MUSE: A MULTILEVEL USER FEEDBACK LEARNING FRAMEWORK FOR SELF-ENHANCING PERSONALIZED MULTIMODAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal agents must produce reliable outputs across modalities and adapt to individual users. A single overall score, however, gives little guidance on whether an error stems from one modality, a conflict between modalities, or a mismatch with user preferences. We introduce MUSE, a Multilevel User Feedback Learning Framework for Self-Enhancing Personalized Multimodal Agents. Within an interaction, Fine-Grained Unimodal Verification (FUV) identifies local defects and constraint violations. Cross-Modal Collaborative Verification (CCV) checks those findings against the combined evidence and identifies conflicts to guide revision. Across interactions, Feedback-Driven Personalization Verification (FPV) learns how verified output properties relate to each user's satisfaction. It updates user-specific adapters from feedback while the task models remain frozen. We evaluate each method on 2,709 instructions completed by 34 participants, yielding 92,106 participant–instruction encounters across 18 task categories and text, visual, and audio inputs. Experiments show that hierarchical verification and feedback-driven adaptation improve multimodal reliability, correction effectiveness, and preference-aligned personalization.

1 INTRODUCTION

Multimodal agents handle requests that combine text, images, audio, and video Koh et al. (2024); Xie et al. (2024); Drouin et al. (2024); Cao et al. (2024); Wang et al. (2024a). Consider meeting minutes drawn from a spoken discussion, shared slides, and chat: the answer must preserve decisions, reconcile numerical details, and present action items in a form the user can use. A fluent summary may still omit an owner, reverse a figure on a slide, or repeat a format the same user previously rejected. Each failure calls for a different correction.

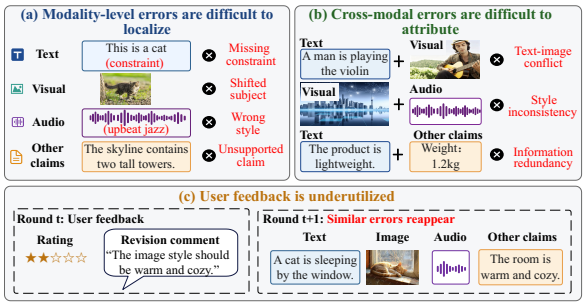


Figure 1: Motivation for MUSE. Existing multimodal agents struggle with modality-specific error localization, cross-modal conflict resolution, and continual personalization.

An overall quality score gives little indication of which source or instruction failed Yu et al. (2024); Yue et al. (2024); Li et al. (2024); Guan et al. (2024). Checking each modality can locate an omission, but a contradiction may emerge only when the sources are compared. A local warning can also be mistaken: changing a valid answer on that basis may introduce a new error. Effective revision therefore requires both specific evidence and a check of whether the reported defect affects the complete answer Guan et al. (2024); Wang et al. (2024b). User feedback poses a related problem across interactions. A rating or correction refers to a particular answer, while the preference it reveals may matter again on a different task. The agent must use that history to judge later outputs without

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.

letting it override the current request. Figure 1 illustrates the three kinds of failure: local defects, conflicts between modalities, and repeated preference mismatches despite earlier feedback.

To address these challenges, we introduce MUSE, a Multilevel User Feedback Learning Framework for Self-Enhancing Personalized Multimodal Agents. MUSE coordinates verification, correction, and personalization over two timescales. Within each interaction, Fine-Grained Unimodal Verification (FUV) localizes modality-specific defects, while Cross-Modal Collaborative Verification (CCV) validates these findings through evidence-grounded adjudication, resolves cross-modal conflicts, and guides targeted revision. Across interactions, Feedback-Driven Personalization Verification (FPV) combines verification evidence, user memory, and real feedback to adapt user-specific satisfaction criteria. The scheduler incorporates the adapted FPV signal into subsequent acceptance and revision decisions. Here, *self-enhancement* refers to adaptation of the agent-level decision and personalization state; the underlying task-model parameters remain frozen. A qualitative example is provided in Appendix C.12.

Our contributions are as follows:

- We introduce MUSE, a self-enhancing framework that unifies within-interaction hierarchical verification and cross-interaction feedback-driven adaptation. Structured verification evidence guides current-round correction and, together with real user feedback, provides personalized supervision for subsequent decisions.
- We design a hierarchical verification mechanism comprising FUV and CCV. FUV localizes modality-specific defects using task-conditioned criteria, while CCV performs evidence-grounded rebuttal and adjudication to validate local findings, attribute cross-modal failures, and produce actionable correction evidence.
- We develop FPV, a feedback-driven personalization module that learns the relationship between structured verification evidence and individual users' satisfaction. By updating only lightweight user-specific adapters, FPV continually adapts personalized decision criteria while preserving the underlying multimodal task models. We also construct a benchmark spanning 18 task categories to evaluate verification, correction, and cross-interaction preference alignment.

2 Related Work

Multimodal foundation models support perception, reasoning, and generation Wu et al. (2024); Zhan et al. (2024); Xie et al. (2025); Yang et al. (2025). Agent frameworks coordinate external tools Gupta & Kembhavi (2023); Surís et al. (2023): HuggingGPT decomposes requests Shen et al. (2023), Chameleon composes modules Lu et al. (2023), and ControlLLM searches tool paths Liu et al. (2024). Their focus on task execution leaves the reliability of the resulting output to be assessed separately.

Self-Refine Madaan et al. (2023), Reflexion Shinn et al. (2023), and CRITIC Gou et al. (2024) use reflection or tool-assisted critique to revise outputs, but their task-level feedback does not explicitly separate local errors from cross-modal conflicts. LaMP Salemi et al. (2024) and Persona-Plug Liu et al. (2025) condition language models on user histories, mainly for textual tasks. MUSE combines localized verification and cross-modal adjudication with online user-specific satisfaction learning, so that current corrections and later preference-aware decisions use different evidence.

3 Method

MUSE is a scheduler-centered framework that combines within-interaction hierarchical verification with cross-interaction feedback-driven adaptation. As shown in Figure 2, the scheduler selects the task tool and verifiers, FUV localizes modality-specific defects, and CCV consolidates the resulting reports into structured cross-modal evidence for acceptance and revision. Across interactions, FPV combines this evidence with user memory and real feedback to adapt a lightweight user-specific adapter whose predictions inform subsequent decision fusion and revision.

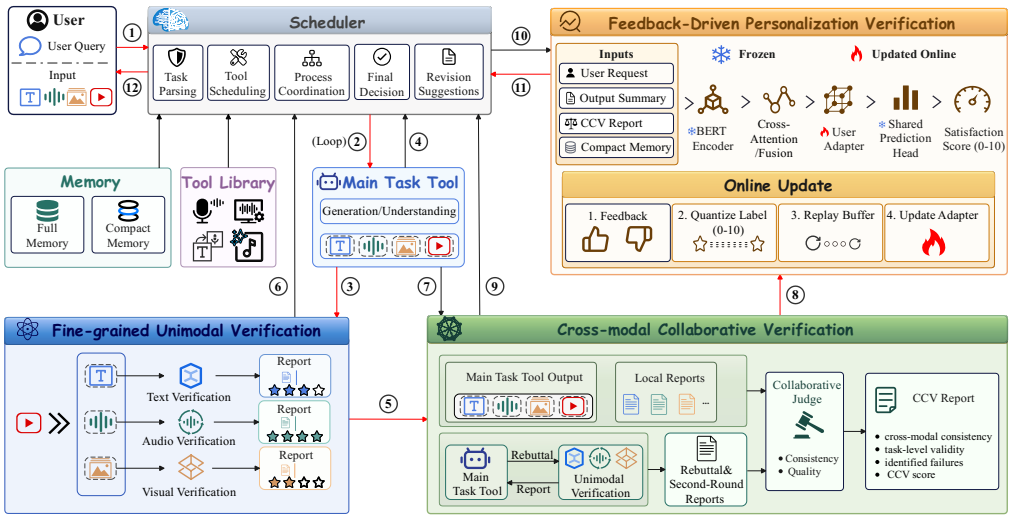


Figure 2: Overview of MUSE. The scheduler coordinates task execution and multilevel verification, while FUV and CCV construct structured evidence for targeted revision. Across interactions, FPV uses user feedback and historical memory to predict personalized satisfaction, which the scheduler incorporates into subsequent decisions.

3.1 Task Parsing and Tool Scheduling

The scheduler parses request q_t and optional input x_t into task profile ρ_t, retaining mandatory constraints separately from historical preferences. It selects task tool A_t and relevant modality verifiers $U_t = \{V_t^k\}_{k=1}^K$ using the task-memory view M_t^{task}. The tool generates $y_t = A_t(q_t, x_t)$; a scheduler-side wrapper supplies a standardized task score and confidence if the tool cannot score itself. Appendix A.1 specifies the full interfaces.

3.2 Memory System

MUSE maintains a full interaction memory M_t^{full}, a task-specific view M_t^{task}, and a compact preference memory $\bar{M}_t^{lite}$, with separate instances for each user. The full memory stores requests, selected tools, outputs, verification reports, feedback, satisfaction labels, confidence scores, and revision traces. M_t^{task} exposes task errors, tool histories, and correction traces to the scheduler, FUV, and CCV, whereas $\bar{M}_t^{lite} = \text{Compress}_{pref}(M_t^{full})$ retains stable preferences, recent satisfaction patterns, decision thresholds, and representative cases for FPV and preference-aware revision. This separation prevents preference signals from implicitly affecting FUV or CCV and confines personalized satisfaction learning to FPV. FPV also maintains a bounded replay buffer B_t to support efficient online adaptation and mitigate forgetting Rolnick et al. (2019). Additional memory analyses and persistent-state considerations are provided in Appendices C.4 and D.

3.3 Fine-Grained Unimodal Verification

Fine-Grained Unimodal Verification (FUV) evaluates individual modalities and constructs localized verification evidence without making the final decision. After selecting A_t and U_t, the scheduler reformulates the original request as verifier-specific instructions:

$$q_t^k = R_k(q_t, \rho_t, A_t, M_t^{task}), \tag{1}$$

where R_k preserves the user constraints and verification criteria relevant to the k-th modality.

Each verifier receives the relevant source, local output, and full-output context:

$$\chi_{t,k} = \left(x_t^{(k)}, y_t^{(k)}, y_t\right), \tag{2}$$

where unavailable components are set to $\varnothing$. The first-round report and report set are computed as

$$v_{t,k}^{(1)} = V_t^k(\chi_{t,k}, q_t^k, M_t^{task}), \quad R_t^{(1)} = \{v_{t,k}^{(1)}\}_{k=1}^K. \tag{3}$$

Each report is represented as

$$v_{t,k}^{(1)} = \{d_{t,k}^{(1)}, s_{t,k}^{(1)}, e_{t,k}^{(1)}, c_{t,k}^{(1)}, \delta_{t,k}^{(1)}\}, \tag{4}$$

where $d_{t,k}^{(1)}$ provides textual evidence for CCV, $s_{t,k}^{(1)}$ is the local quality score, $e_{t,k}^{(1)}$ summarizes detected issues, $c_{t,k}^{(1)}$ denotes confidence, and $\delta_{t,k}^{(1)}$ provides correction suggestions.

FUV checks modality-specific constraints for omissions, distortion, unsupported claims, and other local errors. It can retrieve relevant failures and corrections from M_t^{task}, then passes its localized evidence to CCV.

3.4 Cross-Modal Collaborative Verification

Cross-Modal Collaborative Verification (CCV) checks whether local findings affect the overall answer or conflict across modalities. A task-tool wrapper responds to the FUV reports, the verifiers reconsider their findings, and a cross-modal judge checks both against the source evidence. The resulting adjudication identifies errors and directs correction.

Given the first-round FUV reports, the wrapper produces a targeted rebuttal together with a task-tool score and confidence estimate. Each verifier then re-examines its initial finding:

$$\begin{aligned}(r_t^A, s_t^A, \kappa_t^A) &= W_t^A.\text{assess}(x_t, y_t, R_t^{(1)}, q_t, \rho_t, M_t^{task}),\\ v_{t,k}^{(2)} &= V_t^k.\text{review}(\chi_{t,k}, v_{t,k}^{(1)}, r_t^A, q_t^k, M_t^{task}), \quad R_t^{(2)} = \{v_{t,k}^{(2)}\}_{k=1}^K.\end{aligned} \tag{5}$$

The rebuttal addresses whether each finding is an error, a deliberate choice, or an acceptable trade-off. The judge then combines the source, output, rebuttal, and both rounds of reports:

$$D_t = J_t(x_t, y_t, r_t^A, R_t^{(1)}, R_t^{(2)}, q_t, M_t^{task}). \tag{6}$$

The structured state D_t contains cross-modal consistency, finding validity, and task-level validity scores c_t^{cross}, $c_t^{recheck}$, and c_t^{global}; failure attribution g_t; correction directions δ_t; the CCV score $s_t^{(2)} \in [0, 10]$; and confidence κ_t^J.

CCV provides structured evidence for revision and later FPV training; FPV does not need the raw content or full adjudication history.

3.5 Feedback-Driven Personalization Verification

Feedback-Driven Personalization Verification (FPV) learns from verified output properties and real feedback to predict each user's satisfaction. Its prediction informs acceptance and revision in later interactions.

MUSE shares the FPV backbone and prediction head across users while maintaining a separate lightweight adapter ϕ_u, interaction memory $M_{u,t}^{full}$, and replay buffer $B_{u,t}$ for each user. During online personalization, only the current user's adapter is updated; all shared modules and all other user adapters remain frozen.

Input Encoding and User Adaptation. FPV takes the user request, output summary, structured CCV evidence, and compact user memory as input:

$$X_{u,t}^{(3)} = \left(q_{u,t}, \Sigma(y_{u,t}), D_{u,t}, \bar{M}_{u,t}^{lite}\right). \tag{7}$$

Together, these inputs capture the current intent, the generated output, the verified quality and failure structure, and historical preference patterns. A frozen BERT encoder Devlin et al. (2019) maps them into a shared semantic space. The compact memory is serialized as L preference items $\{m_{u,t}^{\ell}\}_{\ell=1}^{L}$, enabling cross-attention to retrieve relevant historical evidence:

$$\begin{aligned} h_q, h_y, h_d = \mathrm{BERT}\left(q_{u,t}, \Sigma(y_{u,t}), D_{u,t}\right), \quad H_{u,t}^{m} = [\mathrm{BERT}(m_{u,t}^{\ell})]_{\ell=1}^{L}, \\ H_{u,t}^{cur} = [h_q; h_y; h_d], \quad z_{u,t} = \mathrm{MemAttn}\left(H_{u,t}^{cur}, H_{u,t}^{m}, H_{u,t}^{m}\right). \end{aligned} \tag{8}$$

Here, the current evidence serves as the query, while the sequence of memory items provides the keys and values. $\mathrm{MemAttn}$ comprises cross-attention, residual fusion, and pooling.

A user-specific bottleneck adapter, inspired by Houlsby et al. Houlsby et al. (2019), calibrates the shared representation, after which the shared prediction head produces a satisfaction score on a 0–10 scale:

$$\tilde{z}_{u,t} = z_{u,t} + W_{up}^{u}\sigma\left(W_{down}^{u} z_{u,t}\right), s_{u,t}^{(3)} = 10 \cdot \sigma\left(W_s \tilde{z}_{u,t} + b_s\right), \quad s_{u,t}^{(3)} \in [0, 10]. \tag{9}$$

Here, $\phi_u = \{W_{down}^{u}, W_{up}^{u}\}$ denotes the trainable parameters for user u. The shared modules preserve a common representation space and satisfaction scale, whereas ϕ_u captures user-specific satisfaction boundaries. FPV predicts satisfaction only; CCV and the scheduler remain responsible for failure diagnosis and revision.

FPV is initialized for 100 epochs using 800 separate real-user final-satisfaction examples. After an interaction, an LLM-based quantizer converts the participating user's rating or feedback for the final submitted answer into satisfaction label $r_{u,t} \in [0, 10]$ and confidence $c_{u,t} \in [0, 1]$. Feedback below τ_c is retained but does not trigger optimization. Eligible current feedback joins distinct recent and replayed examples from the same user in a batch of at most 16 Rolnick et al. (2019); only that user's adapter is updated. With confidence- and recency-weighted examples, the online objective is

$$\mathcal{L}_{\mathrm{fpv}}^{u} = \frac{\sum_{i \in B_{u,t}^{\mathrm{train}}} c_i \gamma^{t-\tau_i} \mathrm{SmoothL1}(\hat{s}_i, r_i)}{\sum_{i \in B_{u,t}^{\mathrm{train}}} c_i \gamma^{t-\tau_i}} + \lambda_{\mathrm{reg}} \|\phi_u - \phi_u^{\mathrm{old}}\|_2^2. \tag{10}$$

The sample stores the exact answer and CCV evidence used to produce its label; a rating of another method's new answer cannot be inferred from that historical feedback. Each current answer is evaluated before its label updates the adapter, so a decision cannot benefit from feedback on itself. Compared methods receive the same earlier same-user answer–feedback history, while new outputs receive separate benchmark evaluation. Appendix A.2 gives the full replay and encoder specification; Appendix E.3 states the answer–feedback protocol.

