivery-pair query. That behavior is consistent with a selective evidence policy: the program spends extra interaction on ambiguous cases instead of forcing every example through the same sequence. Appendix B analyzes the searched programs.

4.4 RQ2: Is Harness Utility Independent of the Inner-loop Model State?

We measure marginal Harness utility as $\Delta(H, M) = P(H, M) - P(H_0, M)$. After Plain SFT, H_{15} changes from +4.59 to -1.53 points, H_{82} from +5.91 to +0.88, and H_{85} from +5.47 to +1.97. The contractions for H_{15} and H_{82} remain supported after Holm correction (Appendix E). This concerns additional Harness benefit: H_{85}'s absolute score rises from 87.31 to 88.18 despite its smaller gain over H_0 (Appendix F). The absolute-score matrix separates two quantities that can move in opposite directions. Plain SFT raises H_{85} from 87.31 to 88.18, yet reduces its gain over H_0 because the

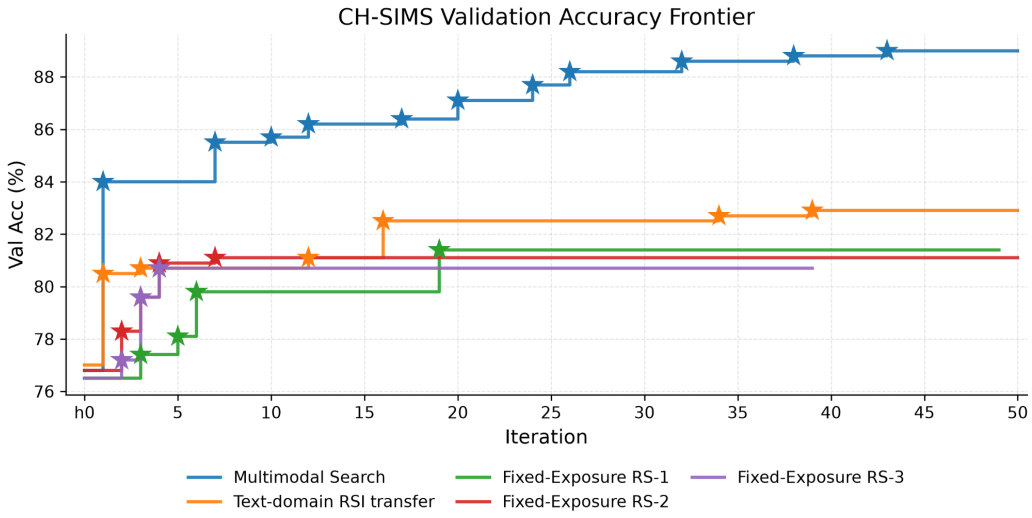


Figure 1: CH-SIMS validation frontiers: full Search-2, text-domain transfer, and three Fixed-Exposure repeats. Search-2 versus RS-2 illustrates the search-space contrast; stars mark frontier updates.

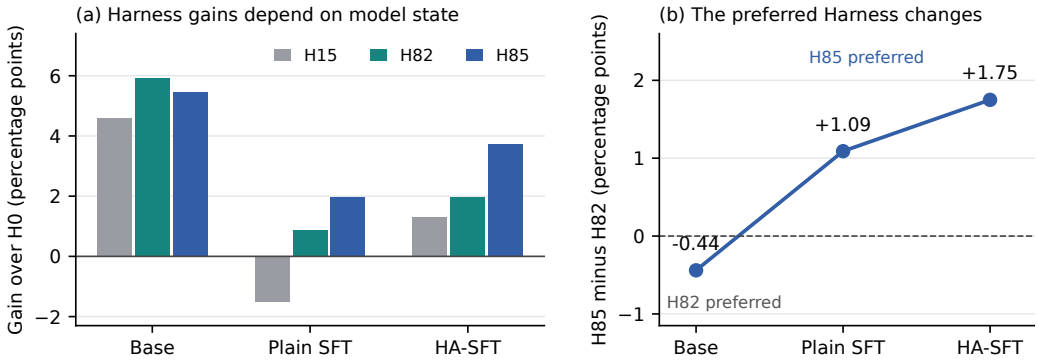


Figure 2: Point-estimate Harness gains across inner-loop model states on CH-SIMS, measured against H_0 within each state. Plain SFT changes H_{15} from beneficial to harmful and contracts the gains of H_{82} and H_{85}; uncertainty for the contraction tests is reported in Appendix E.

standalone model also improves. For H_{15}, the same training raises the reference state while making the fixed structured Harness harmful relative to that reference. This is why the paper reports both $P(H, M)$ and $\Delta(H, M)$: the former measures composed-system quality, while the latter measures the additional value supplied by the external program. Harness selection after adaptation should consider both quantities.

A related system-level pattern appears on the other benchmarks: relative to Base + Searched Harness, Plain SFT reduces the composed-system score by 3.35 points on CMU-MOSI and 2.22 points on MELD. Together with the controlled CH-SIMS matrix, this shows that Harness utility cannot be treated as a property of the Harness alone.

4.5 RQ3: An Exploratory Study of Harness-Conditioned Adaptation

RQ2 motivates evaluating adaptation through the interface retained inside the final Harness. Under the matched CH-SIMS configuration in Section 4.1, Harness-aware SFT trains the H_{15}-compatible first stage of H_{85}, while Plain SFT uses a standalone task interface.

Table 6: Retained-Harness performance under different inner-loop model states (%). The final column indicates whether Plain SFT and Harness-aware SFT use matched optimization budgets.

Dataset	Base + Harness	Plain SFT + Harness	HA-SFT + Harness	Plain/HA budget
CH-SIMS (Acc-2)	87.31	88.18	**90.37**	matched
CMU-MOSI (Acc-2)	**90.24**	86.89	**90.24**	unmatched
MELD (WF1)	66.31	64.09	**66.81**	unmatched

Harness-aware SFT + H_{85} reaches 90.37 Acc-2 versus 88.18 for Plain SFT (+2.19 points). With H_0, the corresponding improvement is +0.44, yielding a +1.75-point increase in marginal H_{85} utility. The latter is directionally consistent with interface adaptation but its confidence interval crosses zero. The experiment therefore establishes the observed composed-system improvement for these checkpoints; Appendix E reports uncertainty, and Appendix H details the other datasets' configurations. The differential between H_0 and H_{85} is the key comparison: adaptation improves the fixed reference by 0.44 points but improves the retained Harness by 2.19 points. This pattern is compatible with interface-conditioned adaptation because the training records expose the response contract used by the first stage, while the external second stage remains unchanged. It does not by itself identify which serialization field or branch caused the gain, and the uncertainty analysis is therefore part of the interpretation rather than a post-hoc qualification. The result supports evaluating adaptation at the composed-system level, where the model state and executable Harness are jointly measured.

The three dataset rows also clarify the scope of this comparison. On CH-SIMS, the adaptation interface is matched to the retained first stage and both training variants use the same optimization budget; this is the setting in which the largest controlled gain is observed. On CMU-MOSI and MELD, the reported Plain-SFT and Harness-aware SFT runs do not have identical packed-token budgets, so their rows are descriptive cross-dataset checks rather than matched adaptation tests. Even so, the direction of the composed-system comparison is informative: Harness-aware SFT recovers the Base + Harness score on CMU-MOSI and improves the Plain-SFT + Harness score on MELD. These rows do not establish that the same training recipe transfers unchanged across benchmarks. They show why the retained execution interface is a useful unit of analysis when the inner-loop state is updated.

The adaptation records are deliberately weaker than demonstrations that expose hidden reasoning traces. Each record supplies the original multimodal request and a deterministic assistant response, with the response containing the transcript-derived overall label and fixed `none` voice/face fields. This design prevents the training target from directly teaching a modality-specific verdict or copying a Base-model trajectory. The learned behavior is therefore evaluated through the request–response contract that the external Harness can consume. The external branch still performs its own evidence request at inference time, which keeps the first-stage training intervention separate from the conditional second-stage policy.

Taken together, the RQ2 and RQ3 results support a practical selection rule. A Harness should be compared with its reference state within each model checkpoint, because a larger absolute score can coexist with a smaller marginal gain. After adaptation, the retained Harness should be re-evaluated rather than assumed to preserve its Base-state ranking. The CH-SIMS matrix provides the controlled evidence for this rule, while the CMU-MOSI and MELD rows indicate how the same accounting can expose budget or interface mismatches in auxiliary checks. This interpretation also explains why the paper reports both absolute performance and $\Delta(H, M)$ throughout the experiments.

4.6 Search Dynamics and Validation

Recursive search is compared with independent candidate search under the same proposer, evaluator, task specification, and 50-round budget. Independent proposals do not use previous candidates or execution feedback. The best validation Acc-2 is approximately 87.7 for recursive search and 85.3 for independent search. Appendix I reports the candidate trajectories and repeated-search structures.

This search-dynamics check complements the held-out RQ1 comparison. The recursive procedure retains execution feedback and the current best candidate when proposing the next round, so later

proposals can revise an observed failure mode instead of restarting from an unrelated prompt. Independent search removes that feedback while keeping the proposer, evaluator, task specification, and round budget fixed. The resulting validation gap is therefore evidence about feedback reuse in the outer loop; it is not a separate estimate of the multimodal evidence-acquisition effect measured by the Fixed-Exposure control. In particular, the RQ1 table compares the editable program spaces, whereas this check compares how proposals are generated within the broader space.

The experiments also separate selection from adaptation. Harnesses are selected on validation before the held-out test rows are reported, and the same selected family is then evaluated across Base, Plain SFT, and Harness-aware SFT states. This ordering prevents the adaptation comparison from using test performance to choose a Harness. It also makes the model-state analysis interpretable: changes in $\Delta(H, M)$ arise from evaluating a fixed executable program against different model states, while the RQ3 intervention changes the model-facing interface during training. The two operations answer different questions and are reported in separate tables.

The results connect two design decisions. First, sentiment-cue acquisition and use benefit from being editable parts of the Harness: separate reads, structured evidence, and conditional checks can be revised through execution feedback. Second, the additional benefit of those programs changes with the inner-loop model state, motivating re-evaluation after training. Harness-aware SFT tests one response by adapting through the retained execution interface. Its CH-SIMS system improvement motivates further study of how request format, response serialization, and branch behavior contribute to compatibility.

5 Conclusion

SentiHarness jointly searches sentiment-cue acquisition, organization, use, and reasoning. Three held-out CH-SIMS comparisons favor this broader space over Fixed-Exposure Reasoning Search. A fixed Harness family further reveals that marginal utility changes across inner-loop model states. Harness-aware SFT improves retained-Harness performance in the matched CH-SIMS experiment, providing an initial adaptation strategy motivated by that interaction.

Limitations

RQ1 identifies the effect of a broader editable search space, not the separate contributions of call topology, modality separation, conditional acquisition, or cross-call organization. Proposal rounds are matched; total inference cost is not fully equalized.

The model–Harness interaction analysis is concentrated on CH-SIMS and one primary inner-loop model family. The observed ranking changes between Harnesses remain descriptive.

RQ3 has one matched CH-SIMS training realization and cannot establish training-run stability or improved audio/visual evidence recognition: its targets contain no modality-specific supervision. CMU-MOSI and MELD use different training budgets, and searched Harnesses can absorb dataset-specific annotation and cue priors (Appendix L).

AI Use Disclosure

We used generative AI tools to support hypothesis and methodology refinement, experimental design feedback, implementation and debugging, literature discovery and summarization, interpretation of experimental results, translation, and manuscript drafting, editing, and formatting. Generative AI is also used as an explicit component of the proposed method: `deepseek-v4-flash` serves as the proposer in the recursive Harness search described in the main text. All AI-assisted code and experimental outputs were executed, checked, and interpreted by the authors; literature and reference claims were verified against source materials; and all manuscript text and scientific claims were reviewed by the authors. The authors take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025. URL https://arxiv.org/abs/2507.19457.

Mingju Chen, Can Lv, Guibin Zhang, Heng Chang, and Shiji Zhou. Harnessforge: Joint harness and policy evolution for adaptive agent systems. *arXiv preprint arXiv:2606.01779*, 2026a. URL https://arxiv.org/abs/2606.01779.

Wei Chen, Junkai Li, Tongguan Wang, Hui Liu, Feiyue Xue, Chuanxiang Ma, and Ying Sha. Semantic-aligned structural abstraction for multimodal sentiment analysis. *arXiv preprint arXiv:2607.27790*, 2026b.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Xinyu Feng, Yuming Lin, Lihua He, You Li, Liang Chang, and Ya Zhou. Knowledge-guided dynamic modality attention fusion framework for multimodal sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14755–14766. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.865. URL https://aclanthology.org/2024.findings-emnlp.865/.

Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. AssistGPT: A general multi-modal assistant that can plan, execute, inspect, and learn. *arXiv preprint arXiv:2306.08640*, 2023.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*, 2023.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14953–14962, 2023.

Tingting Han, Lingyun Yu, Min Tan, Zhou Yu, and Hongxun Yao. Emotional conflict adaptation for multimodal sentiment analysis. *Pattern Recognition*, 177:113315, 2026. doi: 10.1016/j.patcog.2026.113315.

Kang He, Yuzhe Ding, Xinrong Wang, Fei Li, Chong Teng, and Donghong Ji. Enhance-then-balance modality collaboration for robust multimodal sentiment analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30183–30193, 2026.

Haoyu Jiang, Xiaoliang Chen, Duoqian Miao, Xiaolin Qin, Xianyong Li, and Yajun Du. CICA: Coupling confidence-aware pretraining with confidence-informed attention for robust multimodal sentiment analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37842–37851, 2026.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Haechan Kim, Yoonho Lee, Gisang Lee, Chelsea Finn, and Kangwook Lee. WHALE: A simple recipe for joint harness-weight optimization. *arXiv preprint arXiv:2609.00196*, 2026. URL https://arxiv.org/abs/2609.00196.

Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. Meta-harness: End-to-end optimization of model harnesses. *arXiv preprint arXiv:2603.28052*, 2026. URL https://arxiv.org/abs/2603.28052.

Jiang Li, Xiaoping Wang, Yingjian Liu, and Zhigang Zeng. CFN-ESA: A cross-modal fusion network with emotion-shift awareness for dialogue emotion recognition. *IEEE Transactions on Affective Computing*, 15(4):1919–1933, 2024. doi: 10.1109/TAFFC.2024.3389453.

Shanshan Lin, Yuesheng Wu, Chao Chen, Yizhe Yang, Zhihao Chen, Zexian Yang, and Xiangwen Liao. Multi-granularity sentiment integration for llm-based multimodal sentiment analysis. *arXiv preprint arXiv:2608.16201*, 2026. URL https://arxiv.org/abs/2608.16201.

Weijie Lu, Yunfeng Xu, and Jintan Gu. Adaptive weighting in a transformer framework for multimodal emotion recognition. *Speech Communication*, 176:103332, 2026. doi: 10.1016/j.specom.2025.103332.

Miaosen Luo, Jiesen Long, Zequn Li, Yunying Yang, Yuncheng Jiang, and Sijie Mai. Multimodal large language models for end-to-end affective computing: Benchmarking and boosting with generative knowledge prompting. *arXiv preprint arXiv:2508.02429*, 2025.