3.6 Final Decision Fusion and Revision

The scheduler fuses the wrapper-derived task score s_t^A, confidence-weighted FUV score:

$$s_t^{(1)} = \frac{\sum_k c_{t,k}^{(1)} s_{t,k}^{(1)}}{\sum_k c_{t,k}^{(1)} + \epsilon}, \tag{11}$$

the CCV score $s_t^{(2)}$, and FPV prediction $s_t^{(3)}$, where $\epsilon = 0.5$ provides smoothing. Monotonic learned calibration functions map each score to a shared scale: $\bar{s}_t^{(i)} = \mathrm{Cal}_i(s_t^{(i)}) \in [0, 1]$, where $i \in \mathcal{I} = \{A, 1, 2, 3\}$.

For the task wrapper, FUV, and CCV, reliability is estimated from their current confidence:

$$m_t^{(A)} = \kappa_t^A, \qquad m_t^{(1)} = \frac{1}{K}\sum_{k=1}^{K} c_{t,k}^{(1)}, \qquad m_t^{(2)} = \kappa_t^J. \tag{12}$$

FPV reliability additionally depends on the availability of feedback and recent regression performance. Let $\bar{r}_\tau = r_\tau/10$ and $\widetilde{s}_\tau^{(3)} = s_\tau^{(3)}/10$. We define its reliability and maturity as

$$\mathrm{Rel}_t^{(3)} = \begin{cases} 1 - \frac{1}{|\mathcal{W}_t|}\sum_{\tau\in\mathcal{W}_t}\left|\widetilde{s}_\tau^{(3)} - \bar{r}_\tau\right|, & |\mathcal{W}_t| > 0 \\ 0, & |\mathcal{W}_t| = 0 \end{cases}, m_t^{(3)} = \min\left(1, \frac{N_t}{N_0}\right)\mathrm{Rel}_t^{(3)}. \tag{13}$$

where $\mathcal{W}_t$ is a valid prior-feedback window, N_t is the number of valid feedback samples, and N_0 is the reference count at which the feedback-volume factor saturates.

The scheduler computes normalized fusion weights and the final decision score:

$$w_{i,t} = \frac{m_t^{(i)} + \epsilon}{\sum_{j\in\mathcal{I}}(m_t^{(j)} + \epsilon)}, \quad i \in \mathcal{I}, S_t \quad = \sum_{i\in\mathcal{I}} w_{i,t}\bar{s}_t^{(i)}. \tag{14}$$

Thus, $\sum_{i\in\mathcal{I}} w_{i,t} = 1$ and $S_t \in [0,1]$. If $S_t \geq \tau_s$, the scheduler returns y_t; otherwise, it constructs a targeted revision instruction and reinvokes the task tool:

$$p_t^{\mathrm{revise}} = \mathrm{Revise}\left(q_t, y_t, R_t^{(1)}, R_t^{(2)}, D_t, \bar{M}_t^{lite}, \bar{s}_t^{(3)}\right), y_t' \quad = A_t\left(p_t^{\mathrm{revise}}, x_t\right). \tag{15}$$

Verification and revision continue until the threshold or iteration limit is reached; when the limit is reached without acceptance, the latest generated answer is returned. The default uses learned calibration, no hard gate, and no cold-start mask; $\tau_s = 0.75$ and $\epsilon = 0.5$. Smoothing keeps the FPV weight positive at cold start; maturity changes its relative influence. The four score maps are fitted once on the 800-example real-user development collection, which is disjoint from the 2,709-instruction benchmark, and are frozen before benchmark evaluation. The same external collection also initializes FPV; we therefore treat it as a system-development collection rather than a held-out calibration benchmark and do not use in-collection calibration error as evidence of generalization. FUV and CCV themselves remain prompt-based. Appendix C.10 specifies the calibration form and fixed decision settings.

4 Benchmark and Evaluation Protocol

The benchmark contains 2,709 instructions in 18 categories using audio (A), visual (V: images/videos), and text (T); 1,340 are personalized and 1,369 are not (Table 12). Each of 34 participants completes all instructions, and every main-comparison method runs once per participant–instruction pair, yielding 92,106 outputs per method. The instruction catalog stores the 2,709 requests, while a separate anonymous log stores participant index, interaction order, method output, and answer-linked history. There is no fixed train/test split: each new method output is evaluated before the corresponding answer-linked historical feedback record becomes available for subsequent interactions (Appendix E.1).

Outputs receive 0–10 ratings for Goal Fulfillment (GFS), Explicit Instruction Compliance (EIC), Multimodal Faithfulness and Coherence (MFC), and Preference-Aligned Satisfaction (PAS). An LLM evaluator proposes rubric-based scores Liu et al. (2023); Kim et al. (2024); Xiong et al. (2025); Tan et al. (2025), which the 34 participants review for their own encounters under the common rubric; five experts supervise ambiguous cases rather than independently rescoring every output. Only participant-confirmed scores are reported. Scores first average the 34 users for each applicable instruction, then average instructions within a task, and finally macro-average the same 18 tasks for every method. PAS applies only to personalized instructions; Avg. is formed from the four metric means before rounding. PAS is a rubric-based preference-alignment metric, whereas FPV labels come from answer-linked user feedback. Appendix B gives the rubrics and aggregation protocol.

5 EXPERIMENTS

5.1 IMPLEMENTATION DETAILS

The default uses Qwen3.5-Omni, learned score calibration, and at most two verification–revision rounds, without a hard gate or cold-start mask. FPV is initialized for 100 epochs from an 800-example real-user development collection outside the benchmark; the same external collection provides the fixed targets used to fit the four module-score maps before evaluation. No benchmark score is used to refit these maps during the 92,106-encounter evaluation. Online batches contain at most 16 same-user history examples and only the current user's adapter is updated. The dedicated revision study permits five rounds. Additional settings are in Appendix C.1.

5.2 RESULTS AND ANALYSIS

We compare direct models and tool agents (RQ1), test different backbones (RQ2), ablate the three modules (RQ3), track user adaptation (RQ4), and evaluate revision quality and rounds (RQ5). Global Judge + Regeneration provides a unified-critique correction baseline, while the appendix includes no-memory, frozen-global, shared-adapter, and per-user-adapter comparisons for personalization.

5.3 OVERALL COMPARISON (RQ1)

Table 1: Performance comparison of multimodal LLMs and agent frameworks. Each method is run on all 92,106 participant–instruction encounters; scores first average eligible users within each instruction and then macro-average the same 18 tasks. PAS is evaluated only on personalized instructions.

LLM Comparison						**Agent Framework Comparison**					
LLM	**GFS↑**	**EIC↑**	**MFC↑**	**PAS↑**	**Avg.↑**	**Agent Framework**	**GFS↑**	**EIC↑**	**MFC↑**	**PAS↑**	**Avg.↑**
GPT-4o	**8.25**	**8.11**	**8.00**	**7.61**	**7.99**	Global Judge + Regen	8.21	8.10	8.11	7.68	8.02
Gemini 3.1 Pro	7.96	7.84	7.90	7.39	7.77	Chameleon	8.37	8.17	8.09	7.74	8.09
Qwen3-Omni	7.76	7.63	7.69	7.29	7.59	HuggingGPT	8.35	8.20	8.09	8.03	8.17
Qwen3.5-Omni	8.14	7.97	7.83	7.53	7.87	ControlLLM	8.44	8.32	8.23	7.86	8.21
Qwen3.5-Plus	7.62	7.53	7.41	7.23	7.45	**MUSE (Ours)**	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

MUSE scores highest on Avg. (8.75) and PAS (8.59) in Table 1. Its direct Qwen3.5-Omni backbone scores 7.87 on Avg., below direct GPT-4o's 7.99. All methods in this table are evaluated on the same 18 tasks and the same 92,106 user–instruction encounters. The comparison suggests gains beyond backbone quality, although differences in verifier and tool calls preclude attributing the gap to a single component. Appendix C.8 reports native-policy call counts and latency, which should not be taken as matched-budget timings for the quality comparison. Task-level results appear in Appendix C.2.

5.4 BACKBONE GENERALIZATION (RQ2)

Table 2: Backbone generalization of MUSE across different multimodal models.

Backbone	**GFS↑**	**EIC↑**	**MFC↑**	**PAS↑**	**Avg.↑**
GPT-4o	**9.15**	**9.11**	**8.98**	**8.65**	**8.97**
Gemini 3.1 Pro	8.93	8.84	8.70	8.46	8.73
Qwen3-Omni	8.34	8.21	8.24	8.11	8.23
Qwen3.5-Omni	8.88	8.83	8.72	8.59	8.75
Qwen3.5-Plus	8.20	8.09	8.01	7.96	8.07

In Table 2, GPT-4o yields the highest Avg. (8.97), while Qwen3.5-Omni scores 8.75. Their PAS gap (0.06) is smaller than their GFS gap (0.27). FPV adapts satisfaction estimates from prior feedback without changing the task backbone, while backbone quality still affects the other metrics. These results cover the five tested backbones.

Table 3: Ablation study of MUSE components.

Setting	GFS↑	EIC↑	MFC↑	PAS↑	Avg.↑
MUSE w/o FUV	8.52	8.41	8.39	8.45	8.44
MUSE w/o CCV	8.47	8.55	8.18	8.43	8.41
MUSE w/o FPV	8.61	8.75	8.66	7.92	8.49
MUSE	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

5.5 Ablation Study (RQ3)

Table 3 shows distinct effects. Removing FUV reduces EIC from 8.83 to 8.41; removing CCV reduces MFC from 8.72 to 8.18. Removing FPV has its largest effect on PAS, which falls from 8.59 to 7.92. These changes match the modules' respective roles in local verification, cross-modal adjudication, and user adaptation. Because each ablation changes the full agent workflow, the differences include downstream effects of the removed module.

5.6 Continual Personalization Dynamics (RQ4)

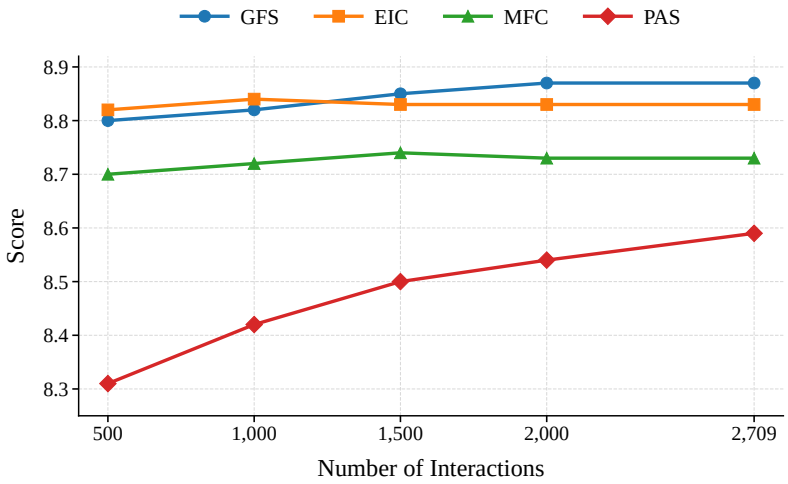


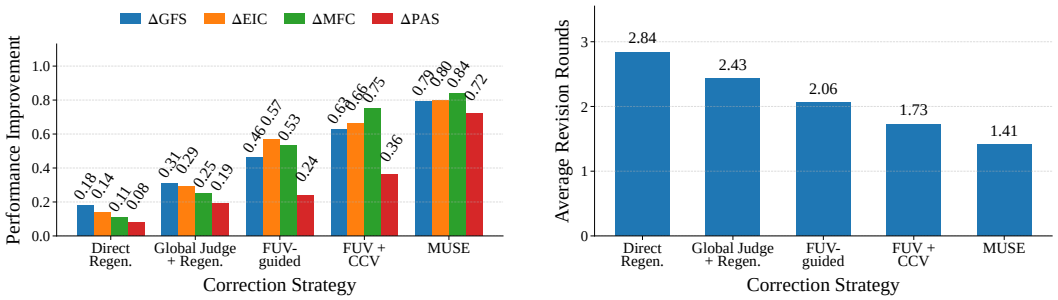
Figure 3: Task-macro performance over interactions with user-specific adaptation.

Figure 3 uses per-user interaction index on the horizontal axis: each participant contributes a 2,709-step trajectory, and the full method evaluation contains 92,106 encounters. At interaction t, a method receives only the same-user historical prefix available before t; its output is benchmark-scored before the next answer-linked historical record becomes available. In this sequential evaluation, PAS rises over the observed interaction range while GFS, EIC, and MFC change less. The pattern is consistent with FPV mainly learning user-specific satisfaction criteria rather than replacing FUV or CCV as general quality verifiers.

The curve averages the 34 participant trajectories while keeping their histories and adapters separate. Initially, positive-ϵ weighting includes the shared FPV prediction before user history is available; later answer-linked feedback records update only that user's adapter and can change FPV's relative weight. The masked-FPV cold-start experiment in Appendix Figure 5 uses a different decision rule. Appendices C.3 and C.5 report robustness and user isolation.

5.7 Revision Effectiveness Analysis (RQ5)

Figure 4 compares successive correction strategies under the same cap of five revision rounds. Direct regeneration and a global judge can initiate revisions but offer limited localization. FUV identifies modality-specific constraints and unsupported details; CCV then checks those reports against the



(a) Performance improvements under different correction strategies.

(b) Average revision rounds under different correction strategies.

Figure 4: Revision effectiveness under different correction strategies.

whole multimodal answer and attributes conflicts. In the displayed scores, adding FUV and CCV particularly improves EIC and MFC. FPV further uses a particular user's history to guide the targeted revision, giving MUSE the largest PAS improvement, 0.72, compared with 0.08 for direct regeneration in this revision analysis.

In this five-round revision study, MUSE uses 1.41 revision rounds on average, versus 2.84 for direct regeneration. Targeted instructions may help avoid repeated revisions, although the aggregate comparison cannot isolate each call's effect. FUV and CCV also incur verifier and judge calls, reported with latency in Appendix C.8.

Cost and feedback quality. The lower revision count does not imply lower latency: MUSE averages 10.21 model/tool calls and 24.3 seconds per encounter, versus 1.00 call and 6.2 seconds for direct Qwen3.5-Omni (Appendix Table 64). These native-policy costs are not matched-budget estimates for every Table 1 result. With 10% of feedback retained, PAS falls from 8.59 to 8.18; with 40% corrupted labels, it reaches 8.17 (Appendix Figure 7), showing degradation under sparse or noisy supervision.

Verifier and memory choices. Task-conditioned verifier selection reaches 8.75 Avg., compared with 8.70 using all verifiers and 8.59 with modality-only selection (Appendix Table 56). Four and five verifiers both reach 8.75, so the four-verifier setting uses fewer calls (Appendix Table 57). Compact memory with retrieval reaches 8.59 PAS, versus 8.34 for full-history encoding and 7.94 without memory (Appendix Table 50). These results support selective verification and retrieval, but not the contribution of every verifier or memory item.

Rating agreement. A separate expert-only agreement audit reports the highest agreement on EIC (ICC 0.89; Krippendorff's $\alpha = 0.84$) and the lowest on PAS (0.79; 0.72; Appendix Table 25). The feedback quantizer has 0.63 MAE on the 0–10 scale, with 88.6% of ratings within one point (Appendix Table 26). The five experts are also benchmark participants, so the roles are not disjoint.

6 Conclusion

MUSE combines modality-level verification, cross-modal adjudication, and feedback-driven user adaptation. Verification evidence guides immediate correction and, together with answer-linked user history, later personalization. Each main-comparison method is run on 2,709 instructions for 34 participants, giving 92,106 encounters across 18 task categories; scores first average the 34 participants for each applicable instruction and then macro-average the 18 tasks. The comparisons show gains in task quality and preference alignment, while verification adds calls and latency. The 800-example development collection is separate from the benchmark but serves both FPV initialization and score-map fitting, so it provides no held-out calibration result. Future work should test new users and tasks while reducing verifier cost.