Miaosen Luo, Zhenhao Yang, Jieshen Long, Jinghu Sun, Yichu Liu, and Sijie Mai. C2F-Thinker: Coarse-to-fine reasoning with hint-guided reinforcement learning for multimodal sentiment analysis. *arXiv preprint arXiv:2604.00013*, 2026. doi: 10.48550/arXiv.2604.00013.

Fuqiang Niu, Yong Liao, Xiaojiang Peng, Hu Huang, and Bowen Zhang. MEERA: Multimodal experts for evidence routing and sparse llm adaptation in sentiment analysis. *Pattern Recognition*, 179:113922, 2026. doi: 10.1016/j.patcog.2026.113922.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Krista Opsahl-Ong, Michael J. Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.525.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 527–536. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1050. URL https://aclanthology.org/P19-1050/.

Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, Xing Zhou, Dongrui Liu, Ling Yang, Yue Wu, Kaixuan Huang, Shilong Liu, Hongru Wang, and Mengdi Wang. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11888–11898, 2023. doi: 10.1109/ICCV51070.2023.01092.

Rishabh Tiwari, Kusha Sareen, Lakshya A. Agrawal, Joseph E. Gonzalez, Matei Zaharia, Kurt Keutzer, Inderjit S. Dhillon, Rishabh Agarwal, and Devvrit Khatri. Learning, fast and slow: Towards llms that adapt continually. *arXiv preprint arXiv:2605.12484*, 2026. URL https://arxiv.org/abs/2605.12484.

Hoang Khang Trang, Bao Quoc Nguyen, Viet Vinh Khanh Dinh, Nhut Minh Nguyen, Nhat Truong Pham, Phuong-Nam Tran, Phuong Luu Vo, and Duc Ngoc Minh Dang. Towards effective multimodal emotion recognition in conversations via joint multi-graph learning and self-distillation. *Information Sciences*, pp. 124138, 2026. doi: 10.1016/j.ins.2026.124138.

Yan Wang, Fuyuan Cao, and Xingwang Zhao. Conflict-aware adaptive cross-reconstruction for multimodal sentiment analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15880–15889, 2026.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.754.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-Omni Technical Report. *arXiv preprint arXiv:2503.20215*, 2025a. doi: 10.48550/arXiv.2503.20215.

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*, 2025b. URL https://arxiv.org/abs/2509.17765.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*, 2023a.

Yang Yang, Xunde Dong, and Yupeng Qiang. MSE-Adapter: A lightweight plugin endowing llms with the capability to perform multimodal sentiment analysis and emotion recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25642–25650, 2025. doi: 10.1609/aaai.v39i24.34755.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. MM-REACT: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023b.

Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. Gödel agent: A self-referential agent framework for recursively self-improvement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27890–27913. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1354.

Wenmeng Yu, Hua Xu, Fanyang Meng, Yilin Zhu, Yixiao Ma, Jiele Wu, Jiyun Zou, and Kaicheng Yang. CH-SIMS: A chinese multimodal sentiment analysis dataset with fine-grained annotation of modality. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3718–3727. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.343. URL https://aclanthology.org/2020.acl-main.343/.

Wenmeng Yu, Hua Xu, Ziqi Yuan, and Jiele Wu. Learning modality-specific representations with self-supervised multi-task learning for multimodal sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10790–10797, 2021. doi: 10.1609/aaai.v35i12.17289.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic "differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. MOSI: Multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259*, 2016. URL https://arxiv.org/abs/1606.06259.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Hangfan Zhang, Shao Zhang, Kangcong Li, Chen Zhang, Yang Chen, Yiqun Zhang, Lei Bai, and Shuyue Hu. Self-harness: Harnesses that improve themselves. *arXiv preprint arXiv:2606.09498*, 2026.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin gödel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*, 2025.

Jiaxing Zhao, Qize Yang, Yixing Peng, Detao Bai, Shimin Yao, Boyuan Sun, Xiang Chen, Shenghao Fu, Weixuan Chen, Xihan Wei, and Liefeng Bo. HumanOmni: A large vision-speech language model for human-centric video understanding. *arXiv preprint arXiv:2501.15111*, 2025a.

Shiyun Zhao, Jinchang Ren, and Xiaojuan Zhou. Cross-modal gated feature enhancement for multimodal emotion recognition in conversations. *Scientific Reports*, 15:30004, 2025b. doi: 10.1038/s41598-025-11989-6.

Xianbing Zhao, Lan Luo, Hengyang Lu, and Buzhou Tang. Prototype-as-prompt: Multimodal sentiment prototypes endowing large language models the capability to perform multimodal sentiment analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23010–23020, 2026.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*, 2022.

A Extended Related Work

Multimodal sentiment analysis (MSA) benchmarks such as CH-SIMS and CMU-MOSI capture sentiment through textual, acoustic, and visual behavior (Yu et al., 2020; Zadeh et al., 2016). Task-specific methods improve modality-specific representations, adaptive weighting, and confidence-, conflict-, or evidence-aware fusion (Yu et al., 2021; Feng et al., 2024; Wang et al., 2026; Han et al., 2026; Jiang et al., 2026; He et al., 2026), while trimodal MLLM benchmarking shows substantial variation across model architectures and tasks (Luo et al., 2025). Programmatic multimodal systems such as VisProg, ViperGPT, MM-ReAct, and AssistGPT further show that external programs can coordinate multimodal modules and model calls (Gupta & Kembhavi, 2023; Surís et al., 2023; Yang et al., 2023b; Gao et al., 2023), but their available tools or program primitives are specified in advance. SentiHarness instead makes sentiment-cue acquisition, organization, and use themselves editable, so evidence access can be searched jointly with reasoning.

Automated optimization has progressed from prompt proposal to structured program and scaffold search. APE, OPRO, and EvoPrompt optimize instructions from model-generated candidates and task feedback (Zhou et al., 2022; Yang et al., 2023a; Guo et al., 2023); DSPy, MIPRO, TextGrad, and GEPA extend this idea to structured LM programs, textual gradients, and trajectory-level reflective feedback (Khattab et al., 2023; Opsahl-Ong et al., 2024; Yuksekgonul et al., 2024; Agrawal et al., 2025). More recent systems, including Meta-Harness, Alita, Gödel Agent, Darwin Gödel Machine, AlphaEvolve, and Self-Harness, recursively revise executable agent scaffolds from execution histories, traces, and evaluated variants (Lee et al., 2026; Qiu et al., 2025; Yin et al., 2025; Zhang et al., 2025; Novikov et al., 2025; Zhang et al., 2026). These works establish execution feedback as an optimization signal for model-external computation; our setting extends the editable program from language-side reasoning and control to multimodal sentiment-evidence handling.

Model-generated reasoning and responses can supervise adaptation, as in STaR, RAFT, ReST, and Self-Instruct (Zelikman et al., 2022; Dong et al., 2023; Gulcehre et al., 2023; Wang et al., 2023). Fast-Slow Training, HarnessForge, and WHALE more directly couple model updates with model-external structures (Tiwari et al., 2026; Chen et al., 2026a; Kim et al., 2026), but their external search centers on textual or symbolic/agentic structure rather than multimodal evidence access. SentiHarness makes evidence exposure itself editable, including modality access, organization, and conditional audio–visual follow-up. These choices define a Harness execution interface: the distribution of model-facing requests together with the structured response schema consumed by downstream program logic. RQ3 asks whether adapting through that interface improves retained-Harness performance under a fixed adaptation budget.

B RQ1 Search-Trajectory Analysis

Fixed-Exposure Reasoning Search: RS-2 analysis. We use RS-2 as the primary matched Fixed-Exposure trajectory. The control fixes the model-facing material and call topology throughout search: each candidate receives the same transcript, full utterance WAV, and full MP4 through one joint Omni call. Subclasses may revise the reasoning prompt / response schema and the deterministic `decide()` rule, but they cannot add or remove modalities, crop or window the media, split modalities into separate calls, issue a second query, or conditionally reacquire evidence.

The fixed T/A/V initialization is 76.8% validation Acc-2. Search reaches 370/456 (81.14%) with `tav_corroborated_intensity_gate`. The prompt asks the single fused T/A/V call to require corroboration across channels before assigning a positive result; the downstream `decide()` rule clips the parsed score to $[-1, 1]$ without adding a second evidence-processing ruleset. Under CH-SIMS Acc-2, changes in confidence or intensity on the same side of zero do not affect the binary decision, so only edits that move examples across zero can improve the selection metric.

This final mechanism also illustrates the limitation of self-consistency inside one fused response. The same model call both observes the modalities and executes the requested corroboration criterion, so downstream logic receives no independent modality-specific observation with which to verify evidence that the fused response omitted, compressed, or mixed. The fixed-exposure result should therefore be interpreted as a limitation of reasoning over one fused multimodal observation, not as evidence that audio or visual information is intrinsically harmful.

Table 7: Held-out CH-SIMS test Acc-2 (%) of the validation-selected Harness from each Fixed-Exposure repeat. No test-time reselection is performed.

Repeat	Validation-selected Harness	Val Acc-2	Test Acc-2
RS-1	`tav_witness_ledger_rung_floor`	81.36	81.4
RS-2	`tav_corroborated_intensity_gate`	81.14	81.2
RS-3	`h11_span_validity_gate`	80.70	79.2

Across all three Fixed-Exposure repeats, the validation-selected Harness scores below its corresponding full-search counterpart on the held-out test set (Table 5). We use RS-2 for the matched validation trajectory analysis in the main text; Table 7 records the selected control programs and their validation scores.

Full-search contrast. Table 5 in the main text brings the three held-out comparisons together. The audited Search-1 and Search-2 trajectories contain different executable evidence-handling structures; Table 14 records representative snapshots and their evidence access patterns.

C Detailed H_0 and H_{85} Logic

H_0^{TAV}. The multimodal reference Harness makes a single request containing text, audio, and the full video. The inner-loop model returns a polarity, a rationale, and a scalar sentiment score in $[-1, 1]$. The Harness parses the model-written score and applies no subsequent decision rule.

H_{85} first stage. The first request contains text, audio, video, and a 2×2 visual grid. The inner-loop model is asked for modality-specific evidence labels rather than for sole control of the final score. The labels include `pro_strong`, `pro_weak`, `neg`, and `none`. Program rules then combine these outputs. Crossing zero is allowed only when the text is `pro_weak`/`pro_strong`, or when both voice and face are `pro_strong`. A single positive acoustic or visual cue cannot move the prediction above zero. The corresponding grounded values are 0.35 for `pro_weak`, 0.8 for `pro_strong`, and -0.6 for `neg`.

Conditional second stage. A second request is issued only when the ancestor decision remains exactly 0 with a neutral polarity. Text is removed. Voice and face are judged separately as positive, negative, or neutral. The score changes from 0 to 0.35 only when one of the two channels is positive and the other is neutral. Two positives, any negative, or a parse failure leave the score at 0. On the 457-example CH-SIMS test set used for this evaluation, this additional request is triggered for 19 examples.

Case: conditional promotion. For test example 259 (gold $+1.0$; utterance gloss: "I can only say I got lucky this time"), the first stage returns no grounded positive evidence and leaves the prediction at 0. The text-free follow-up judges voice as neutral and face as positive. The program therefore changes the prediction to 0.35, correcting the binary decision.

Case: preventing a visual override. For test example 140 (gold -0.6; utterance gloss: "give up applying to be a pilot"), the first stage contains positive visual evidence but insufficient corroboration to cross zero. The follow-up judges voice as negative and face as positive. Because this is not the permitted positive–neutral pattern, the score remains 0; the positive visual cue cannot override conflicting negative acoustic evidence.

D Search-Space Specification and Emergent Evidence Organization

For the full multimodal arm, the proposer skill specifies the admissible edit space rather than a target Harness architecture. Each candidate is a complete executable `answer()` program, so request construction, modality access, conditional model calls, response handling, and decision logic can be revised within the same search unit. The skill preserves modality-specific access to text, audio, and video: text remains textual input, audio is passed through the audio channel, and video is passed through the video channel rather than being collapsed into text by ASR. The Fixed-Exposure Reasoning Search control uses the same T/A/V modalities but freezes them into one joint call; only the reasoning/output contract and downstream decision rule are editable. Full SentiHarness may instead drop a modality, crop or select visual evidence, restrict an audio window, issue an additional model call, modify conflict handling, or revise the reasoning and decision logic.

Table 8: How the proposer skill realizes the multimodal editable space. "Prescribed" denotes interface or search-unit constraints explicitly fixed by the skill; "permitted" denotes mechanisms the proposer may use without requiring every candidate to do so; "emergent" denotes concrete structures discovered by search rather than specified in advance.

Paper-level dimension	Skill-level realization	Scope	Status
Modality-specific evidence access	Separate text, audio, and video channels; audio/video are not folded into the text input	Preserves non-textual evidence as independently accessible input	Prescribed
Sentiment-cue acquisition	Modality dropping, visual cropping/selection, audio-window restriction, and additional model calls	Allows the candidate to change which evidence is acquired for a sample	Permitted
Evidence organization	The executable program may change modality-specific request construction, call ordering and conditionality, and how returned evidence is represented and consumed by later program logic	Makes organization across model calls editable without fixing a specific evidence schema	Permitted
Cross-modal evidence use	Conflict-dependent modality dropping or trust selection and tighter conflict rules	Allows returned modality-specific evidence to alter later decisions	Permitted
Reasoning strategy	The same candidate program may revise the reasoning/gating logic while retaining a parseable final score	Places evidence handling and reasoning in one executable search unit	Permitted
Harness-dependent execution interface	Candidate edits can change the content, number, timing, and requested response format of inner-loop model calls	Changes both the request distribution and the response schema faced by the inner-loop model	Consequence

Importantly, the skill does not prescribe the final organization of multimodal evidence. Structures observed later in searched Harnesses—including isolated modality reads, structured text/voice/face evidence labels, agreement or veto rules, and text-free audiovisual verification—are not enumerated as a target architecture in the skill. They emerge from recursive search within the broader allowed mechanism class. Likewise, the skill permits sample-dependent branching through conflict conditions, gates, and conditional follow-up calls, but does not require every candidate to be adaptive. Thus the implementation supports the paper-level claim that sentiment-cue acquisition, organization across model calls, cross-modal evidence use, and reasoning are jointly editable, while keeping the specific executable organization itself a product of search.

E RQ2 Paired Significance Analysis

We quantify the model-state dependence of Harness utility on the CH-SIMS test set ($N = 457$) using paired bootstrap resampling. For each bootstrap replicate, the same resampled example indices are applied jointly to all compared conditions so that within-example dependence is preserved. We use 50,000 bootstrap replicates and report percentile 95% confidence intervals. For a bootstrap statistic S^*, the two-sided raw p-value is computed as

$$p_{\text{raw}} = 2 \cdot \min\left(\frac{\#\{S^* \le 0\} + 1}{B + 1}, \frac{\#\{S^* \ge 0\} + 1}{B + 1}\right),$$

with $B = 50{,}000$. Holm correction is applied separately to the three contraction tests and to the three recovery tests.

For a fixed Harness H, the contraction statistic is

$$C_H = \Delta(H, M_{\text{Base}}) - \Delta(H, M_{\text{Plain}}),$$

and the recovery statistic is

$$R_H = \Delta(H, M_{\text{HA}}) - \Delta(H, M_{\text{Plain}}).$$

Absolute performance under the retained Harness. For Harness-aware SFT versus Plain SFT with H_{85}, the observed difference is +2.19 Acc-2 points; a paired bootstrap gives a 95% CI of $[0.22, 4.38]$, while exact McNemar gives $p = 0.064$. These tests condition on the trained checkpoints, not repeated training runs. The same-interface H_{15} comparison is 87.97 versus 84.68. Relative to the +0.44-point H_0 improvement, the +2.19-point H_{85} improvement corresponds to the +1.75 marginal-utility recovery reported below.