REFERENCES

Ruisheng Cao, Fangyu Lei, Haoyuan Wu, Jixuan Chen, Yeqiao Fu, Hongcheng Gao, Xinzhuang Xiong, Hanchong Zhang, Yuchen Mao, Wenjing Hu, Tianbao Xie, Hongshen Xu, Danyang Zhang, Sida Wang, Ruoxi Sun, Pengcheng Yin, Caiming Xiong, Ansong Ni, Qian Liu, Victor Zhong, Lu Chen, Kai Yu, and Tao Yu. Spider2-V: How far are multimodal agents from automating data science and engineering workflows? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, Vancouver, Canada, December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/c2f71567cd53464161cab3336e8fc865-Abstract-Datasets_and_Benchmarks_Track.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 4171–4186, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Alexandre Drouin, Maxime Gasse, Massimo Caccia, Issam H. Laradji, Manuel Del Verme, Tom Marty, David Vazquez, Nicolas Chapados, and Alexandre Lacoste. WorkArena: How capable are web agents at solving common knowledge work tasks? In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11642–11662, Vienna, Austria, July 2024. PMLR. URL https://proceedings.mlr.press/v235/drouin24a.html.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, Tennessee, USA, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Fu_Video-MME_The_First-Ever_Comprehensive_Evaluation_Benchmark_of_Multi-modal_LLMs_in_CVPR_2025_paper.html.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, May 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination & visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14375–14385, Seattle, Washington, USA, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Guan_HallusionBench_An_Advanced_Diagnostic_Suite_for_Entangled_Language_Hallucination_and_CVPR_2024_paper.html.

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 (CVPR)*, pp. 14953–14962, Vancouver, Canada, June 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Gupta_Visual_Programming_Compositional_Visual_Reasoning_Without_Training_CVPR_2023_paper.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, Long Beach, California, USA, June 2019. PMLR.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4334–4353, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 881–905, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50. URL https://aclanthology.org/2024.acl-long.50/.

Reno Kriz, Kate Sanders, David Etter, Kenton Murray, Cameron Carpenter, Hannah Recknor, Jimena Guallar-Blasco, Alexander Martin, Eugene Yang, and Benjamin Van Durme. MultiVENT 2.0: A massive multilingual benchmark for event-centric video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, Tennessee, USA, June 2025.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13299–13308, Seattle, Washington, USA, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Li_SEED-Bench_Benchmarking_Multimodal_Large_Language_Models_CVPR_2024_paper.html.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. LLMs + persona-plug = personalized LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 9373–9385, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.461. URL https://aclanthology.org/2025.acl-long.461/.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153.

Zhaoyang Liu, Zeqiang Lai, Zhangwei Gao, Erfei Cui, Ziheng Li, Xizhou Zhu, Lewei Lu, Qifeng Chen, Yu Qiao, Jifeng Dai, and Wenhai Wang. ControlLLM: Augment language models with tools by searching on graphs. In *Proceedings of the 18th European Conference on Computer Vision (ECCV)*, volume 15070 of *Lecture Notes in Computer Science*, pp. 89–105, Milan, Italy, September–October 2024. Springer. doi: 10.1007/978-3-031-73254-6_6. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/1838_ECCV_2024_paper.php.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, New Orleans, Louisiana, USA, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/871ed095b734818cfba48db6aeb25a62-Abstract-Conference.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 46534–46594, New Orleans, Louisiana, USA, 2023. Curran Associates, Inc. doi: 10.52202/075280-2019.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, Vancouver, Canada, December 2019.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 7370–7392, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, New Orleans, Louisiana, USA, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/77c33e6a367922d003ff102ffb92b658-Abstract-Conference.html.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 8634–8652, New Orleans, Louisiana, USA, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

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 (ICCV)*, pp. 11888–11898, Paris, France, October 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Suris_ViperGPT_Visual_Inference_via_Python_Execution_for_Reasoning_ICCV_2023_paper.html.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Singapore, April 2025. URL https://openreview.net/forum?id=G0dksFayVq.

Jize Wang, Zerun Ma, Yining Li, Songyang Zhang, Cailian Chen, Kai Chen, and Xinyi Le. GTA: A benchmark for general tool agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, Vancouver, Canada, December 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8a75ee6d4b2eb0b777f549a32a5a5c28-Abstract-Datasets_and_Benchmarks_Track.html.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 2024b. URL https://openreview.net/forum?id=jp3gWrMuIZ.

Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExT-GPT: Any-to-any multimodal LLM. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53366–53397, Vienna, Austria, July 2024. PMLR.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Singapore, April 2025.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, Vancouver, Canada, December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. LLaVA-Critic: Learning to evaluate multimodal models. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, Tennessee, USA, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiong_LLaVA-Critic_Learning_to_Evaluate_Multimodal_Models_CVPR_2025_paper.html.

Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, Lars Liden, and Jianfeng Gao. Magma: A foundation model for multimodal AI agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14203–14214, Nashville, Tennessee, USA, June 2025.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57730–57754, Vienna, Austria, July 2024. PMLR. URL https://proceedings.mlr.press/v235/yu24o.html.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024)*, pp. 9556–9567, Seattle, Washington, USA, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yue_MMMU_A_Massive_Multi-discipline_Multimodal_Understanding_and_Reasoning_Benchmark_for_CVPR_2024_paper.html.

Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, Hang Yan, Jie Fu, Tao Gui, Tianxiang Sun, Yu-Gang Jiang, and Xipeng Qiu. AnyGPT: Unified multimodal LLM with discrete sequence modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 9637–9662, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.521.

AI USE STATEMENT

Generative AI tools assisted with manuscript editing, LaTeX formatting and organization. The authors are responsible for the paper's claims and content.

APPENDIX

CONTENTS

A Additional Method Details **15**
A.1 Verifier Report and Adjudication Details . . . 15
A.2 System Realization and FPV Learning Procedure . . . 15
A.3 Feedback Label Scoring Rubric . . . 17
A.4 Prompt Templates . . . 19
A.5 Multimodal Tool Library . . . 22
A.6 Reliability-Aware Decision Fusion . . . 24

B Benchmark Details **26**
B.1 Dataset Statistics and Distribution . . . 26
B.2 Task Categories and Modality Configurations . . . 28
B.3 Data Collection and Annotation . . . 37
B.4 Annotation Agreement . . . 41

C Additional Experimental Results **43**
C.1 Experimental Setup . . . 43
C.2 Per-Task Results on the 18 Benchmark Tasks . . . 46
C.3 Robustness to Sparse and Noisy User Feedback . . . 50
C.4 Memory Compression and Retrieval Strategy Analysis . . . 52
C.5 User-adapter Isolation and Cross-User Interference . . . 53
C.6 Verifier Selection and Coverage Analysis . . . 54
C.7 Supplementary Fusion Variants and Experimental Results . . . 55
C.8 Computational Fairness and Latency . . . 57
C.9 Verifier, Judge, and FPV Backbone Sensitivity . . . 58
C.10 Calibration Procedure and Decision Settings . . . 59
C.11 Cost–Quality Pareto Analysis . . . 59
C.12 Qualitative Case Study . . . 60

D Persistent User Memory **61**

E Online Evaluation and Additional Protocol Details **62**
E.1 Instruction Catalog and Per-User Interaction Order . . . 62
E.2 Decision Configurations and Result Interpretation . . . 63
E.3 Answer-Specific Feedback and Evaluation Histories . . . 63

A Additional Method Details

Related main text: Section 3.

This section describes the scheduler, hierarchical verification, FPV supervision, prompts, tool interfaces, and final decision fusion. Backbone and hyperparameter settings are given in Appendix C.1.

A.1 Verifier Report and Adjudication Details

Related main text: Sections 3.1–3.4.

The scheduler's task profile includes input and output modalities, explicit constraints, risks, and a preference context kept separate from verifier inputs. Its task-memory projection exposes prior task errors and correction traces. Selection and generation obey

$$A_t, U_t = \mathrm{Select}(q_t, \rho_t, M_t^{task}; G, V), \quad U_t = \{V_t^k\}_{k=1}^K, \qquad y_t = A_t(q_t, x_t). \tag{16}$$

The wrapper W_t^A derives a standardized task score, confidence, and rebuttal from tools without native self-evaluation. FUV receives the source, local output, and full-output context $\chi_{t,k} = (x_t^{(k)}, y_t^{(k)}, y_t)$; missing components are set to $\varnothing$. For each selected verifier,

$$q_t^k = R_k(q_t, \rho_t, A_t, M_t^{task}), \quad v_{t,k}^{(1)} = V_t^k(\chi_{t,k}, q_t^k, M_t^{task}). \tag{17}$$

The first-round report $v_{t,k}^{(1)} = \{d_{t,k}^{(1)}, s_{t,k}^{(1)}, e_{t,k}^{(1)}, c_{t,k}^{(1)}, \delta_{t,k}^{(1)}\}$ carries textual evidence, score, detected issues, confidence, and correction suggestions. Verifiers locate omission, ambiguity, redundancy, distortion, unsupported claims, and modality-specific constraint violations; historical subjective preferences remain outside their scope.

The wrapper assesses the candidate against these reports, and each verifier rechecks the rebuttal:

$$\begin{aligned}(r_t^A, s_t^A, \kappa_t^A) &= W_t^A.\mathrm{assess}(x_t, y_t, R_t^{(1)}, q_t, \rho_t, M_t^{task}),\\ v_{t,k}^{(2)} &= V_t^k.\mathrm{review}(\chi_{t,k}, v_{t,k}^{(1)}, r_t^A, q_t^k, M_t^{task}), \quad R_t^{(2)} = \{v_{t,k}^{(2)}\}_{k=1}^K,\\ D_t &= J_t(x_t, y_t, r_t^A, R_t^{(1)}, R_t^{(2)}, q_t, M_t^{task}).\end{aligned} \tag{18}$$

The rebuttal distinguishes genuine failures from acceptable trade-offs. The judge state D_t includes cross-modal consistency, rechecked finding validity, task-level validity, failure attribution, correction directions, score $s_t^{(2)}$, and confidence κ_t^J. CCV consolidates the evidence without asking FPV to examine raw media.

A.2 System Realization and FPV Learning Procedure

Related main text: Sections 3.1–3.5.

End-to-End Runtime. MUSE is implemented as a scheduler-centered, model-agnostic orchestration layer. At interaction (u, t), the scheduler parses the user request $q_{u,t}$ and optional multimodal input $x_{u,t}$, selects a primary task tool, and activates only the FUV verifiers relevant to the task profile. FUV first produces localized modality-level evidence. CCV then validates the local findings, resolves cross-modal conflicts, and packages the verified state as $D_{u,t}$. FPV interprets this state together with compact user memory to estimate personalized satisfaction. The scheduler finally applies the epsilon-smoothed reliability-weighted fusion and threshold decision in Section 3.6. Appendix A.6 restates this default and separately defines the optional masking and hard-gate variants. Thus, information flows in one direction from task execution to local verification, cross-modal adjudication, personalized interpretation, and final decision.

FPV Representation and User-Specific State. FPV receives

$$X_{u,t}^{(3)} = \left(q_{u,t}, \Sigma(y_{u,t}), D_{u,t}, \bar{M}_{u,t}^{\mathrm{lite}}\right), \tag{19}$$

where $\Sigma(y_{u,t})$ is a compact textual summary of the current multimodal output. The compact memory $\bar{M}_{u,t}^{\mathrm{lite}}$ retains stable preferences, recent satisfaction patterns, decision tendencies, and representative

positive and negative interactions. As in Section 3.5, a frozen BERT encoder maps the current evidence and each compact-memory item into a shared semantic space. Write the memory as $\{m_{u,t}^{\ell}\}_{\ell=1}^{L}$ and let each encoded item be a d-dimensional vector:

$$\begin{aligned} h_{u,t}^{q} &= E_{\mathrm{enc}}(q_{u,t}), & h_{u,t}^{y} &= E_{\mathrm{enc}}\big(\Sigma(y_{u,t})\big), \\ h_{u,t}^{d} &= E_{\mathrm{enc}}(D_{u,t}), & H_{u,t}^{m} &= [E_{\mathrm{enc}}(m_{u,t}^{\ell})]_{\ell=1}^{L}. \end{aligned} \tag{20}$$

The current evidence $H_{u,t}^{\mathrm{cur}} = [h_{u,t}^{q}; h_{u,t}^{y}; h_{u,t}^{d}] \in \mathbb{R}^{3\times d}$ queries the item sequence $H_{u,t}^{m} \in \mathbb{R}^{L\times d}$:

$$z_{u,t} = \mathrm{Pool}\left(\mathrm{LN}\left(H_{u,t}^{\mathrm{cur}} + \mathrm{MHA}\left(H_{u,t}^{\mathrm{cur}}, H_{u,t}^{m}, H_{u,t}^{m}\right)\right)\right). \tag{21}$$

The compact-memory budget counts serialized tokens across all items; it is distinct from the encoder input limit for an individual item. Overlength memory is compressed into concise summaries of key behaviors and actions before encoding, with each serialized item kept within the encoder limit. When memory is insufficient, relevant similar records may be retrieved only from the same user's authorized history. If no suitable record is available, FPV uses no historical memory: for $L = 0$, the attention contribution is zero, and residual fusion and pooling retain the current evidence. Retrieved context does not create feedback labels or supply another user's replay samples. The encoder, memory-attention module, and prediction head are shared across users, whereas each user maintains an independent bottleneck adapter ϕ_u. The personalized prediction is

$$\begin{aligned} \widetilde{z}_{u,t} &= z_{u,t} + W_{\mathrm{up}}^{u}\sigma\left(W_{\mathrm{down}}^{u} z_{u,t}\right), \\ s_{u,t}^{(3)} &= 10\,\sigma\left(W_s \widetilde{z}_{u,t} + b_s\right), \qquad s_{u,t}^{(3)} \in [0, 10]. \end{aligned} \tag{22}$$

Each user also maintains an independent full interaction memory $M_{u,t}^{\mathrm{full}}$, compact memory $\bar{M}_{u,t}^{\mathrm{lite}}$, and replay buffer $B_{u,t}$.

Per-User Adapter Storage. A bias-free user adapter contains $2 \cdot 768 \cdot 64 = 98{,}304$ parameters in its two bottleneck matrices. Stored as FP32 weights, it uses $98{,}304 \cdot 4 = 393{,}216$ bytes (0.375 MiB) per user; the shared encoder and prediction head are stored once.

Offline Initialization. FPV is initialized on 800 feedback examples from actual users whose satisfaction judgments are comparatively clear. These examples are separate from the 2,709 benchmark instructions and are used for 100 epochs to initialize shared memory attention, a generic adapter, and the prediction head on a common satisfaction scale. Each benchmark participant receives an independent copy of the generic adapter. Subsequent feedback updates only that user's copy; shared modules and the other users' adapters remain frozen. Let N be the number of eligible initialization interactions with $c_i \geq \tau_c$. Given their satisfaction labels $r_i \in [0, 10]$ and confidences $c_i \in [0, 1]$, the initialization objective is

$$\mathcal{L}_{\mathrm{off}} = \frac{1}{N}\sum_{i=1}^{N} c_i\,\mathrm{SmoothL1}\left(s_i^{(3)}, r_i\right). \tag{23}$$

Low-confidence samples remain available for auditing and memory analysis but are excluded from these N optimization samples. Optimization is skipped when no eligible sample is available.

Online Personalization. After the current output has been scored and returned or revised, the feedback quantizer converts newly observed feedback $f_{u,t}$ into a satisfaction label and confidence:

$$e_{u,t} = \left(X_{u,t}^{(3)}, f_{u,t}, r_{u,t}, c_{u,t}\right). \tag{24}$$

For valid feedback, FPV forms the same user-specific mixed batch as in Section 3.5, containing the current interaction and sampled recent and replayed interactions:

$$\begin{aligned} B_{u,t}^{\mathrm{train}} = \{e_{u,t}\} \cup \mathrm{Sample}_{n_r}(B_{u,t}^{\mathrm{recent}}) \cup \mathrm{Sample}_{n_p}(B_{u,t}^{\mathrm{replay}}), \\ 1 + n_r + n_p \leq 16. \end{aligned} \tag{25}$$

The target size is 16, with all available eligible samples used when same-user history is insufficient. History sampling is without replacement: the recent subset is chosen first, and the replay subset excludes those selected records. Both history pools contain only eligible samples from rounds

preceding t; low-confidence records retained in memory are not replay supervision. Thus, each historical record appears at most once, and the current sample is introduced only by the first term. For samples collected at round τ_i, the confidence- and recency-weighted loss is normalized by the sum of weights, exactly as in the main text:

$$\mathcal{L}_{\text{sup}}(\mathcal{E}) = \frac{\sum_{e_i \in \mathcal{E}} \omega_i \, \text{SmoothL1}(\hat{s}_i, r_i)}{\sum_{e_i \in \mathcal{E}} \omega_i}, \quad \omega_i = c_i \gamma^{t-\tau_i}. \tag{26}$$

Optimization requires a nonempty batch with positive total weight; otherwise the update is skipped. With eligible feedback, $c_i \geq \tau_c > 0$ and $\gamma \in (0, 1)$ ensure a positive denominator. The complete online objective is

$$\mathcal{L}^u_{\text{fpv}} = \mathcal{L}_{\text{sup}}(B^{\text{train}}_{u,t}) + \lambda_{\text{reg}} \|\phi_u - \phi^{\text{old}}_u\|_2^2. \tag{27}$$

Thus, current, recent, and replayed samples enter a single normalized objective; there are no separate λ_{recent} or λ_{replay} coefficients in the default method. The reference ϕ^{old}_u is the user's adapter immediately before the current update. Only the current user's adapter is updated. The shared modules and all other users' adapters remain frozen, and no samples from other users are introduced into the batch.