Table 9: CH-SIMS Harness gains and model-state shifts (percentage points). C measures the Base-to-Plain contraction in marginal Harness gain; R measures the Plain-to-HA change.

Harness	Δ_{Base}	$\Delta_{\text{Plain SFT}}$	$\Delta_{\text{HA–SFT}}$	C	R
H_{15}	+4.59	-1.53	+1.32	**+6.13***	+2.84
H_{82}	**+5.91**	+0.88	+1.97	**+5.03***	+1.09
H_{85}	+5.47	**+1.97**	**+3.72**	+3.50	+1.75

*Holm-adjusted $p < 0.05$ within the three contraction tests.

All three recovery quantities are positive, but their 95% confidence intervals include zero.

Table 10: Paired bootstrap analysis of Harness-gain contraction and recovery on CH-SIMS. Effects and confidence intervals are in percentage points. Holm correction is performed separately within the contraction and recovery families.

Effect	Harness	Estimate	95% CI	Raw p	Holm p
Contraction	H15	6.13	[2.19, 10.28]	0.0030	**0.0089**
	H82	5.03	[1.09, 8.97]	0.0147	**0.0294**
	H85	3.50	[-0.44, 7.44]	0.0978	0.0978
Recovery	H15	2.84	[-0.88, 6.56]	0.1367	0.4102
	H82	1.09	[-2.63, 4.81]	0.6037	0.7916
	H85	1.75	[-1.97, 5.47]	0.3958	0.7916

The contraction estimates are positive for all three Harnesses. H15 and H82 remain statistically supported after Holm correction, whereas H85 is directionally consistent but not statistically resolved. The three recovery estimates are also positive, but none is statistically supported after correction and all three confidence intervals include zero. We therefore use the contraction results to support RQ2's model–Harness coupling claim, while treating marginal-utility recovery under Harness-aware SFT as descriptive.

F Absolute Harness–Model Scores

Table 11: CH-SIMS absolute Acc-2 (%) for the fixed Harness family.

Harness	Base	Plain SFT	Harness-aware SFT
H_0	81.84	86.21	86.65
H_{15}	86.43	84.68	87.97
H_{82}	**87.75**	87.09	88.62
H_{85}	87.31	**88.18**	**90.37**

G HARNESS-AWARE SFT OBJECTIVE AND INTERFACE SERIALIZATION

Why H_{15} is the training interface. The CH-SIMS Harness-aware data were designed after H_{85} had already been chosen as the evaluation Harness. H_{15} is used because its request–response contract is the first stage inherited by H_{85}, not because H_{15} was separately selected as a training-time champion. Concretely, H_{85} first executes the inherited H_{15}-compatible stage and only then, for unresolved zero/flat cases, may issue an additional text-free delivery-pair query. Harness-aware SFT targets only this shared first stage: the H_{15} prompt, full utterance audio, full video, the 2×2 visual mosaic inherited from the earlier visual-grid revision, and the structured evidence fields consumed by the downstream Harness. The conditional delivery-pair query is not included in SFT.

Deterministic target construction. No Base-model execution, external teacher, parsing, or response repair is used to build the CH-SIMS Harness-aware targets. For example i, let M_i be the original overall sentiment-score string from `label.csv`, let

$$\bar{M}_i = \mathrm{clip}(\mathrm{float}(M_i), -1, 1),$$

and let t_i be the transcript after removing whitespace for quote extraction. The assistant target is constructed deterministically as

$$z_i^{H_{15}} = g(M_i, t_i), \tag{3}$$

and the training set is

$$\mathcal{D}_{H_{15}} = \{(c_i^{H_{15}}, z_i^{H_{15}})\}_{i=1}^{N}. \tag{4}$$

The conversion writes one record per original training example. Table 12 gives the exact provenance of each supervised field.

Table 12: Deterministic provenance of the CH-SIMS Harness-aware SFT target fields. No field is generated by the Base model or an external teacher.

Target field	Construction rule
`<polarity>`	Sign of $\bar{M}_i$: positive if > 0, negative if < 0, neutral if $= 0$.
`<score>`	Original overall-score string M_i from `label.csv`, with surrounding whitespace removed; it is not replaced by a model prediction or reformatted from $\bar{M}_i$.
`<reason>`	Fixed template with the gold-derived polarity inserted; no model-generated rationale is used.
`<text_quote>`	Deterministic transcript substring: `none` for neutral examples or when the whitespace-stripped transcript has fewer than four characters; otherwise the first 15 characters.
`<voice_verdict>`	Constant `none` for all training examples.
`<face_verdict>`	Constant `none` for all training examples.

The audio- and visual-specific annotations available in CH-SIMS are deliberately discarded during conversion. Thus the structured target adds neither modality pseudo-labels nor teacher-generated reasoning. The text quote is included to satisfy the quote-validation contract used by the downstream Harness. In the 1,368-example training split, 218 targets use `none` for `<text_quote>`; both modality-verdict fields are `none` in all 1,368 targets.

Record construction and filtering. A record is skipped only if the CSV row has fewer than nine columns, its split is not one of `train`/`valid`/`test`, or the required WAV or MP4 file is missing. Mosaic-generation failure does not remove the record; it only omits the `<image>` component. All 1,368 CH-SIMS training examples used here have mosaics available. Dataset construction performs no model call and therefore has no parse-failure, invalid-verdict, retry, or repair path.

Training objective and loss mask. Let $\mathcal{M}_i$ denote token positions with nonzero training loss. The Harness-aware objective is

$$\mathcal{L}_{\mathrm{HA}}(\theta) = -\mathbb{E}_i \sum_{\ell \in \mathcal{M}_i} \log p_\theta \left(z_{i,\ell}^{H_{15}} \mid c_i^{H_{15}}, z_{i,<\ell}^{H_{15}} \right). \tag{5}$$

The stored training arguments use `ms-swift 4.5.3`, `template=qwen3_omni`, `use_chat_template=true`, and `loss_scale=default`; samples contain no per-example `loss` override. Under this implementation, `ContextType.RESPONSE` and `ContextType.SUFFIX` receive weight 1 and all other positions weight 0. The user-side request, media placeholders, source material, and the `<|im_start|>assistant` role header are therefore masked; loss is applied to the structured assistant response and closing `<|im_end|>` suffix.

G.1 Concrete CH-SIMS Harness-aware training instance

The following instance is taken from the first record of the CH-SIMS Harness-aware training file. It instantiates the H_{15}-compatible first-stage interface retained by H_{85}; it is not the output of running H_{15} or a teacher model. The source example is `video_0001/0001` from the training split with overall gold sentiment score -1.0 (Negative). To avoid machine-specific paths, we report media locations relative to the CH-SIMS root:

```
audio/video_0001/0001.wav
video/video_0001/0001.mp4
mosaic/video_0001/0001.png
```

Line wrapping below is added only for typesetting; the textual content and output tags are otherwise reproduced from the training instance.

Instantiated first-stage user request.

```
<audio><video><image>
你是多模态情感分析模型。下面给出一段很短的对话片段，已切成三路材料：
文本、语音、画面。

材料说明：除整段视频与音频外，另附一张 2x2 拼图，按从左到右、从上
到下的顺序排了这段短对话中 4 个时间点的画面。拼图是为了把人物放大
看清细节，它和视频是同一段对话，不要当成多个不同场景。
拼图放大以后更容易编出并不存在的表情：只有当你确实看清了眼、眉、
嘴的具体形状并能够复述出来时，才算看到了情绪；否则一律写"看不出"。

【你要做的事】
把三路材料各自读到的东西如实写下来，并给每路一个可核对的判定。
**三路的合并由系统完成，你不需要自己计算最终分数，也不需要对结果
负责。**正因如此，请不要为了"稳妥"而把有方向的一路写成 `none`：
读出方向就写方向，读不出才写 `none`。也不要为了让三路一致而改动
某一路的判定，冲突就如实写冲突。

【必须区分"内容"与"情绪"】
文本只告诉我们说话人说了什么，不等于说话人自己开心。
"没有出现负面词"不等于"正面"；语气平静、表情自然、没有明显波动
也不等于"正面"。

【逐路写观察与证据】
文本：这句话本身带情绪还是只是在说明、提问、招呼、解释？
      若带情绪，方向是哪种。
      此外必须从上面给出的那句话里**原样抄一小段**（连同标点，
      4 到 15 个字）作为证据，抄最能支撑你判断的那几个字。
      抄不出（整句确实没有情绪）就写 `none`。
语音：你实际听到的语气是什么（上扬/低沉/急促/平淡/停顿）？
      听到的写具体描述，读不出写 `none`。
画面：拼图里眼、眉、嘴、下颌的具体形状是什么？
      看到的写具体描述，看不出写 `none`。

【给三路各下一个可核对的判定】
文本判定从这四个里选一个：
  `pro_strong` = 明确的正面表达（赞美、喜爱、感谢、真诚的高兴）
  `pro_weak`   = 有正面倾向但幅度小（客气、暖意、善意的调侃）
  `neg`        = 负面
  `none`       = 读不出方向
语音判定从这四个里选一个：
  `pro_strong` / `pro_weak` / `neg` / `none`
画面判定从这四个里选一个：
  `pro_strong` / `pro_weak` / `neg` / `none`

【最后写你的整体判断】
在三个判定之外，再给出你自己对整段材料的整体估计：`<polarity>` 与
`<score>`。这只是你的整体估计，系统会结合三路判定重新合成；照实写即可。

严格按下面顺序输出，不要附加其它顶层内容：
```

```
<polarity>positive 或 negative 或 neutral</polarity>
<reason>分点写文本、语音、画面各自读到什么，哪里冲突，以谁为准。</reason>
<evidence>
<text_quote>原样抄的 4-15 个字，或 none</text_quote>
<voice_verdict>pro_strong 或 pro_weak 或 neg 或 none</voice_verdict>
<face_verdict>pro_strong 或 pro_weak 或 neg 或 none</face_verdict>
</evidence>
<score>
[-1, 1] 之间的一个小数。-1 非常负面，0 中性，1 非常正面。
</score>

文本：我不想嫁给李荼
```

Supervised assistant completion.

```
<polarity>negative</polarity>
<reason>
文本语义倾向为negative。语音读不出明确方向，写 none。
画面读不出明确方向，写 none。最终以文本整体估计为准。
</reason>
<evidence>
<text_quote>我不想嫁给李荼</text_quote>
<voice_verdict>none</voice_verdict>
<face_verdict>none</face_verdict>
</evidence>
<score>-1.0</score>
```

This example makes the supervision source explicit. The structured completion is not copied from a Base-model or teacher trajectory: its polarity and score come from the original overall sentiment annotation, its reason is a fixed template, its text quote is deterministically extracted from the transcript, and both modality verdicts are fixed to `none`. The resulting serialization matches the first-stage parser contract retained by H_{85} without introducing additional audio/visual labels or model-generated pseudo-supervision.

H Adaptation-Budget Accounting

Table 13 separates the controlled CH-SIMS comparison from the configuration-specific CMU-MOSI and MELD runs. We use optimization steps $\times$ effective batch size $\times$ cutoff length as a *nominal token-slot budget*; this is an accounting proxy rather than a claim of identical realized FLOPs because actual sequence lengths and multimodal tokenization may differ.

Table 13: Training-budget accounting for the Plain-SFT and Harness-aware configurations used in the paper. The nominal token-slot budget is an upper-bound-style proxy based on optimizer steps, effective batch, and cutoff length.

Dataset	Configuration	Steps	Eff. batch	Cutoff	LR	Nominal token slots / status
CH-SIMS	HA-SFT	91	4	8192	1×10^{-4}	$\approx$ 3.0M; matched
	Plain SFT	91	4	8192	1×10^{-4}	$\approx$ 3.0M; matched
CMU-MOSI	HA-SFT (reported ckpt-20)	20	4	8192	1×10^{-4}	$\approx$ 0.66M; unmatched
	Plain SFT	161	8	2048	1×10^{-5}	$\approx$ 2.6M; unmatched
MELD	HA-SFT (reported high-score run)	853	–	8192	5×10^{-6}	$\approx$ 224M; unmatched
	Plain SFT	313	–	2048	1×10^{-5}	$\approx$ 20.5M; unmatched

On CH-SIMS, Harness-aware conversion produces 1,368 raw training records, exactly one per original training example; 8192-token bin packing turns these into 365 packed sequences with average packed length about 7,413 tokens. The corresponding Plain-SFT data form 115 packed sequences. Both runs still use the same 91 optimization steps and effective batch size, so Plain SFT resamples its smaller packed set as needed to fill the matched update budget. The comparison therefore holds the principal optimization budget fixed while changing the model-facing training interface, rather than comparing different amounts of optimization.

Plain-SFT interface. The Plain-SFT baseline is built from the same 1,368 CH-SIMS training examples through a standalone task-level request–response format rather than the H_{15} evidence schema. It does not supervise `text_quote`, `voice_verdict`, or `face_verdict` fields. Under the same 8192 cutoff its shorter records pack into 115 sequences, which are resampled to fill the

same 91 updates. This matching controls the principal optimizer budget but not the number of times an original short record is revisited.

The CMU-MOSI result used in the main text comes from Harness-aware checkpoint 20, not the completed 54-step run. Its nominal token-slot budget is therefore substantially smaller than the Plain-SFT budget, which rules out "more Harness-aware compute" as an explanation but does not remove differences in learning rate, cutoff, batch size, or training duration. For MELD, a Harness-aware run aligned with the Plain-SFT settings (313 steps, cutoff 2048, learning rate 1×10^{-5}, approximately 20.5M nominal token slots) achieved only about 60–62 WF1 with the Harness. The reported 66.81 result instead comes from the later 853-step, 8192-cutoff run with a much larger nominal budget. We therefore do not use CMU-MOSI or MELD to claim a controlled Harness-aware advantage.

I Repeated Proposer-Search Dynamics

Recursive versus independent proposals. Independent candidate search uses the same proposer, task specification, evaluator, and 50-round candidate budget, but does not condition new proposals on the accumulated candidate programs, scores, or execution traces from previous rounds. Evaluated candidates therefore do not become parents for later proposals. Recursive search retains this history and revises prior candidates from execution feedback. For both methods, the connected trajectory reports the best of the three candidates in each round; the remaining two candidate scores are shown as pale points.

In this dedicated comparison, the best observed validation Acc-2 is approximately 87.7 for recursive search and 85.3 for independent search. Repeated proposer-search runs reach high-performing regions through structurally distinct Harnesses and are analyzed separately in Appendix I.

The score-level audit uses three completed CH-SIMS proposer-search trajectories, denoted Search-1, Search-2, and Search-3. The suffixes identify independent outer-loop proposal trajectories; the inner-loop model and evaluation protocol are unchanged. As in the main search figures, each connected point is the round-best Harness among three candidates, while stars identify rounds that update the maintained frontier Harness. The lines can therefore decrease even though the frontier itself is monotonic.

The structural audit uses Search-1 and Search-2 because complete candidate programs and execution records are available for those trajectories. The score-level analysis above measures trajectory-level variability across three completed searches, whereas the structural audit below compares executable structures for the two trajectories with complete program records.