Temporal Causality and Reproducibility Boundary. The current interaction is evaluated before the next historical feedback record becomes available; feedback from interaction t can therefore influence only later decisions. In the benchmark replay, that feedback stays paired with its original answer and is never assigned to a compared method's new output. Encoder, optimizer, batch, threshold, decoding, and hardware settings are given in Appendix C.1.

A.3 Feedback Label Scoring Rubric

Related main text: Section 3.5.

The online learning procedure above requires a consistent target that separates task failure from user-specific dissatisfaction. We therefore convert heterogeneous explicit and implicit feedback into a common satisfaction label, confidence value, and structured reason code.

Feedback Sources. MUSE supports both explicit and implicit feedback. Explicit signals include numerical ratings, natural-language evaluations, correction requests, preference statements, and direct acceptance or rejection. Implicit signals include regeneration, repeated editing, interaction abandonment, output reuse, and other observable behaviors. The feedback quantizer jointly considers the original request, generated output, raw feedback, revision behavior, and available interaction context.

For each interaction, the quantizer produces a satisfaction label, confidence, and structured feedback category:

$$(r_{u,t}, c_{u,t}, g_{u,t}) = Q\left(q_{u,t}, y_{u,t}, f_{u,t}, b_{u,t}\right), \tag{28}$$

where $r_{u,t} \in \{0, 1, \ldots, 10\}$, $c_{u,t} \in [0, 1]$, $g_{u,t}$ denotes the dominant feedback category, $f_{u,t}$ is the raw feedback, and $b_{u,t}$ represents observed interaction behavior. The notation $Q(f_{u,t})$ in Section 3.5 abbreviates this context-conditioned quantizer and shows only the label and confidence used for optimization; the reason code is retained as metadata. The 0–10 label is used directly as FPV supervision, matching the satisfaction scale used throughout the paper.

Scoring Principles. The scoring procedure follows five principles. First, a direct numerical rating has the highest priority. Second, clear natural-language feedback is more reliable than behavioral inference alone. Third, the score reflects both the severity and scope of the issue. Fourth, conflicting signals receive a conservative score with reduced confidence. Fifth, the absence of negative feedback is not interpreted as satisfaction unless accompanied by an observable acceptance or reuse signal.

The operational mapping from feedback evidence to the upper and lower score bands is summarized in Tables 4 and 5, respectively.

Table 4: Feedback rubric for scores 6–10.

Score	Level	Typical Evidence
10	Fully satisfied	Fully correct, directly accepted or used, with no revision.
9	Highly satisfied	Clear approval with only an optional cosmetic adjustment.
8	Satisfied	Main result accepted with one minor local edit.
7	Generally satisfied	Main task completed; one limited issue remains.
6	Moderately satisfied	Usable output requiring several minor or one meaningful local revision.

Table 5: Feedback rubric for scores 0–5.

Score	Level	Typical Evidence
5	Mixed or neutral	Partially useful but requiring substantial restructuring.
4	Partially dissatisfied	A major issue affects an important component.
3	Dissatisfied	Several major defects require extensive rewriting.
2	Strongly dissatisfied	Core task is poorly completed; full regeneration requested.
1	Near-complete failure	Almost entirely irrelevant or incorrect, with little reusable content.
0	Complete failure	Empty, unrelated, unsafe, technically invalid, or wholly rejected output.

Together, Tables 4 and 5 separate minor preference adjustments from task-level failures while preserving the common 0–10 supervision scale.

Feedback Categories and Conflict Resolution. The structured categories include acceptance, content error, explicit instruction violation, modality-specific defect, cross-modal conflict, preference mismatch, formatting issue, safety issue, and insufficient evidence. When multiple signals are available, MUSE applies the priority

$$\begin{aligned}\text{Rating} > \text{Textual Feedback} > \text{Correction Behavior} \\ > \text{Implicit Behavior}.\end{aligned} \tag{29}$$

A high rating accompanied by a request for complete regeneration is treated as ambiguous rather than strongly positive. Cosmetic wording or formatting changes generally correspond to scores of 8–9, localized content or preference corrections to 6–7, major restructuring to 3–5, and complete rejection caused by fundamental failure to 0–2.

Partial and Modality-Specific Feedback. When feedback targets only one modality, the quantizer considers whether that modality is central to task completion. A local defect in a secondary modality receives a moderate penalty, whereas failure of the primary requested modality produces a substantially lower score. Cross-modal conflicts are scored according to their effect on the usability and coherence of the complete output.

Confidence Estimation. Explicit ratings accompanied by consistent textual explanations receive high confidence. Clear natural-language acceptance or rejection receives medium-to-high confidence. Labels inferred only from behavior receive lower confidence because the same action may have multiple causes. Conflicting or incomplete signals receive low confidence and do not trigger online optimization when $c_{u,t} < \tau_c$.

Representative conversions from raw feedback and behavior to labels are reported in Table 6.

Table 6: Representative examples of feedback quantization.

Feedback or Behavior	Score	Conf.
Exactly satisfies the request	10	High
Accepted with a minor title edit	8	High
Correct content but mismatched tone	6	High
Cross-modal semantic conflict	4	High
Complete regeneration requested	2	High
Accepted without textual feedback	9	Med.
Regenerated without explanation	3	Low
Conflicting rating and comments	5	Low

As Table 6 illustrates, explicit and internally consistent signals receive both more extreme labels and higher confidence, whereas ambiguous behavioral signals remain conservative.

A.4 Prompt Templates

Related main text: Sections 3.1, 3.3, 3.4, and 3.6.

The evidence and labels consumed by FUV, CCV, FPV, and the scheduler are produced through standardized prompts with explicit input fields and output schemas. MUSE uses structured prompts for task scheduling, fine-grained verification, cross-modal adjudication, feedback quantization, and targeted revision. The prompts presented below show the common templates used across experiments. Task-specific information, tool descriptions, modality names, and interaction records are inserted into the corresponding fields at runtime. Unless otherwise stated, all modules are instructed to return concise structured outputs rather than unrestricted natural-language judgments.

Scheduler and Task Profiling Prompt. The scheduler first converts the user request and optional multimodal inputs into a task profile. The profile distinguishes explicit task constraints and verification risks from historical subjective preferences. Only the task-specific projection is exposed to FUV, CCV, and the task-tool wrapper; preference context remains available to FPV and scheduler-directed revision.

> **Role.** You are the scheduler of a personalized multimodal agent. Analyze the current request and construct an executable task plan.
>
> **Inputs.** User request: *[user request]*
>
> Optional multimodal inputs: *[input descriptions]*
>
> Compact user memory: *[retrieved user preferences]*
>
> Available task tools: *[generation and understanding tools]*
>
> Available verification tools: *[verification tool descriptions]*
>
> **Instructions.** Identify the task type, required input and output modalities, explicit constraints, likely user preferences, and potential failure risks. Select one primary task tool and all verification tools required to evaluate its output. Do not select a verifier that is unrelated to the requested modalities. Separate mandatory constraints from optional preferences. Store historical subjective preferences in a distinct field and exclude that field from verifier-facing instructions and the task-memory projection.
>
> **Output.** Return the task type, input modalities, output modalities, primary tool, selected verifiers, mandatory constraints, optional preferences, risk items, and execution plan.

FUV Instruction Reformulation Prompt. Before invoking a modality-specific verifier, the scheduler reformulates the original request into a verification instruction specialized for the target modality. This prevents a verifier from judging irrelevant properties or overlooking modality-specific constraints.

> **Role.** You reformulate a multimodal user request into a precise verification instruction for one target modality.
>
> **Inputs.** Original request: *[user request]*
>
> Task profile: *[scheduler output]*
>
> Target modality: *[text, visual, or audio]*

Primary tool description: *[tool profile]*

Task-specific memory: *[retrieved task failures and correction traces]*

Instructions. Extract only requirements that can be verified within the target modality. Preserve all explicit constraints, including style or format requirements stated in the current request. Add relevant task-failure patterns and successful correction traces from task-specific memory. Exclude historical subjective preferences; FPV evaluates their effect on satisfaction. Do not introduce requirements unsupported by the current request or verified task evidence. Do not evaluate cross-modal consistency in this step.

Output. Return the target modality, verification objectives, mandatory constraints, explicitly requested properties, likely failure types, and required evidence.

Fine-Grained Unimodal Verification Prompt. FUV uses a common verification structure across modalities, while the detailed inspection criteria are determined by the selected verifier.

Role. You are a fine-grained verifier specialized in the target modality. Evaluate the candidate output according to the reformulated instruction.

Inputs. Original request: *[user request]*

Verification instruction: *[reformulated instruction]*

Candidate output: *[target-modality output]*

Task-specific memory: *[retrieved task failures and correction traces]*

Instructions. Inspect the candidate for instruction compliance, semantic correctness, modality-specific quality, unsupported content, and mismatches with style or format constraints explicitly stated in the current request. Do not use historical subjective preferences as verification criteria. Localize each issue and provide observable evidence. Distinguish a hard constraint violation from a soft quality issue. Do not infer cross-modal conflicts that cannot be established from the target modality alone.

Output. Return an overall modality score in the range $[0, 10]$, satisfied requirements, detected issues, evidence for each issue, severity, confidence, and a concise correction suggestion.

For understanding tasks, the verifier receives the source modality together with the primary agent's response. It checks whether claims in the response are supported by the source input. For generation tasks, the verifier directly evaluates the generated modality against the reformulated instruction.

Primary Agent Rebuttal Prompt. After the initial FUV reports are produced, the scheduler-side task-tool wrapper explains or contests the reported issues using the tool output and available metadata. This standardized interface also supports task tools without native self-evaluation. The step reduces false alarms caused by incomplete modality-level inspection.

Role. You are the scheduler-side wrapper for the primary task tool. Review the candidate output, tool metadata, and initial verification reports, then provide an evidence-based rebuttal.

Inputs. User request: *[user request]*

Candidate output summary: *[output summary]*

Initial FUV reports: *[modality-level reports]*

Instructions. For each reported issue, state whether you agree, partially agree, or disagree. Support the response using evidence from the candidate output and the original request. Acknowledge valid errors directly. Do not revise the candidate output in this step. Do not dismiss an issue without explicit evidence.

Output. Return one response for each reported issue, including the position, supporting evidence, and remaining uncertainty.

Verifier Second-Review Prompt. Each FUV verifier receives the primary agent's rebuttal and performs a second review before CCV adjudication.

Role. You are the original modality-specific verifier. Reassess your initial findings after reviewing the primary agent's rebuttal.

Inputs. Original verification report: *[initial FUV report]*

Primary-agent rebuttal: *[rebuttal]*

Original request: *[user request]*

Candidate output: *[candidate output]*

Instructions. Determine whether each initial issue should be confirmed, revised, or withdrawn. Consider the rebuttal only when it is supported by observable evidence. Preserve valid findings even when the primary agent disagrees. Clearly identify unresolved uncertainty.

Output. Return the final status of each issue, updated evidence, updated severity, confidence, and any newly detected modality-specific problem.

CCV Adjudication Prompt. CCV integrates the initial FUV reports, the primary-agent rebuttal, and the second-review results. Its role is to validate modality-level findings and explicitly identify cross-modal conflicts.

Role. You are a cross-modal adjudicator. Produce a unified evaluation from the available modality-level evidence and rebuttal process.

Inputs. User request: *[user request]*

Task profile: *[scheduler output]*

Output summary: *[multimodal output summary]*

Initial FUV reports: *[initial reports]*

Primary-agent rebuttal: *[rebuttal]*

Verifier second reviews: *[updated reports]*

Instructions. Validate or reject the modality-specific findings based on the complete evidence. Identify semantic conflicts, factual inconsistencies, style mismatches, redundancy, omission, and coordination failures across modalities. Determine whether explicit task constraints are satisfied globally. Do not treat the primary agent's self-assessment as decisive. Prioritize issues according to their effect on task success and output usability. Assess style only against current explicit instructions and cross-modal coherence; exclude historical subjective preferences from this adjudication.

Output. Return validated strengths, validated modality-specific issues, cross-modal conflicts, satisfied and violated constraints, unresolved uncertainty, an overall CCV score in the range $[0, 10]$, and prioritized revision targets.

Feedback Quantization Prompt. The feedback quantizer transforms heterogeneous user feedback into the satisfaction label described in Appendix A.3.

Role. You convert user feedback and interaction behavior into a calibrated satisfaction label for continual personalization.

Inputs. User request: *[user request]*

Output summary: *[output summary]*

Structured verification evidence: *[CCV evidence]*

Raw user feedback: *[rating or natural-language feedback]*

Observed behavior: *[acceptance, editing, regeneration, abandonment, or reuse]*

Relevant interaction context: *[user history]*

Instructions. Assign an integer satisfaction score from 0 to 10 using the predefined scoring rubric. Prioritize explicit ratings over textual judgments, textual judgments over correction behavior, and correction behavior over implicit behavior. Consider the severity and scope of the reported issue. When signals conflict or remain ambiguous, assign a conservative score and reduce confidence. Do not infer strong satisfaction only from the absence of negative feedback.

Output. Return the satisfaction score on the 0–10 scale, confidence in the range $[0, 1]$, feedback categories, concise evidence, preference information that may be stored in memory, and whether the sample is reliable enough for online updating.

Final Decision Prompt. The scheduler determines whether to accept the current output or initiate another revision round by combining the task-agent score, FUV evidence, CCV evidence, FPV satisfaction prediction, and the maturity of the personalized predictor.

Role. You make the final acceptance or revision decision for the current interaction.

Inputs. Primary-agent score: *[score]*

Aggregated FUV score: *[score]*

CCV score and evidence: *[score and structured evidence]*

FPV satisfaction prediction: *[predicted satisfaction]*

FPV maturity: *[maturity value]*

Current revision round: *[round index]*

Instructions. For the default method, use the epsilon-smoothed weights in Section 3.6 and accept exactly when the fused score reaches τ_s. A zero FPV maturity still gives a positive smoothed weight, not an activation mask. Treat verified failures as evidence for the scores and revision targets, without an additional deterministic veto. When an explicitly identified hard-gate variant is requested, also apply its gate. When revision is required, identify the verified problems that must be corrected. Do not request another revision after the maximum revision budget has been exhausted.

Output. Return the decision, fused confidence, decisive evidence, unresolved issues, and whether another revision round is permitted.

Targeted Revision Prompt. When the current output is rejected, the scheduler converts the verified evidence into a targeted instruction for the primary tool. FPV contributes an overall user-alignment signal, while the concrete correction targets are primarily derived from FUV, CCV, and retrieved user memory.

Role. You construct a targeted revision instruction for a multimodal generation or understanding tool.

Inputs. Original request: *[user request]*

Current output summary: *[output summary]*

Validated FUV issues: *[modality-specific issues]*

Validated CCV issues: *[cross-modal issues]*

Relevant user preferences: *[retrieved memory]*

Current revision round: *[round index]*

Instructions. Preserve all correct and accepted content. Correct each validated issue using explicit and actionable instructions. Resolve cross-modal conflicts jointly rather than revising each modality independently. Satisfy applicable user preferences without overriding mandatory task constraints. Avoid generic instructions such as "improve the answer" or "regenerate the output." Do not introduce unrelated changes.

Output. Return the content to preserve, the issues to correct, modality-specific revision actions, cross-modal coordination requirements, applicable personalization requirements, and explicit completion criteria.

A.5 Multimodal Tool Library

Related main text: Section 3.1.

MUSE represents task execution through a registry of capability interfaces. Each interface specifies a functional operation and its input/output modalities; multiple interfaces may share the same underlying model or software backend.

Library Construction. The task-tool registry contains 435 interfaces in 10 functional categories. A fixed Core-50 subset is used in the controlled experiments, while the remaining 385 interfaces are retained as extended capabilities and are not introduced into the reported comparisons. Verification is handled by a separate 33-interface registry, giving 468 interfaces in the complete system.

We use T, I, V, A, and D for text, image, video, audio, and document inputs. At the benchmark level, image and video are grouped as the visual modality V; the tool registry keeps them separate because static and temporal visual processing require different operations.

Table 7: Task-tool interfaces by functional category. "Core" denotes the fixed subset used in the controlled experiments.

Category	All	Core	Ext.	Representative Interfaces and I/O
Omnimodal	15	8	7	`omni_understand` (T/I/V/A → T/I/V/A/JSON); `omni_reason` (T/I/V/A → T/I/V/A/JSON); `omni_generate` (T/I/V/A → T/I/V/A/JSON)
Text	50	8	42	`text_question_answering` (T → T/JSON); `text_summarization` (T → T/JSON); `text_generation` (T/JSON → T/JSON)
Image	83	14	69	`image_caption` (I/T → T/JSON); `object_detection` (I/T → I/Mask/JSON); `image_inpainting` (I/T → I)
Video	70	10	60	`video_caption` (V/T → T/JSON/Frames); `video_object_tracking` (V/A/T → T/JSON/V); `text_to_video` (T/I/V/A → V)
Audio	72	6	66	`speech_recognition` (A → T/JSON); `audio_caption` (A → T/JSON); `text_to_speech` (T/A → A)
Cross-modal	40	2	38	`text_to_image_retrieval` (T/I/V/A → Ranked items/JSON); `audio_to_video_retrieval` (T/I/V/A → Ranked items/JSON); `audio_video_alignment` (T/I/V/A → Score/JSON)
Document	30	1	29	`pdf_parse` (D/I/T → T/JSON/D); `xlsx_parse` (D/I/T → T/JSON/D); `document_question_answering` (D/I/T → T/JSON/D)
Retrieval	25	0	25	`web_search` (T/I/V/A/Query → Results/JSON); `vector_search` (T/I/V/A/Query → Results/JSON); `evidence_retrieval` (T/I/V/A/Query → Results/JSON)
Utility	30	1	29	`python_execute` (T/Code/Data/Media → T/Data/Media); `video_processing_ffmpeg` (T/Code/Data/Media → T/Data/Media); `resource_cost_estimator` (T/Code/Data/Media → T/Data/Media)
Personalization	20	0	20	`user_preference_extraction` (T/History/Feedback → Profile/Memory/Label); `memory_update` (T/History/Feedback → Profile/Memory/Label); `replay_sample_selection` (T/History/Feedback → Profile/Memory/Label)
Total	**435**	**50**	**385**	–

Coverage of the 18 Benchmark Tasks. The Core-50 task interfaces are selected jointly rather than assigning an independent tool set to every benchmark task. A capability interface can therefore be reused across multiple categories. Speech recognition, for example, can support audio question answering, meeting-minutes generation, multimodal summarization, and audio editing; cross-modal retrieval and alignment interfaces can support retrieval, audio-visual reasoning, conflict detection, and joint generation or editing. Tables 8–10 summarize the benchmark-level task and modality coverage.

Table 8: Tool coverage for understanding and reasoning tasks.

ID	Task	I/O
1	Visual Question Answering	$V + T \rightarrow T$
2	Audio Question Answering	$A + T \rightarrow T$
3	Audio-Visual Question Answering	$A + V + T \rightarrow T$
4	Multimodal Fact Verification	$A + V + T \rightarrow T$
5	Cross-modal Conflict Detection	$A + V + T \rightarrow T$
6	Multi-source Multimodal Reasoning	$A + V + T \rightarrow T$

Table 9: Tool coverage for retrieval and transformation tasks.

ID	Task	I/O
7	Text–Visual Retrieval	$T \rightarrow V$
8	Text–Audio Retrieval	$T \rightarrow A$
9	Audio–Visual Retrieval	$A \rightarrow V$
10	Meeting Minutes Generation	$A + V + T \rightarrow T$
11	Multimodal Content Summarization	$A + V \rightarrow T$
12	Multi-source Information Integration	$A + V + T \rightarrow T$

Table 10: Tool coverage for generation and editing tasks.

ID	Task	I/O
13	Text-to-Visual Generation	$T \to V$
14	Text-to-Audio Generation	$T \to A$
15	Text-to-Audio-Visual Generation	$T \to A + V$
16	Instruction-Guided Visual Editing	$V + T \to V$
17	Instruction-Guided Audio Editing	$A + T \to A$
18	Instruction-Guided Audio-Visual Editing	$A + V + T \to A + V$

Verification Tool Pool. The verification registry is separate from task execution. It contains 22 modality-specific FUV interfaces, six CCV interfaces, three FPV-related interfaces, and two safety interfaces. The FPV entries denote functions surrounding the single shared predictor with user-specific adapters rather than separate FPV models.

Table 11: Verification interfaces by family and representative function.

Family	No.	Primary Coverage	Representative Interfaces
FUV–Text	6	Factuality, constraints, relevance, coherence, style, hallucination	`text_factuality_verifier`; `text_constraint_verifier`
FUV–Image	6	Fidelity, prompt alignment, objects, spatial relations, quality, edit preservation	`image_prompt_alignment_verifier`; `image_edit_preservation_verifier`
FUV–Video	5	Temporal consistency, prompt alignment, motion, identity, event correctness	`video_temporal_consistency_verifier`; `video_event_correctness_verifier`
FUV–Audio	5	Content, style, speaker properties, quality, speech–text alignment	`audio_content_verifier`; `speech_text_alignment_verifier`
CCV	6	Cross-modal semantics, conflicts, redundancy, completeness, synchronization, style	`cross_modal_conflict_verifier`; `audio_visual_sync_verifier`
FPV	3	Satisfaction prediction, preference compliance, personalization calibration	`user_satisfaction_predictor`; `preference_compliance_verifier`
Safety	2	General safety and privacy	`safety_policy_verifier`; `privacy_verifier`
Total	**33**	**Hierarchical, personalized, and safety verification**	

Tool Selection During Execution. The scheduler does not invoke all registered tools for every interaction. It selects one primary task interface from the fixed Core-50 experimental pool and a task-dependent subset of supporting and verification interfaces. Historical preferences may guide task-tool choice and preference-aware revision, whereas FUV and CCV selection criteria use only current explicit constraints, required modalities, and objective task-risk records. Extended task interfaces are excluded from the controlled comparisons. The controlled comparison protocol governing shared backends, decoding budgets, and invocation accounting is described in Appendix C.1.

A.6 Reliability-Aware Decision Fusion

Related main text: Section 3.6.

After task execution, FUV, CCV, and FPV have produced complementary but heterogeneous signals. The default final stage calibrates these signals and combines them with the epsilon-smoothed reliability weights in Section 3.6. Explicit cold-start masking and hard-constraint gating are additional variants, not components of the default method.

Functionally Distinct Signals. The scheduler combines four signals with complementary roles: the wrapper-derived task score s_t^A, the aggregated FUV score $s_t^{(1)}$, the CCV adjudication score $s_t^{(2)}$, and the FPV satisfaction prediction $s_t^{(3)}$. They respectively represent task completion, modality-local quality and instruction compliance, cross-modal consistency, and user-specific satisfaction. As specified in Section 3.4, CCV also assesses the validity of local findings and the task-level implications of cross-modal evidence. These signals can share evidence and are not assumed statistically independent. Their distinct primary roles organize the evidence; calibration and confidence weighting do not by themselves prove that all overlap has been removed.

Because external task tools do not always expose comparable self-evaluation scores, a scheduler-side wrapper produces a standardized task score and confidence:

$$(r_t^A, s_t^A, \kappa_t^A) = W_t^A.\text{assess}\left(x_t, y_t, R_t^{(1)}, q_t, \rho_t, M_t^{\text{task}}\right). \tag{30}$$

Here, r_t^A is the evidence-grounded response of the task tool to the verifier findings, $s_t^A \in [0, 10]$ measures task completion, and $\kappa_t^A \in [0, 1]$ denotes confidence.

FUV Aggregation and Score Calibration. The FUV score is aggregated according to verifier confidence:

$$s_t^{(1)} = \frac{\sum_{k=1}^K c_{t,k}^{(1)} s_{t,k}^{(1)}}{\sum_{k=1}^K c_{t,k}^{(1)} + \epsilon}. \tag{31}$$

We use $\epsilon = 0.5$, matching the fusion weights below. For total verifier confidence $C_t = \sum_k c_{t,k}^{(1)}$, the raw FUV aggregate lies in $[0, 10C_t/(C_t + 0.5)]$ and is zero when $C_t = 0$. Thus, this constant shrinks the raw aggregate as well as preventing a zero denominator. Although all four module scores use a 0–10 scale, their empirical distributions can differ. We use one learned monotonic calibration function for each signal:

$$\bar{s}_t^{(i)} = \text{Cal}_i\left(s_t^{(i)}\right), \quad \bar{s}_t^{(i)} \in [0, 1], \quad i \in \mathcal{I} = \{A, 1, 2, 3\}. \tag{32}$$

The current decision uses only calibration parameters fitted before the answer is evaluated. Appendix C.10 describes the separate 800-example score–rating pairs.

Reliability and FPV Maturity. The reliability of the task wrapper, FUV, and CCV is estimated from their current confidence:

$$m_t^{(A)} = \kappa_t^A, \qquad m_t^{(1)} = \frac{1}{K}\sum_{k=1}^K c_{t,k}^{(1)}, \qquad m_t^{(2)} = \kappa_t^J. \tag{33}$$

FPV reliability additionally depends on the amount of valid user feedback and its recent prediction accuracy. Let $\bar{r}_\tau = r_\tau/10$, $\widetilde{s}_\tau^{(3)} = s_\tau^{(3)}/10$, and let $\mathcal{W}_t$ contain only valid feedback from rounds preceding the current decision. The maturity error compares normalized satisfaction predictions with satisfaction labels; the calibrated acceptance score $\bar{s}_\tau^{(3)}$ is reserved for decision fusion. We define

$$\begin{aligned} e_t^{(3)} &= \begin{cases} \dfrac{1}{|\mathcal{W}_t|}\displaystyle\sum_{\tau\in\mathcal{W}_t}\left|\widetilde{s}_\tau^{(3)} - \bar{r}_\tau\right|, & |\mathcal{W}_t| > 0, \\ 1, & |\mathcal{W}_t| = 0, \end{cases} \\ \text{Rel}_t^{(3)} &= \text{clip}\left(1 - e_t^{(3)}, 0, 1\right). \end{aligned} \tag{34}$$

and

$$m_t^{(3)} = \min\left(1, \frac{N_t}{N_0}\right)\text{Rel}_t^{(3)}, \tag{35}$$

where N_t is the number of valid feedback samples and N_0 is the reference count at which the feedback-volume factor saturates.

Default Reliability-Aware Weights and Final Decision. The default method uses the same smoothing constant $\epsilon = 0.5$ as Section 3.6:

$$\begin{aligned} w_{i,t} &= \frac{m_t^{(i)} + \epsilon}{\sum_{j\in\mathcal{I}}(m_t^{(j)} + \epsilon)}, \quad i \in \mathcal{I}, \\ S_t &= \sum_{i\in\mathcal{I}} w_{i,t}\bar{s}_t^{(i)}. \end{aligned} \tag{36}$$

There is no activation mask in this default. At cold start, $m_t^{(3)} = 0$, but $w_{3,t} = 0.5/(\sum_{j\in\mathcal{I}} m_t^{(j)} + 2)$ remains positive. Because the other three reliabilities lie in $[0, 1]$, this cold-start weight lies in

$[0.10, 0.25]$. If all reliabilities are zero, the weights are exactly $1/4$, so the denominator and fused score remain defined. The default decision is

$$\text{Decision}_t = \begin{cases} \text{Accept}, & S_t \geq \tau_s, \\ \text{Revise}, & S_t < \tau_s. \end{cases} \tag{37}$$

When revision is required and the iteration budget remains, the scheduler constructs a targeted instruction from verified evidence, compact memory, and the personalized satisfaction signal. Exhausting the budget terminates further revision; it does not change whether the acceptance threshold has been reached. The current interaction is scored before its feedback is used for online updating, so feedback from round t can affect only future interactions.

Additional Cold-Start Masking Variant. For the separate masking analysis in Appendix C.7, an activation mask suppresses FPV when its maturity is zero:

$$a_{i,t} = \begin{cases} 0, & i = 3 \text{ and } m_t^{(3)} = 0, \\ 1, & \text{otherwise.} \end{cases} \tag{38}$$

The corresponding unsmoothed variant uses

$$w_{i,t}^{\text{mask}} = \frac{a_{i,t} m_t^{(i)}}{D_t^{\text{mask}}}, \qquad D_t^{\text{mask}} = \sum_{j \in \mathcal{I}} a_{j,t} m_t^{(j)}, \qquad D_t^{\text{mask}} > 0.$$

It yields exactly zero FPV weight at cold start and lets its influence grow with maturity. Unlike the default, this unsmoothed expression has no defined score when $D_t^{\text{mask}} = 0$; such a state cannot justify acceptance and requires additional verification or an explicit failure report. This boundary condition is part of the variant's definition, not an empirical claim that such a state occurred in the reported runs.

Additional Hard-Constraint Gate Variant. A weighted average can retain a high score despite a severe failure. The separate hard-gate variant supplements a selected fusion rule with

$$H_t = \mathbb{I}\left[\mathcal{E}_t^{\text{hard}} \neq \varnothing\right], \tag{39}$$

where $\mathcal{E}_t^{\text{hard}}$ includes confirmed explicit-instruction violations, factual errors, missing primary modalities, severe cross-modal conflicts, safety or privacy violations, and failures to preserve required content in editing tasks. Under this additional gate, acceptance requires both $H_t = 0$ and the selected fusion score reaching τ_s; otherwise revision is requested subject to the remaining budget. "Masked + Gated Fusion" in Appendix C.7 denotes this additional configuration rather than the default MUSE decision rule.

Boundedness. For the default, $m_t^{(i)} \in [0, 1]$ and $\epsilon > 0$ make every weight positive with $\sum_i w_{i,t} = 1$. Since each calibrated score lies in $[0, 1]$, their weighted sum satisfies $S_t \in [0, 1]$, including the all-zero-reliability case. The masked variant has the same convex-combination bound only when $D_t^{\text{mask}} > 0$; its zero FPV weight at cold start and the gate's veto of critical failures are properties of the respective additional variants, not guarantees of the default method.

B Benchmark Details

Related main text: Section 4.

B.1 Dataset Statistics and Distribution

Overall Composition. The benchmark contains 2,709 distinct instructions across 18 tasks and three capability groups. Thirty-four participants each complete all 2,709 instructions, and every evaluated method is run once for every participant–instruction pair. Thus the instruction catalog contains 2,709 unique requests, whereas one complete method evaluation contains 92,106 encounter outputs. Table 12 reports both counting units.

Table 12: Benchmark composition and participant counts.

Statistic	Count
Total samples	2,709
Task categories	18
Capability groups	3
Supported modalities	T, V, A
Personalized samples	1,340
Non-personalized samples	1,369
Expert participants	5
Graduate-student participants	17
Other student participants	12
Total participants	34
Participant–instruction encounters per method	92,106

The benchmark has no fixed training, validation, or test split. Online evaluation follows a separate anonymous per-user interaction log; the catalog row order is not used as chronology. For a given participant and interaction index, all compared methods receive the same instruction and the same prefix of answer-linked historical feedback, and the current output is evaluated before the next history item can affect later interactions (Appendix E.1).

Task-Group Distribution. The group sizes are determined by the actual sample counts of their six constituent tasks. Understanding and reasoning contains 917 samples, retrieval and information transformation contains 913, and generation and editing contains 879.

The sample allocation across the three capability groups is reported in Table 13.

Table 13: Distribution across the three task groups.

Task Group	Tasks	Samples
Understanding and reasoning	6	917
Retrieval and information transformation	6	913
Generation and editing	6	879
Total	**18**	**2,709**

Table 13 shows that the three capability groups have similar sample counts. The macro-average gives equal weight to each applicable task category, independently of these counts.

Modality Distribution. Table 14 aggregates tasks sharing the same benchmark-level input and output modality configuration. The counts sum to the complete benchmark size.

Table 14: Distribution by modality configuration.

Input	Output	Samples
$V+T$	T	137
$A+T$	T	172
$A+V+T$	T	996
$A+V$	T	128
T	V	298
T	A	283
A	V	91
T	$A+V$	178
$V+T$	V	147
$A+T$	A	194
$A+V+T$	$A+V$	85
Total		**2,709**

Personalization Distribution. Personalization spans all 18 task categories. The instruction catalog contains 1,340 personalized and 1,369 non-personalized instructions. Table 15 gives the instruction counts.

Table 15: Personalization flags in the instruction catalog. These are instruction counts, not counts of participant feedback events.

Instruction type	Count
Personalized	1,340
Non-personalized	1,369
Total	**2,709**

Chronological Online Use. All 2,709 samples belong to the online evaluation collection. There are no separate training and test subsets. For user u at interaction t, only earlier same-user interactions and feedback enter the memory and adapter state. The new output receives benchmark scores before the next historical answer–feedback pair is released; replay draws only from eligible earlier same-user feedback.

B.2 Task Categories and Modality Configurations

The benchmark contains 18 task categories covering multimodal understanding, retrieval, information transformation, generation, and editing. The tasks are divided into three groups of six according to their principal capability requirements.