Table 14: Audited intermediate structural snapshots from Search-1 and Search-2. These historical candidate states are shown only to compare executable architectures; they are not final frontier results.

Trajectory	Audited snapshot	Primary evidence interface	Conditional re-query
Search-1	H27	Fused main decision; A/V separated only during follow-up	Yes; text-free A/V verification on unresolved neutral/positive cases
Search-2	H18	Persistent isolated text, audio, and visual reads with programmatic voting/veto	Yes; fallback evidence acquisition for veto/visual-read failures

Search-1 follows a fused-decision path with post-hoc gating and conditional verification, whereas Search-2 follows an isolated-vote path with vetoes and fallback evidence acquisition. These audited snapshots show that recursive search can reach strong regions through structurally different programs; the recurring pattern is that the Harness, rather than a single unconstrained model call, controls evidence acquisition, organization, and integration.

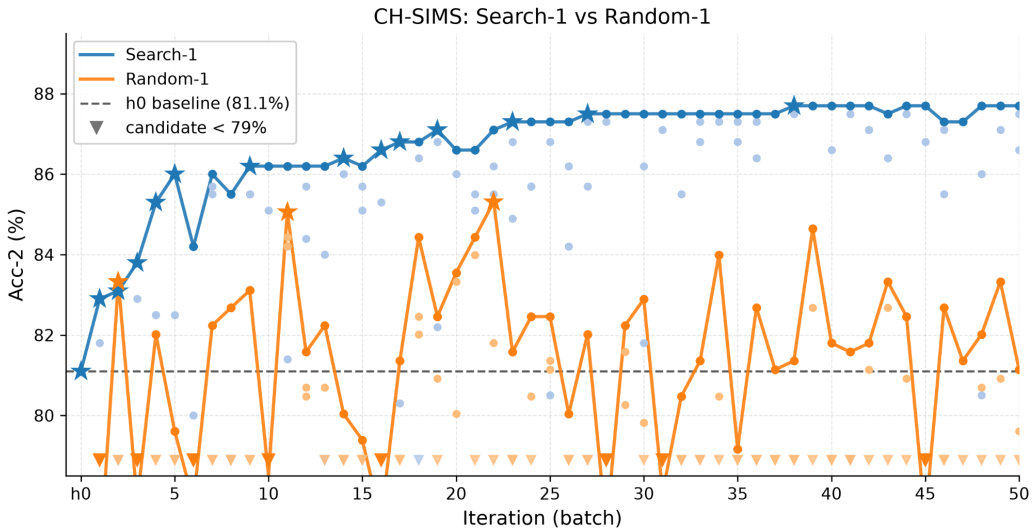


Figure 3: Dedicated Search-1 versus Random-1 comparison on CH-SIMS under the same 50 proposal rounds. Random-1 is the independent candidate-search control. Connected points show the round-best Harness; pale points show the other candidates, stars mark frontier updates, and downward triangles denote candidates below the plotted range. This experiment is separate from the repeated recursive-search trajectories in Appendix I.

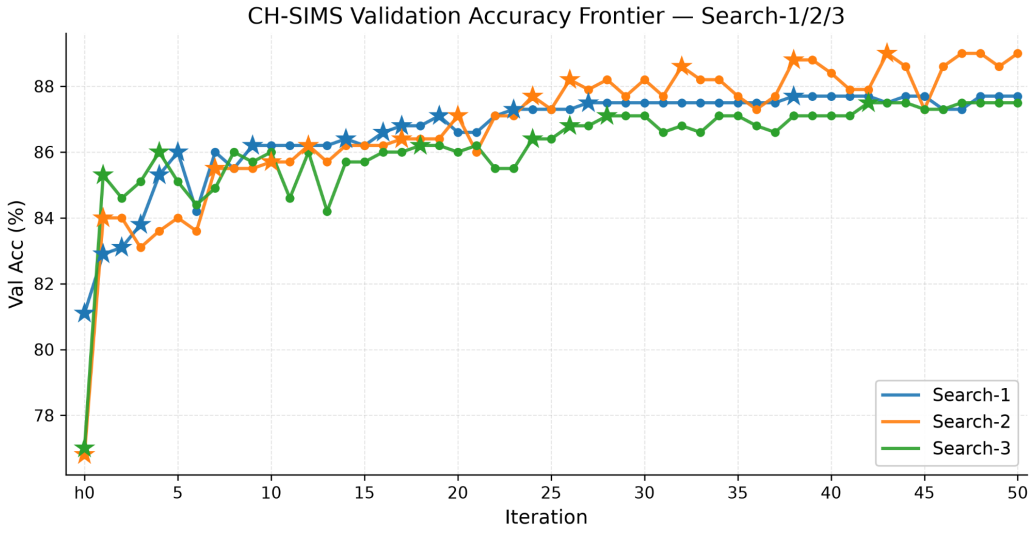


Figure 4: CH-SIMS round-best validation Acc-2 across three completed proposer-search trajectories (Search-1, Search-2, and Search-3). Each connected point is the best of three candidates in that proposal round; stars mark updates to the maintained frontier Harness.

J Detailed Experimental Protocol

Round-best candidates and maintained frontiers. Each proposal round t produces three candidate Harnesses, denoted $\mathcal{C}_t$. We distinguish the *round-best Harness*

$$R_t = \arg\max_{H \in \mathcal{C}_t} P_{\mathcal{D}_{\text{val}}}(H, M_\theta)$$

from the maintained *frontier Harness*

$$F_t = \arg \max_{H \in \{H_0\} \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_t} P_{\mathcal{D}_{\text{val}}}(H, M_\theta).$$

The controller retains F_t as the highest-validation Harness found so far, with $H^\star = F_B$. Search-trajectory figures connect the round-best scores $P_{\text{val}}(R_t)$; stars mark rounds where R_t becomes the new F_t, and pale points show the other two candidates when available. Thus the connected trajectory may decrease although the maintained frontier is monotonic.

Shared search configuration. Searches on all three benchmarks use frozen Qwen3-Omni-30B-A3B-Instruct as the inner-loop model and `deepseek-v4-flash` as the proposer. Omni inference uses `temperature=0`, `enable_thinking=false`, `vlm_timeout=1800 s`, `batch_size=1`, and `max_tokens=2048`. Video frames are sampled at request time with `target_side=640` and `frame_budget=16`; one request contains at most 320 frames. The outer loop imposes no fixed limit on the number of Omni calls that a candidate may issue for one example; this is determined by the executable candidate itself.

Each search run allows at most 50 proposal rounds. The proposer writes exactly three candidates per round, for an upper bound of 150 candidates, and candidates are evaluated serially. The proposer timeout is 7,200 s and the per-candidate timeout is 21,600 s. The proposer launcher does not set a separate output-token limit. Evaluator concurrency is `max_workers=12` on CH-SIMS, 5 on CMU-MOSI, and 8 on MELD.

Recursive versus independent proposal context. The recursive and independent comparisons share the same proposer, evaluator, dataset, candidate count, and per-candidate execution procedure. The intended difference is proposal context: recursive search conditions each new round on accumulated candidate programs, scores, and training-split execution traces, whereas independent search omits this accumulated candidate history. Consequently, an independently evaluated candidate cannot become the parent of a later proposal through stored search history.

Repeated-search suffixes. Search-1, Search-2, and Search-3 are presentation labels for independent proposer-search trajectories. The suffix does not alter the frozen inner-loop weights, Omni decoding configuration, response-cache definition, or official dataset splits. Identical Omni requests therefore have the same model-side configuration across trajectories. Repeated searches are used to measure proposer-induced search-path variability rather than model, decoding, or split variability.

Table 15: Dataset splits and frontier-selection metrics used by Harness search. All official split examples are used without subsampling.