We use T, V, and A to denote textual, visual, and audio modalities, respectively. The visual modality V includes both images and videos at the benchmark level. The specific visual input type is determined by each sample. Personalization is not treated as an independent task category; instead, user-specific preferences and interaction histories may be attached to samples from any of the 18 tasks.

Multimodal Understanding and Reasoning. The first group evaluates whether an agent can extract, align, verify, and reason over information from one or more modalities. These tasks primarily produce textual outputs, allowing the benchmark to evaluate whether generated claims are supported by the provided multimodal evidence. Table 16 lists these tasks.

Table 16: Understanding and reasoning tasks.

ID	Task	I/O	No.
1	Visual Question Answering	$V + T \rightarrow T$	137
2	Audio Question Answering	$A + T \rightarrow T$	172
3	Audio-Visual Question Answering	$A + V + T \rightarrow T$	109
4	Multimodal Fact Verification	$A + V + T \rightarrow T$	198
5	Cross-modal Conflict Detection	$A + V + T \rightarrow T$	86
6	Multi-source Multimodal Reasoning	$A + V + T \rightarrow T$	215

Retrieval and Information Transformation. The second group evaluates whether an agent can retrieve semantically matching content across modalities or transform heterogeneous source information into a structured textual result. The retrieval tasks require the system to identify content whose semantics correspond to a query expressed in another modality. The information-transformation tasks require extraction, compression, organization, and integration of multimodal evidence. Table 17 presents the six tasks in this group.

Table 17: Retrieval and information-transformation tasks.

ID	Task	I/O	No.
7	Text–Visual Retrieval	$T \rightarrow V$	142
8	Text–Audio Retrieval	$T \rightarrow A$	164
9	Audio–Visual Retrieval	$A \rightarrow V$	91
10	Meeting Minutes Generation	$A + V + T \rightarrow T$	203
11	Multimodal Content Summarization	$A + V \rightarrow T$	128
12	Multi-source Information Integration	$A + V + T \rightarrow T$	185

Multimodal Generation and Editing. The third group evaluates whether an agent can produce or modify multimodal content under explicit task constraints and user-specific preferences. Generation tasks require the system to create one or more output modalities from a textual instruction. Editing tasks require the system to modify only the requested content while preserving unaffected regions, events, identities, or acoustic properties. Joint audio-visual tasks additionally require semantic and temporal coordination between the generated or edited modalities. Table 18 summarizes this group.

Table 18: Generation and editing tasks.

ID	Task	I/O	No.
13	Text-to-Visual Generation	$T \rightarrow V$	156
14	Text-to-Audio Generation	$T \rightarrow A$	119
15	Text-to-Audio-Visual Generation	$T \rightarrow A + V$	178
16	Instruction-Guided Visual Editing	$V + T \rightarrow V$	147
17	Instruction-Guided Audio Editing	$A + T \rightarrow A$	194
18	Instruction-Guided Audio-Visual Editing	$A + V + T \rightarrow A + V$	85

Representative User Instructions. Tables 19–21 show three non-personalized and three personalized requests for each task. Requests are shortened for readability; the preference column records user history available before the current request.

Table 19: Example requests for understanding and reasoning (N: non-personalized; P: personalized).

Task	Type	Current request	Prior preference
1 Visual Question Answering			
1	N	On the contour map, determine the slope direction between points A and B and identify the steeper side. Cite the relevant contour lines.	—
1	N	Using the exploded-view diagram, identify the labeled component, where it sits, and which adjacent parts it connects to. Limit the answer to 80 words.	—
1	N	In the road-camera video, when does the white vehicle cross the stop line, and is its turn signal on at that moment? Give the relevant timestamps.	—
1	P	Read the two dataset scores from the results figure and calculate the method's improvement over the baseline. Give the two values and the difference.	Prefers the main result before the supporting figures.
1	P	Identify the marked component in the exploded-view diagram and explain its connection to the two neighboring parts.	Prefers compact tables and brief explanations of technical terms.
1	P	In the surveillance video, find the first frame where the white vehicle crosses the stop line and determine whether its indicator was already on.	Earlier feedback: a visible location or timestamp beside each factual claim.
2 Audio Question Answering			
2	N	Listen to the bilingual explanation. Retain the English technical terms and explain what each means and how the procedure works.	—
2	N	In the news interview, what is the interviewee's central claim? Summarize two reasons given in the recording, without adding unstated facts.	—
2	N	Identify the lead instrument in the music excerpt, any tempo change, and the point of greatest increase in volume. Give approximate timestamps.	—
2	P	From the customer-service call, extract the order number, promised resolution time, and issue that remains open.	Prefers the answer first, followed by the precise statement that supports it.
2	P	In the three-person discussion, who proposes a small-scale test, and how do the other two speakers respond?	Prefers a compact table with short explanations of technical terms.
2	P	Use the maintenance recording to reconstruct the operator's actions and identify the stage at which the failure is first mentioned.	Earlier feedback: timestamps for claims drawn from recordings.
3 Audio-Visual Question Answering			
3	N	In the remote-meeting recording, does the chart element on screen match the one the speaker names? Give the audio and video times.	—
3	N	Compare the demonstrator's spoken description with the color change visible in the laboratory video. Report any discrepancy and its timestamp.	—
3	N	In the cooking video, what does the cook do immediately after the timer sounds? Use the sound and visible action to support the answer.	—
3	P	In the industrial-operation video, describe how the device changes state after the warning beep. Distinguish the audible signal from the visible change.	Prefers a direct finding before a short account of the evidence.
3	P	Compare the alert sound with the pedestrian signal in the traffic video. Estimate the delay between the two events.	Prefers a compact table when comparing audio and visual events.

Task	Type	Current request	Prior preference
3	P	In the performance video, check whether the music change coincides with the lighting change and the performer's entrance.	Earlier feedback: time ranges when describing alignment between sound and video.
4 Multimodal Fact Verification			
4	N	Check the claim that the signal stayed green before the incident. Use the video, audio, and written record to give a verdict with evidence locations.	—
4	N	Check whether the new method beats the baseline on every dataset in the supplied results. Identify datasets that support or contradict the claim.	—
4	N	Evaluate the claim that staff evacuated immediately after the alarm. Separate the alarm time from the first visible movement of staff.	—
4	P	Verify the claim that the venue and start time did not change. Examine the announcement, recording, and schedule separately.	Prefers a verdict before the supporting and conflicting evidence.
4	P	Check the claim that the product demonstration had no failures. Identify any interruption shown or mentioned in the supplied materials.	Prefers a compact table of claims, evidence, and verdicts.
4	P	Assess the claim that every student completed all required experimental steps. Note what the footage or records do not establish.	Earlier feedback: a timestamp or visible location for each cited step.
5 Cross-modal Conflict Detection			
5	N	Compare the subtitles with the original speech and visible events in the video. List actual contradictions and keep missing details separate.	—
5	N	Compare the sports footage, stadium audio, and match report for differences in the score, foul, or event time. Give the source of each discrepancy.	—
5	N	Check the travel video, route map, and narration for conflicting place names, directions, or arrival order. Identify which source differs.	—
5	P	Compare the experiment footage, observation notes, and spoken result for differences in color change, temperature, and conclusion.	Prefers the overall finding before the individual conflicts and their evidence.
5	P	Compare the door sound, the person's entrance, and the timestamps in the script. Report any discrepancy of at least half a second.	Prefers short comparison tables with plain-language explanations.
5	P	Compare the meeting captions, shared screen, and original audio for differences in the agreed decision, owner, and deadline.	Earlier feedback: the time and source of each reported conflict.
6 Multi-source Multimodal Reasoning			
6	N	Use the experiment log, procedure video, and device audio to reconstruct the failure sequence from the first abnormal signal. Identify any unsupported link.	—
6	N	Combine the incident report, scene photographs, and call recording to order the events. Mark causal explanations that lack direct support.	—
6	N	Compare the environmental measurements, satellite image, and interviews to assess the proposed sources. Keep observation and explanation distinct.	—
6	P	Compare the two product documents, demonstration video, and support call on battery life, startup time, and limitations; recommend a suitable use case.	Prefers the recommendation first and a brief explanation of the evidence afterward.
6	P	Read the paper, results figure, and author Q&A. Does the evidence support the paper's main conclusion? Identify its limits.	Prefers short comparison tables and plain-language explanations of research terms.

6	P	Use the traffic logs, surveillance video, and field audio to reconstruct how the congestion developed. Distinguish observed events from possible causes.	Earlier feedback: timestamps or source locations when reconstructing events.

Table 20: Example requests for retrieval and information transformation (N: non-personalized; P: personalized).

Task	Type	Current request	Prior preference
7 Text–Visual Retrieval			
7	N	Find a continuous shot of a white city bus entering a seaside promenade and stopping beside a sign. Return the best match and identify the stop in the clip.	—
7	N	Find footage of a goggled student at a laboratory bench putting an object down before opening a device. Rank the top three clips by whether both actions occur in that order.	—
7	N	Retrieve a forest-trail clip showing three runners of different ages moving from right to left. Return two candidates and explain their positions and direction of travel.	—
7	P	Find the two best clips of a cyclist in a yellow raincoat turning back at an early-morning market. Exclude clips with readable advertisements.	Prefers a brief best-match summary before the detailed ranking.
7	P	Find the two best clips of two workers operating the same device in a school corridor. Check the action and the camera view in each result.	Earlier feedback: repeated ranking evidence to be merged without losing distinct details.
7	P	Find a clip of three runners of different ages approaching the camera in an early-morning market. Exclude clips where the runners look into the camera.	Prefers ranking decisions supported by specific visual matches.
8 Text–Audio Retrieval			
8	N	Retrieve two recordings of a customer-service employee speaking in a natural conversational voice, without motor-vehicle noise. Compare voice style and background sound over a continuous 60-second interval.	—
8	N	Find two recordings of steady machinery followed by a single brief abnormal sound. Give the onset of that sound and inspect at least 30 continuous seconds in each recording.	—
8	N	Retrieve two recordings of steady rain without thunder or speech. Identify a continuous 30-second interval in each and explain any differences in the rain and background sounds.	—
8	P	Find three city recordings with a bicycle bell and distant construction, but no speech. Check a continuous 60-second interval in each and remove any candidate with speech.	Prefers rankings backed by specific audible events and their locations in the recording.
8	P	Find three recordings of restrained piano with soft strings and no sudden loud noise. Check at least 30 continuous seconds in each candidate.	Prefers the best match stated first, followed by short comparisons of the remaining candidates.
8	P	Retrieve three recordings of steady machinery followed by one brief abnormal sound, without background music. Check a continuous 40-second interval in each and report when the abnormal sound occurs.	Prefers one description of recurring background sounds, with each anomaly timestamped.
9 Audio–Visual Retrieval			
9	N	Find the video matching an audio clip of keyboard typing, a notification, and a printer starting. Give the times of the matching visual events.	—

Task	Type	Current request	Prior preference
9	N	Use a recording of a doorbell, approaching footsteps, and a lock opening to find two matching videos. Rank the candidates by event order, not merely by the appearance of the doorway.	—
9	N	Retrieve two videos matching ocean waves, rope friction, and distant seagulls. For each, identify which sounds have visible sources and give the relevant timestamps.	—
9	P	Match a recording of a glass set on a table and a door opening about two seconds later to two videos. Compare event order and cite the relevant timestamps.	Prefers explicit time-based evidence for each ranked match.
9	P	Find three videos matching an animal call, shaking branches, and rapid running in that order. Distinguish a visible animal from an animal inferred only from the sound.	Likes a short best-match conclusion before the evidence for the other candidates.
9	P	Find three videos matching a machine beep followed by a metal component stopping. Mark the sound and visible action times without assuming the source of an off-screen sound.	Prefers each distinct sound and visible action paired with its own timestamp.
10 Meeting Minutes Generation			
10	N	Write minutes for a safety review from its recording, shared slides, chat, and agenda. Record safety decisions, risks, action owners, and deadlines; mark any unassigned item as pending.	—
10	N	Prepare minutes for an experiment-planning meeting on the launch date and acceptance criteria. Separate confirmed decisions from proposals and cite a speaking time or slide for each decision.	—
10	N	Summarize a supply-chain meeting on delivery scope and ownership using its audio, shared screen, chat, and agenda. Include action items and unresolved delays, without assigning an owner who was not named.	—
10	P	Prepare minutes for an experiment-planning meeting on user feedback and the next iteration. Include confirmed decisions, owners, deadlines, and unresolved issues, using only what was known by the end of the meeting.	Prefers a short summary before the decision and action-item lists.
10	P	Write minutes for a supply-chain meeting on delivery scope and ownership. Preserve distinct viewpoints in the recording and chat, and mark an unconfirmed deadline as pending.	Previously asked for duplicate points across audio, slides, and chat to be merged without losing unique details.
10	P	Record the launch-date discussion and acceptance criteria from a course-design meeting. Distinguish decisions actually approved from proposed dates and unresolved criteria.	Prefers confirmed statements, indirect support, and open questions in separate sections.
11 Multimodal Content Summarization			
11	N	Summarize an experiment demonstration in three to five bullets: question, procedure, observations, and limits of the evidence. Use what is shown and said without repeating the same event twice.	—
11	N	Summarize a sports clip's score changes, key plays, and crowd response. Give timestamps for up to five moments and separate the commentator's claims from actions visible on screen.	—
11	N	Summarize a training video, including required steps, incorrect examples, and safety warnings. Label any instruction heard in the narration but not demonstrated visually.	—

Task	Type	Current request	Prior preference
11	P	Summarize a meeting screen recording, covering discussion results, figures shown on screen, and follow-up items. Keep disagreements and unanswered questions visible.	Prefers a one-sentence overview before the supporting points.
11	P	Summarize a travel video with its main route and up to five timestamped moments. Use both narration and footage, and keep the account within 250 words.	Previously asked for one account of an event appearing in both narration and footage.
11	P	Summarize a news scene, identifying what happened, who made each claim, and what the camera actually shows. Note details supported by only one modality.	Prefers verified observations and uncorroborated statements to be presented separately.
12 Multi-source Information Integration			
12	N	Combine sales records, store footage, and customer feedback into a report on sales changes and customer behavior. Take figures from the sales records and attribute any suggested cause to its source.	—
12	N	Integrate logistics notes, warehouse footage, and driver calls into a delay report. Identify where the delay occurred, separate confirmed causes from hypotheses, and list corrective actions.	—
12	N	Combine experiment logs, result figures, and researcher comments into a report on conditions, results, anomalies, and limits of the conclusion. Cite the figure for each numerical result.	—
12	P	Combine teaching records, performance charts, and classroom audio into a report on learning outcomes and difficult topics. Take all numerical results from the charts and attribute proposed actions to their speakers.	Prefers a brief summary of the findings before the source-by-source detail.
12	P	Compare Plan A and Plan B using proposal documents, cost charts, and review audio. Report cost, risks, delivery time, and any disagreement between sources.	Prefers shared claims consolidated, with source-specific qualifications retained.
12	P	Integrate a safety report, surveillance video, and alarm records into an event timeline. Separate observed response actions from possible explanations and identify risks still unresolved.	Prefers direct evidence, indirect support, and missing evidence to appear in distinct parts of the report.

Table 21: Example requests for generation and editing (N: non-personalized; P: personalized).

Task	Type	Current request	Prior preference
13 Text-to-Visual Generation			
13	N	Create a 10-second portrait video of an empty reading desk at night, lit by a small lamp beside an open book. Keep people and branding out of frame.	—
13	N	Draw a portrait infographic tracing evaporation, condensation, rainfall, and runoff around the water cycle. Keep the four stages legible and omit branding.	—
13	N	Create a landscape image of two students in goggles working on the same device in a naturally lit lab. Show both students and the device clearly, without logos.	—
13	P	Create a 16:9 image of a rainy junction, with a cyclist in a yellow raincoat on the right and a red bus behind. Include no text.	Chooses subdued colours and sparse backgrounds.
13	P	Animate the water cycle in a square frame, moving from evaporation to condensation, rainfall, and runoff. Leave the lower-right corner empty.	Prefers restrained colours and minimal ornament.

Task	Type	Current request	Prior preference
13	P	Make a landscape image for a riverside night run, with distant lights and room for a later title at the lower right.	Earlier feedback: no extra labels, branding, or decorative elements.
14 Text-to-Audio Generation			
14	N	Compose 12 seconds of slow piano and light strings, with no drums, vocals, or abrupt swell.	—
14	N	Produce a 20-second device alert: a brief tone within the first three seconds, spoken guidance in the middle, and a two-second completion tone at the end. Keep tones separate from speech.	—
14	N	Record the warning 'The device is starting. Do not touch moving components.' in a calm but firm voice within 15 seconds, without shouting.	—
14	P	Record 'The road ahead is wet. Please slow down.' as a short safety warning, with a clear but calm emphasis.	Prefers a restrained delivery over dramatic effects.
14	P	Compose 12 seconds of slow piano with light strings. Omit strong drums, vocals, and abrupt crescendos.	Prefers sparse, low-key arrangements.
14	P	Create a 12-second device alert with an opening tone, spoken guidance, and a closing tone. Keep speech and tones apart.	Recent feedback: no background music or decorative sounds unless requested.
15 Text-to-Audio-Visual Generation			
15	N	Make a 25-second plant-growth lesson showing germination, leaves, and flowering in order. Explain each stage as it appears on screen.	—
15	N	Make a 40-second riverside night-run film: runners assemble, start, follow the river, and reach the finish. Match narration and captions to each stage.	—
15	N	Make a lab-safety demonstration in which a student puts on goggles, adds reagent, observes the reaction, and closes the valve. Keep the spoken steps in the same order.	—
15	P	Make a 25-second pedestrian-safety film showing a person stop, look both ways, and cross as the narration says 'Stop, look, and cross safely.'	Prefers low-saturation visuals and uncluttered frames.
15	P	Produce a short equipment tutorial showing power-on, calibration, a test run, and a success display. Align the spoken steps and captions with the video.	Earlier feedback: no extra music, branding, or text beyond the requested captions.
15	P	Make an 18-second night-run film in which runners gather, start, follow the river, and cross the finish. Place each sound and caption with its matching action.	Repeatedly flags narration or sound effects that arrive before the corresponding on-screen event.
16 Instruction-Guided Visual Editing			
16	N	Remove the red bin from the lower-left corner of the street photo and reconstruct the road. Leave pedestrians, cars, buildings, and framing as they are.	—
16	N	On the uploaded poster, replace 'June 18' with 'June 28' in the same font and perspective. Preserve every other word and the background.	—
16	N	Change only the sky visible through the window to a cloudy blue; adjust its reflection while keeping the people and furniture in place.	—
16	P	Mark the specified route on the uploaded map with a thin line and start and end markers. Keep roads, names, and scale legible.	Favours restrained annotations with little visual decoration.
16	P	Change the product shell from black to white in the photo. Keep its shape, logo, reflections, and background intact.	Has repeatedly asked that edits preserve objects and details outside the requested region.

Task	Type	Current request	Prior preference
16	P	Soften the background behind the pet in the video. Leave its fur, movement, and the video's length unchanged.	Recent feedback: no added text, branding, or embellishment.
17 Instruction-Guided Audio Editing			
17	N	Remove the horn around second 20 from the outdoor recording, keeping the birds, wind, and continuous ambience audible.	—
17	N	Lower the music beneath the narration until the words are clear. Preserve the speaker's voice, musical phrasing, and duration.	—
17	N	Reduce wind and distant traffic in the outdoor interview without flattening the speech or erasing the sense of place.	—
17	P	Replace 'Wednesday' with 'Thursday' near second 12 of the recording. Keep the same voice and background, and leave the timing intact.	Expects corrections to stay within the affected word and preserve the rest of the recording.
17	P	Reduce keyboard clicks in the meeting recording without cutting spoken words, natural breaths, or pauses.	Prefers light processing that leaves voices sounding natural.
17	P	Correct the wrong model name in the product-demonstration audio, preserving the speaker's rhythm and all other words.	Recent feedback: no added music, effects, or promotional sound cues.
18 Instruction-Guided Audio-Visual Editing			
18	N	Replace the old chart in the meeting recording and correct the matching spoken value. Keep gestures, captions, and speech aligned.	—
18	N	Remove the false beep from the experiment video and correct the warning light shown at that moment. Leave the other machine sounds and steps intact.	—
18	N	Remove the dog crossing the right side of the outdoor clip and its two barks. Preserve the people's actions, ambient sound, camera movement, and length.	—
18	P	Blur background bystanders in the interview video and suppress their identifiable speech. Leave the interviewee, main dialogue, and captions clear.	Insists that privacy edits leave the main speaker and unaffected parts of a clip intact.
18	P	Turn the product shell white in the demonstration, and change 'black version' to 'white version' in both speech and captions.	Has asked for no added branding, extra captions, or background music in product edits.
18	P	Replace the travel video's rapid montage with no more than three cuts. Keep the narration, event order, and overall duration aligned.	Repeatedly notices edits that shift dialogue or change events outside the requested sequence.

Personalization Across Tasks. The personalized requests use only preferences established before the current interaction. An explicit requirement in the current request takes precedence when it conflicts with that history.

Evaluation Dimensions. All 18 tasks are evaluated using the four metrics defined in the main paper. **Goal Fulfillment Score (GFS)** measures whether the requested task is successfully completed. **Explicit Instruction Compliance (EIC)** measures adherence to explicit content, format, modality, style, length, and preservation requirements. **Multimodal Faithfulness and Coherence (MFC)** jointly evaluates grounding in the provided evidence or requested content and semantic, temporal, and stylistic consistency across modalities. **Preference-Aligned Satisfaction (PAS)** measures alignment with relevant user preferences and historical feedback without overriding the current explicit instruction.

Each metric is scored on a 0–10 scale. Within a task, Avg. is the arithmetic mean of GFS, EIC, MFC, and PAS after each metric is averaged over its applicable instructions. All main-comparison methods are executed on the same 18 task categories. For each applicable instruction, its 34 participant scores

are averaged first; overall results then macro-average the resulting task scores, as in Section 4. PAS uses only the 1,340 personalized instructions and is not assigned to non-personalized instructions.

B.3 Data Collection and Annotation

Data Sources. The MUSE benchmark is constructed from multimodal materials collected from real-world application scenarios, publicly available or authorized data resources, and manually designed task instances. The collection process focuses on practical multimodal agent use cases rather than isolated perception problems. The source materials include textual documents, images, videos, speech recordings, environmental audio, meeting records, and combinations of these modalities.

The collected materials are converted into task instances according to the definitions in Appendix B.2. Understanding and reasoning samples are constructed by pairing multimodal evidence with questions, claims, or reasoning instructions. Retrieval samples contain a query and a candidate pool from another modality. Information-transformation samples require the system to summarize, reorganize, or integrate heterogeneous sources. Generation samples contain detailed content and style requirements, while editing samples additionally provide source media and preservation constraints.

User requests are collected from real interaction scenarios or manually rewritten while preserving their original task intention and explicit requirements. Personalization feedback comes from actual system use by five experts, 17 graduate students, and 12 other students, totaling 34 participants. Real-user feedback collection is distinct from instruction preparation and the benchmark scoring procedure. Per-user sample assignment and interaction order are specified in Appendix E.1.

Related work evaluates video understanding and multimodal retrieval through Video-MME Fu et al. (2025) and MultiVENT 2.0 Kriz et al. (2025). Our custom benchmark focuses on the task combinations, constraints, and feedback histories described here; the reported results concern this collection, and no performance on those external benchmarks is claimed. The source components and instance-construction procedure for each task group are summarized in Table 22.

Table 22: Data construction by task group.

Group	Construction Procedure
Understanding and reasoning	Pair grounded questions or claims with designated multimodal evidence; multi-source items require complementary evidence.
Retrieval and transformation	Construct cross-modal query–candidate sets with hard negatives, or require compression and integration of heterogeneous sources.
Generation and editing	Specify target content and controllable attributes; editing instances also define preservation constraints.

Table 22 highlights that each group is constructed with a different evidence requirement: grounded reasoning, cross-modal matching or transformation, and controlled generation or preservation.

Sample Structure. A general benchmark instance is represented as

$$d_i = (q_i, x_i, m_i, a_i)\,, \tag{40}$$

where q_i is the user request, x_i denotes the optional multimodal input, m_i specifies the task and modality configuration, and a_i contains the human annotation criteria. Depending on the task, x_i may contain one or more textual, visual, or audio inputs.

For retrieval tasks, the instance additionally contains a candidate set:

$$d_i^{\mathrm{ret}} = (q_i, x_i, \mathcal{C}_i, m_i, a_i)\,, \tag{41}$$

where $\mathcal{C}_i$ contains the candidate visual or audio items. For editing tasks, a_i includes both modification requirements and preservation constraints.

A personalized instance further contains the user context available before the current interaction:

$$d_{u,t}^{\mathrm{per}} = \left(q_{u,t}, x_{u,t}, m_{u,t}, a_{u,t}, M_{u,t}^{\mathrm{ctx}}\right), \tag{42}$$

where $M_{u,t}^{\mathrm{ctx}}$ may contain explicit preferences, previous interactions, earlier revisions, and historical feedback. Feedback produced during or after the current interaction is not included in $M_{u,t}^{\mathrm{ctx}}$.

Each method is executed separately for every (u, t) encounter while receiving the same prefix of that participant's answer–feedback history. Historical feedback remains paired with the exact answer that originally received it and is never reassigned as a label for another method's newly generated response. Each method output receives its own benchmark evaluation, including PAS when applicable; this participant-confirmed benchmark score is distinct from the answer-linked feedback label used for FPV adaptation.

Candidate Output Collection. For evaluation, every compared method produces one output for each of the 92,106 participant–instruction encounters. The resulting outputs include both high-quality and imperfect examples. Common defects include omitted constraints, unsupported textual claims, incorrect visual objects, inaccurate audio content, temporal inconsistencies, failed preservation during editing, and conflicts between generated modalities.

The collected candidate outputs are not treated as gold-standard answers. Instead, they provide realistic evaluation examples for constructing scoring criteria, validating the automatic evaluator, and identifying modality-specific and cross-modal error categories. For open-ended generation tasks, evaluation is based on whether the output satisfies the request and annotation criteria rather than whether it matches a single reference output.

Benchmark Evaluation Dimensions. Each encounter output is evaluated using the same four metrics reported in the main experiments. Encounter-level participant ratings use integers from 0 to 10 and follow the same metric definitions:

- **Goal Fulfillment Score (GFS):** whether the output completes the intended task and provides a practically usable result;
- **Explicit Instruction Compliance (EIC):** whether the output follows explicit content, format, modality, style, length, and preservation requirements;
- **Multimodal Faithfulness and Coherence (MFC):** whether the output is grounded in the provided evidence or requested content and remains semantically, temporally, and stylistically coherent across modalities;
- **Preference-Aligned Satisfaction (PAS):** whether the output follows applicable user preferences and historical feedback without overriding the current explicit instruction.

PAS is scored only for samples containing valid personalization context. For non-personalized samples, PAS is marked as not applicable rather than assigned a score of zero. The benchmark PAS score is an evaluation metric, whereas the feedback label $r_{u,t}$ is supervision derived from answer-linked post-interaction feedback for FPV training; the two share the same 0–10 range but serve different purposes.

Metric-Specific Scoring Rubric. Table 23 defines the common score anchors. Evaluators assign integer scores by considering the severity, scope, and task criticality of each defect. The rubric is anchor-based rather than a mechanical deduction scheme, allowing the same defect to receive different penalties when it affects a core task requirement rather than a secondary detail.

Table 23: Metric-specific 0–10 scoring rubric used in the MUSE benchmark.

Score	GFS	EIC	MFC	PAS
9–10	The requested task is fully or almost fully completed and the result is directly usable.	All explicit instructions are satisfied, with at most a negligible presentation deviation.	Claims and generated content are well grounded; modalities are mutually consistent with at most negligible ambiguity.	The output follows the current request and all applicable stable preferences, with at most a negligible mismatch.

Continued on next page

Table 23 (continued)

Score	GFS	EIC	MFC	PAS
7–8	The main task is completed, but a minor omission or local defect slightly reduces utility.	Most instructions are satisfied; one minor constraint is imperfectly realized without affecting the core result.	The output is generally faithful and coherent, with a minor unsupported detail or weak local alignment.	The output is broadly aligned with the user, but one secondary preference is missed or only partially satisfied.
5–6	The main objective is partially achieved, but noticeable omissions or errors require revision before use.	One important constraint or several minor constraints are violated, although the output remains partly usable.	The output contains a noticeable grounding error or cross-modal inconsistency, but substantial correct content remains.	The output captures some preferences but misses an important applicable preference or shows moderate over-personalization.
3–4	A major subgoal fails and only a limited portion of the result is reusable.	Multiple important instructions are violated or a required property is substantially incorrect.	Major unsupported content, semantic conflict, temporal mismatch, or failed preservation substantially harms the result.	Multiple applicable preferences are ignored, or personalization conflicts with the current request in an important way.
1–2	The task is largely unsuccessful and the output contains little usable content.	Most explicit instructions are ignored, including one or more critical requirements.	The output is pervasively ungrounded or inconsistent across modalities and is largely unreliable.	The output strongly conflicts with established preferences or repeatedly applies irrelevant preferences.
0	The output is empty, irrelevant, invalid, or does not attempt the requested task.	A mandatory modality, format, safety requirement, or other defining instruction is completely absent.	The output is unrelated to the evidence or contains fundamental contradictions that make it unusable.	The output is completely incompatible with the available user context or overrides the current request.

Scoring and Annotation Procedure. The benchmark uses an LLM-assisted, participant-reviewed evaluation workflow. The LLM evaluator receives the task definition, user request, multimodal inputs, candidate output, applicable user context, and the metric-specific rubric. For each applicable metric, it returns a preliminary integer score, localized supporting evidence, detected error types, severity, and confidence. These preliminary scores structure the review process but are not directly treated as final benchmark labels.

For the main-text experiments, all 34 participants take part in evaluation. For each encounter, the participant receiving the method output reviews the preliminary metric scores against the original request, multimodal evidence, generated output, and common rubric, and confirms or revises the scores. The five domain experts supervise rubric interpretation across the study and review ambiguous or contested cases; they are not treated as five independent raters for every output. The resulting participant-confirmed encounter score is denoted $s_{u,i,m} \in \{0, 1, \ldots, 10\}$ for user u, instruction i, and metric $m \in \{\mathrm{GFS}, \mathrm{EIC}, \mathrm{MFC}, \mathrm{PAS}\}$, with PAS included only when personalization context is available.

The complete workflow is summarized in Table 24.

Score Aggregation. The aggregation preserves the user–instruction evaluation unit. Because every instruction is completed by all $U = 34$ participants, we first average the encounter scores for instruction i:

$$\bar{s}_{i,m} = \frac{1}{U} \sum_{u=1}^{U} s_{u,i,m}, \qquad U = 34. \tag{43}$$

For task t, let $\mathcal{I}_{t,m}$ be the instructions for which metric m is applicable. The task score is then

$$S_{t,m} = \frac{1}{|\mathcal{I}_{t,m}|} \sum_{i \in \mathcal{I}_{t,m}} \bar{s}_{i,m}. \tag{44}$$

Table 24: Scoring and annotation procedure of the MUSE benchmark.

Stage	Operation
1	Verify the task type, required modalities, input completeness, and metric applicability.
2	Extract explicit instructions, applicable user preferences, grounding evidence, and preservation requirements.
3	Generate or collect the candidate output under the controlled task setting.
4	Produce preliminary 0–10 scores, localized evidence, error categories, severity, and confidence using the LLM evaluator.
5	Review and, when needed, revise the preliminary scores and evidence by the participant assigned to the encounter.
6	Review ambiguous or contested cases under domain-expert supervision and harmonize rubric interpretation across participants.
7	Aggregate participant-confirmed scores and check consistency, duplicate records, interaction order, and feedback leakage.

Finally, all methods use the same 18-task set $\mathcal{T}$ and the overall metric is the equal-weight task macro-average

$$S_m = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} S_{t,m}, \qquad |\mathcal{T}| = 18. \tag{45}$$

PAS is defined only for personalized instructions, so $\mathcal{I}_{t,\mathrm{PAS}}$ contains the personalized subset of task t; GFS, EIC, and MFC use all instructions. Avg. is the arithmetic mean of the four resulting metric macro-averages. Within each task, averaging first over the 34 users and then over eligible instructions equals the mean across that task's eligible encounters. Across the full benchmark, GFS, EIC, and MFC each cover $34 \times 2{,}709 = 92{,}106$ encounters, while PAS covers $34 \times 1{,}340 = 45{,}560$. The overall metric is a task macro-average, so tasks receive equal weight regardless of their instruction counts.

Error Annotation. In addition to scalar scores, the benchmark records structured error labels. Modality-specific errors include unsupported textual content, missing constraints, incorrect visual entities, inaccurate spatial relations, speech-content errors, speaker mismatches, audio-style errors, visual artifacts, and temporal defects. Cross-modal errors include semantic contradiction, information redundancy, missing correspondence, audiovisual desynchronization, style inconsistency, and conflicting entity or event descriptions.

Each detected error is associated with its affected modality, severity, supporting evidence, and applicable evaluation dimension. This structured annotation supports both final-output evaluation and the independent analysis of FUV and CCV error-detection capabilities.

Personalization Annotation. For personalized samples, annotators distinguish explicit preferences from preferences inferred from interaction history. A historical preference is considered applicable only when it is sufficiently stable and relevant to the current task. Current explicit requirements always take precedence over historical preferences.