Dataset	Train	Valid	Test	Frontier selection
CH-SIMS	1368	456	457	Valid Acc-2
CMU-MOSI	1284	229	686	Valid Non0 Acc-2
MELD	9989	1109	2610	Valid weighted F1

Search/data separation. For CH-SIMS, all 1,368 training examples are executed to produce sample-level traces, all 456 validation examples are used to update and finally select the frontier Harness, and all 457 test examples remain sealed during search. CMU-MOSI follows the same protocol with 1,284 training, 229 validation, and 686 test examples; train and validation are not merged. MELD likewise uses all 9,989 training examples for traces, all 1,109 validation examples to update and select the frontier, and keeps all 2,610 test examples sealed. Validation examples and validation traces are not exposed to the proposer; only aggregate validation performance is provided.

CH-SIMS and CMU-MOSI metrics. The search evaluator computes Acc-2 and support-weighted binary F1 on the same scored examples, but frontier selection is based on Acc-2 for CH-SIMS and Non0 Acc-2 for CMU-MOSI. Binary F1 is computed with `sklearn.metrics.f1_score(...,average="weighted")`: the two classwise F1 values are weighted by their gold-label supports. Parse failures count as errors and remain in the metric denominator; for binary scoring, the predicted class is set to the class opposite the gold label.

For CH-SIMS, prediction and gold scores are clipped to $[-1, 1]$. The MMSA `Mult_acc_2` protocol assigns $(-1.01, 0]$ to the non-positive class and $(0, 1.01]$ to the positive class; gold score 0 remains in the non-positive class and is not removed. For CMU-MOSI, scores are clipped to $[-3, 3]$; values ≤ 0 are non-positive and values > 0 are positive. Gold-zero examples are excluded from both Non0 Acc-2 and its associated weighted binary F1.

MELD metric. MELD has no Acc-2 thresholding. Frontier selection uses seven-class support-weighted F1 over `neutral`, `surprise`, `fear`, `sadness`, `joy`, `disgust`, and `anger`. Each class F1 is weighted by its gold support over all 1,109 validation utterances. Validation accuracy is retained only as a diagnostic and does not participate in frontier selection.

CH-SIMS Acc-3. Acc-3 is not computed by the searcher and never affects the frontier. For final CH-SIMS reporting, scores are clipped to $[-1, 1]$ and mapped using the MMSA `Mult_acc_3` intervals: $(-1.01, -0.1]$ is negative, $(-0.1, 0.1]$ is neutral, and $(0.1, 1.01]$ is positive. CMU-MOSI does not use Acc-3 in our evaluation; its MMSA Acc-5 and Acc-7 metrics are also not used by the searcher.

CH-SIMS preprocessing. CH-SIMS is read from the headerless nine-column annotation file as `video_id, clip_id, text, M, T, A, V, tag, split`. The target is the M score. The `clip_id` is zero-padded to four digits. Video and audio are loaded from `video/{video_id}/{clip}.mp4` and `audio/{video_id}/{clip}.wav`, respectively.

CMU-MOSI preprocessing. CMU-MOSI is read from the headered annotation file using `video_id, clip_id, text, label, mode`. Labels remain on their original $[-3, 3]$ scale. Video is resolved under `Raw/{video_id}/` using filenames such as `{video_id}_{clip_id}.mp4`, with audio loaded from the corresponding `.wav` file.

MELD preprocessing. MELD uses the official `train_sent_emo.csv`, `dev_sent_emo.csv`, and `test_sent_emo.csv` files. The target is the lowercase value of the `Emotion` field. Media are resolved by `dia{Dialogue_ID}_utt{Utterance_ID}` within the split-specific directories: `train_splits` for training, `dev_splits_complete` for validation, and `output_repeated_splits_test` for test. Up to the six most recent preceding turns in the same dialogue are concatenated as `Speaker: Utterance` strings and supplied as `dialogue_context`. If audio cannot be decoded from a video, that example is retained with text only and both video and audio inputs are left empty.

General input handling. The dataset loaders do not tokenize text and do not resample media at load time. Labels are passed through unchanged as target strings. Video frames are sampled only when an Omni request is issued, using the shared frame settings above.

K Aggregate Revision Statistics

Each candidate is assigned to exactly one edit category based on its primary change relative to its parent. An added `ask_omni` inside a conditional branch is categorized as conditional-query/gating; an unconditional increase in calls is request construction; changed model-facing labels are response format; remaining changes are decision rules. For a candidate evaluated in round t, delta is measured against the frontier score before that round, $P_{\text{val}}(F_{t-1})$; negative averages are therefore expected when most proposals fail to improve the maintained frontier.

The aggregate counts are descriptive rather than causal. In particular, decision-only edits do not exceed the frontier in Search-1, but two such edits improve Search-2. Conditional queries contribute frontier gains in Search-1, while the Search-2 conditional-query category does not exceed its contemporaneous frontier. The main paper therefore emphasizes the concrete frontier transitions rather than ranking edit categories by their average delta.

Table 16: Aggregate candidate-edit statistics. "Positive" counts candidates with frontier-relative $\Delta > 0$.

Trajectory	Category	Count	Share	Mean Δ	Positive
Search-1	Request	9	18%	-23.4	1 (+2.5)
	Format	20	39%	-14.8	3 (avg. +1.9)
	Decision	8	16%	-3.8	0
	Conditional/gate	14	27%	-4.2	2 (avg. +0.6)
Search-2	Request	9	16%	-2.1	2 (avg. +0.9)
	Format	22	39%	-2.9	4 (avg. +1.6)
	Decision	13	23%	-3.0	2 (avg. +1.9)
	Conditional/gate	12	21%	-2.4	0

L Cross-Dataset Harness Transfer

We additionally test whether the CH-SIMS-searched H_{85} transfers directly to CMU-MOSI without re-searching or target-domain calibration. This audit was executed on a separate machine from the main CH-SIMS evaluation; all headline CH-SIMS numbers therefore follow the main-table evaluation, while the transfer run is used only for behavioral analysis.

The direct transfer does not preserve the within-dataset Harness gain. Under the same frozen Base model, direct inference on the CMU-MOSI test set obtains Non0 Acc-2 of 0.8765 (575/656), whereas adding the CH-SIMS-searched H_{85} reduces Non0 Acc-2 to 0.7805 (512/656). Under the all-sample SIMS-style binary protocol, the transferred Harness obtains 0.7784 Acc-2, with MAE 0.679 and correlation 0.585, requiring 2,278 model calls.

The failure is concentrated in examples that the Harness interprets as flat or affectively weak. Among MOSI examples whose gold score is above 0.3 but whose final Harness prediction is negative, 89 cases are observed; 84 of their traces explicitly describe the sample as flat, neutral, lacking obvious emotion, or merely declarative. This is consistent with the CH-SIMS-derived decision rule being too conservative under the MOSI annotation regime.

For example, MOSI test utterance `c7UH_rxdZv4_30`, "AND IT WAS KIND OF INTERESTING", has gold sentiment $+1.2$. Direct Base inference predicts $+0.6$, whereas the transferred Harness treats the linguistic content and delivery as insufficiently positive and exports -0.35. A similar pattern occurs for `cW1FSBF59ik_4`, "SIGH I LOVE WHEN THEY SHOW ALL THE THE WAR SCENES...", whose gold score is $+1.6$: the trace recognizes positive lexical content but downweights it because the vocal delivery is judged flat, again yielding -0.35.

These results distinguish within-dataset search generalization from cross-dataset transfer. In the main experiments, search is performed and selected entirely within each benchmark while the test split remains sealed. Direct transfer is a stricter setting in which label priors, annotation conventions, language, and cue reliability can change simultaneously. We therefore treat cross-dataset Harness transfer as a separate problem rather than as part of the main RQ1–RQ3 evaluation.