Annotators additionally determine whether the output:

- follows the relevant historical preference;
- ignores an applicable preference;
- incorrectly applies an irrelevant preference;
- over-personalizes the output and violates the current request; or
- lacks sufficient historical evidence for personalization.

The PAS annotation therefore evaluates appropriate preference use rather than simply measuring whether more personalized content is produced.

Quality Control. Multiple quality-control procedures are applied before finalizing the benchmark. First, samples with incomplete inputs, unclear instructions, or unverifiable annotations are removed or revised. Second, near-duplicate samples are identified using textual and multimodal similarity checks. Third, annotation distributions are inspected across tasks, modalities, and score ranges to avoid excessive concentration in high-quality or low-quality examples.

The expert reviewers also verify that each multimodal task genuinely requires the designated input modalities. Samples that can be solved reliably without using one of the claimed modalities are revised or removed from the corresponding multimodal category. For generation and editing tasks, preservation constraints are checked separately from requested modifications to prevent an output from receiving a high score merely because the requested change is present.

Chronological Evaluation and Leakage Prevention. The 2,709-sample collection is not divided into training, validation, and test subsets. Instead, user-specific adaptation and evaluation follow interaction order. A current output is evaluated with the user state available before its feedback update. Later corrections, feedback, and preference changes cannot enter its memory, acceptance score, or calibration state. Related media identifiers support duplicate checks and source attribution; they do not define a held-out partition.

Privacy and Data Handling. User-related information is anonymized before inclusion in the benchmark. Personally identifying information that is unnecessary for task completion is removed or replaced. Personalized context contains task-relevant preferences and interaction patterns rather than sensitive identity attributes.

Public and authorized data sources are used according to their applicable usage conditions. For source media that cannot be redistributed directly, the benchmark release can provide identifiers, timestamps, annotations, and evaluation metadata rather than the original files. The benchmark is intended for research on multimodal agent evaluation, verification, and personalization.

B.4 Annotation Agreement

The main benchmark scores are contributed by all 34 participants as described in Appendix B.3. The agreement analysis here is a separate expert-only audit on a common set of outputs independently reviewed by the five domain experts. Table 25 reports their agreement using ICC(2,k) for mean continuous ratings and Krippendorff's α for ordinal ratings. Table 26 compares the LLM feedback quantizer with expert satisfaction consensus on the same audited records, using correlations, MAE, weighted κ, and categorical agreement. Because the five experts are also among the 34 study participants, participant and expert-review roles overlap.

Table 25: Inter-annotator agreement across the four benchmark metrics.

Metric	**ICC(2,k)**	**Krippendorff α**
GFS	0.86	0.80
EIC	0.89	0.84
MFC	0.82	0.76
PAS	0.79	0.72
Overall	0.84	0.78

Table 26: Agreement between the LLM feedback quantizer and expert consensus.

Statistic	**Value**
Pearson r	0.80
Spearman ρ	0.82
MAE on the 0–10 scale	0.63
Quadratic-weighted κ	0.76
Exact agreement	46.8%
Agreement within ± 1 point	88.6%

The experts agree most on EIC (ICC 0.89; $\alpha = 0.84$) and least on PAS (0.79; 0.72), consistent with the greater subjectivity of preference ratings. The quantizer has 0.63 MAE relative to expert consensus; 88.6% of its ratings fall within one point, although exact agreement is 46.8%. These figures describe the dedicated expert audit rather than the production scoring process used for all 92,106 main-study encounters.

Representative Disagreements. Three recurring cases account for most disagreements. First, an explicit high rating accompanied by a request for complete regeneration produces inconsistent evidence; the expert audit assigns a conservative middle score with low confidence. Second, content may be objectively correct while strongly mismatching a user's stable style preference, leading to high GFS/EIC but substantially lower PAS. Third, immediate regeneration without textual feedback is ambiguous because it may indicate dissatisfaction or exploratory behavior; such samples are assigned lower confidence and do not trigger online updating when confidence is below τ_c.

Table 27: Difficulty distribution in the instruction catalog.

Group	Easy	Med.	Hard	Total
Understanding	228	463	226	917
Retrieval/Transformation	229	459	225	913
Generation/Editing	220	443	216	879
Total	677	1,365	667	2,709

Medium-difficulty instructions account for 1,365 of 2,709 cases (50.4%); easy and hard cases each account for about one quarter. The three capability groups have similar proportions, so no single group dominates the reported difficulty distribution.

Cold-Start Breakdown. To isolate the effect of available user evidence, we use a matched subset of 168 evaluation interactions whose users have at least eight valid prior feedback items. For every interaction, the same chronological history is truncated to $k \in \{0, 1, 2, 4, 6, 8\}$ feedback items, so performance changes are not confounded by comparing different users or tasks. This matched-history analysis uses the additional masked-FPV variant defined in Appendix A.6: at $k = 0$, FPV is excluded, and its influence grows after valid feedback arrives. Figure 5 therefore evaluates this supplementary masked-FPV protocol; the main-paper default instead retains the positive ϵ floor.

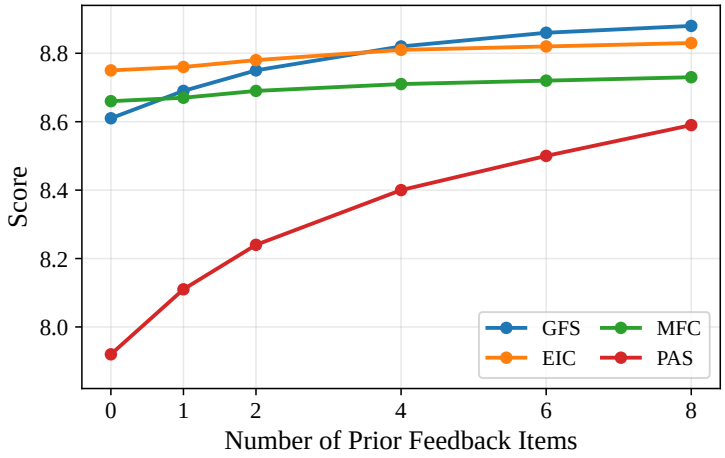


Figure 5: Cold-start personalization of the supplementary masked-FPV variant with increasing amounts of prior user feedback. PAS improves progressively as user-specific evidence accumulates, while GFS, EIC, and MFC remain comparatively stable.

As shown in Figure 5, the largest personalization gains occur within the first four feedback items: PAS increases from 7.92 at cold start to 8.40 at $k = 4$. Subsequent feedback continues to improve PAS, but the curve begins to saturate, reaching 8.59 at $k = 8$. In contrast, GFS, EIC, and MFC vary

by no more than 0.27 points across the full range, indicating that online personalization improves user alignment without destabilizing general task execution or hierarchical verification.

C Additional Experimental Results

Related main text: Section 5.

C.1 Experimental Setup

Related main text: Section 5.1.

This subsection gives the experimental settings for the method in Appendix A. The controlled agent-framework comparison uses the same benchmark, task backbone, tool backends, decoding limits, and maximum revision budget across frameworks; direct-model and backbone-replacement experiments use the models specified in their respective tables. Rows describing the default MUSE configuration report the same task-macro scores as Appendix C.2. Modified configurations report their own experimental results.

Online Evaluation and Common Protocol. The main comparison uses the 2,709-instruction collection without fixed training, validation, or test subsets. Each main-comparison method is executed once for every one of the $34 \times 2{,}709 = 92{,}106$ participant–instruction encounters. At interaction t, the method receives the same-user history prefix available before t. Its new output receives benchmark scores; only afterward does the next actual answer-linked historical feedback record become available to later interactions. The shared BERT encoder and shared modules remain frozen during online use. GFS, EIC, and MFC use all encounter outputs, whereas PAS uses only the personalized encounters. Detailed task scores provide the basis for the two-stage user/instruction aggregation and task macro-average in Appendix C.2. FPV initialization and module-score calibration use the separate 800-example development collection outside the benchmark.

Supplementary Evaluation Panels. The main-text experiments use evaluations contributed by all 34 participants. Appendix entries that simply reconstruct or reuse those main-benchmark scores inherit the same 34-participant evaluation. Newly run appendix-only ablations, sensitivity studies, and robustness analyses use smaller evaluation panels: each scalar-score experiment includes one domain expert together with 3–6 graduate-student or student participants, so every supplementary evaluation has expert oversight. The exact panel size varies with the task and modality requirements. For appendix experiments evaluated through voting rather than scalar ratings, one expert organizes the session and the participating graduate-student and student users cast the votes; the recorded vote result is used for the reported comparison. This protocol separates the full main-benchmark evaluation from the smaller follow-up panels used to analyze specific design choices.

FPV Offline and Online Optimization. Table 28 reports initialization on the benchmark-disjoint 800-example real-user development collection, which also supplies score-map calibration pairs; Table 29 reports the online settings. The online batch has a target maximum size of 16: one current valid sample, up to seven distinct recent samples, and up to eight distinct replay samples. Both historical pools contain only eligible same-user feedback; replay excludes the current and selected recent samples. Sampling is without replacement, and the batch remains smaller when history is insufficient, matching Section 3.5. Low-confidence feedback is retained for audit but excluded from these training pools.

Table 28: FPV initialization on 800 real-user satisfaction examples outside the benchmark.

Configuration	Value
Initialization source	Separate real-user satisfaction examples
Initialization samples / epochs	800 / 100
Encoder backbone / hidden size	BERT-base-uncased / 768
Maximum length per encoded item	512 tokens
Encoder status	Frozen
Memory-attention layers / heads	2 / 8
adapter bottleneck / parameters	64 / 98,304
Trainable modules	Memory attention, prediction head, adapter
Optimizer / learning rate	AdamW / 1×10^{-4}
Batch size / weight decay	16 / 1×10^{-2}
Objective	Confidence-weighted SmoothL1
Label / confidence ranges	$[0, 10]$ / $[0, 1]$

Table 29: Online personalization and decision settings.

Configuration	Value
adapter learning rate / update steps	5×10^{-4} / 3
Target maximum batch size	16
Current / recent / replay composition	1 / up to 7 / up to 8
Replay-buffer capacity per user	64
Temporal decay γ	0.95
adapter regularization λ_{reg}	1×10^{-3}
Feedback confidence threshold τ_c	0.60
FPV volume-saturation count N_0	4 valid feedback items
Reliability window $\lvert\mathcal{W}_t\rvert$	4 interactions
Acceptance threshold τ_s	0.75
FUV aggregation / fusion smoothing ϵ	0.5
Default / extended revision budget	2 / 5 rounds

Verifier and Judge Configuration. The verifier stack is hybrid: prompt-based multimodal models perform task-conditioned semantic inspection, while frozen specialist models supply grounding, alignment, transcription, segmentation, and quality evidence. Specialist outputs are inserted as evidence fields rather than treated as final judgments. The full generic prompts are provided in Appendix A.4.

Concrete backends, implementation types, and principal hyperparameters are listed in Table 30.

Table 30: Concrete verifier, judge, and personalization backends. Prompt-based verification modules use deterministic decoding.

Component	Backend	Type / Setting	Primary Function and Hyperparameters
Text FUV	Qwen3.5-Omni	Prompt-based LLM	Instruction compliance, relevance, coherence, style, and localized defect reporting; temperature 0, maximum 512 output tokens.
Text entailment	DeBERTa-v3-large-MNLI	Frozen classifier	Claim–evidence support and contradiction; maximum length 512, entailment threshold 0.65.
Image FUV	Qwen3.5-Omni	Prompt-based MLLM	Subject, attribute, spatial-relation, prompt-alignment, and edit-preservation checks; temperature 0, maximum 512 output tokens.
Object grounding	Grounding DINO-SwinT	Frozen detector	Object existence and localization; box threshold 0.35 and text threshold 0.25.

Continued on next page

Table 30 (continued)

Component	Backend	Type / Setting	Primary Function and Hyperparameters
Region preservation	SAM-family segmenter	Frozen segmenter	Requested edit mask and preservation of unaffected regions; mask-IoU threshold 0.50.
Text–image alignment	SigLIP-B/16	Frozen encoder	Prompt–image semantic alignment; cosine threshold 0.30.
OCR verification	PaddleOCR	Frozen tool	Text rendered in images and visual instruction constraints; confidence threshold 0.80.
Video FUV	Qwen3.5-Omni	Prompt-based MLLM	Event correctness, temporal order, action, identity, and preservation; 16 uniform frames plus scene-change keyframes.
Temporal evidence	InternVideo2-L	Frozen encoder	Event and clip similarity; eight clips with 16 frames per clip.
Audio FUV	Qwen3.5-Omni	Prompt-based multimodal model	Speech semantics, acoustic events, emotion, style, timing, and preservation; temperature 0, maximum 512 output tokens.
Speech recognition	Whisper-large-v3	Frozen tool	Transcript and timestamp evidence; beam size 5.
Speaker evidence	pyannote.audio 3.1	Frozen tool	Speaker diarization and identity-preservation evidence; segmentation threshold 0.50.
Text–audio alignment	CLAP	Frozen encoder	Text–audio semantic correspondence; 48-kHz audio and cosine threshold 0.28.
Audio quality	DNSMOS	Frozen predictor	Noise, distortion, and intelligibility; quality threshold 3.0.
CCV judge	Qwen3.5-Omni	Prompt-based judge	Rebuttal adjudication, cross-modal conflict detection, hard-violation identification, and prioritized revision; temperature 0, maximum 768 output tokens.
Hard gate (extra variant only)	Deterministic rules	Rule-based	Supplementary gated-fusion variant only: confirmed factual errors, missing primary modality, explicit hard-instruction violations, severe cross-modal conflicts, and safety/privacy failures.
FPV	Frozen BERT + user adapter	Trained regressor	BERT-base-uncased, hidden size 768; two-layer eight-head memory attention; 64-dimensional adapter; 800 external satisfaction examples over 100 initialization epochs.

As shown in Table 30, MUSE uses a hybrid design: LLM prompts provide open-ended diagnosis and adjudication, while frozen specialist models supply localized or calibrated evidence.

Modality-Specific Prompt Specialization. All FUV prompts share the structured template in Appendix A.4, but the target criteria differ by modality. Text verification requests claim-level evidence, explicit-instruction status, unsupported statements, and style deviations. Image verification requests object and attribute grounding, spatial relations, perceptual artifacts, and preservation masks. Video verification additionally requests event timestamps, identity continuity, motion plausibility, and temporal ordering. Audio verification requests transcript spans, speaker evidence, sound-event intervals, prosody, style, and acoustic-quality evidence. Specialist-model results are inserted into the prompt as evidence fields rather than being treated as final decisions.

Structured FUV Output. Each modality-specific verifier returns the following schema:

> `score`: integer in $[0, 10]$; `confidence`: value in $[0, 1]$; `satisfied_instructions`: list; `issues`: list of type, severity, evidence, location, and hard-violation status; and `revision_suggestion`: concise actionable correction.

CCV Adjudication. CCV receives the source evidence, initial FUV reports, task-tool rebuttal, and verifier second reviews. It confirms, revises, or rejects each local claim; detects semantic, factual,

temporal, stylistic, redundancy, and missing-correspondence conflicts; and outputs a cross-modal score, confidence, validated local issues, cross-modal conflicts, hard violations, failure attribution, and prioritized revision targets. As in Section 3.4, CCV jointly considers cross-modal consistency, finding validity, and overall task validity. Its structured adjudication is distinct from the local FUV report and wrapper assessment; the signals need not be statistically independent.

Task Decoding, Calibration, and Runtime. The default task backbone is Qwen3.5-Omni with temperature 0.2 and a maximum of 1,024 output tokens. Prompt-based FUV and CCV modules use temperature 0, with maximum outputs of 512 and 768 tokens, respectively. Each of the four fusion signals is mapped to $[0, 1]$ by a learned monotonic logistic calibrator fitted from agent-produced scores and the corresponding participant ratings on the separate 800-example collection (Appendix C.10). The default verification–revision budget is two rounds; only the dedicated revision-effectiveness analysis allows up to five rounds. The initial task output is produced before these rounds. A revision round is a completed correction cycle that can trigger one further task-tool generation, so the corresponding generation cap is three for the default budget and six for the extended budget. If none of these versions reaches the acceptance criterion within the budget, the final generated version is returned and recorded as below threshold. Verifier calls within a round are counted separately. Experiments run with PyTorch 2.5.1 on eight H100 80GB GPUs. Independent FUV calls are dispatched in parallel when their inputs are available.

Controlled Fairness Across Agent Frameworks. All controlled agent-framework comparisons use the same Qwen3.5-Omni task backbone, the same Core-50 task-tool interfaces and their fixed backends, identical decoding limits, a maximum of two verification–revision iterations, and no hidden retry mechanism. Every compared framework is run on the same 18 tasks, 34 participants, and 92,106 participant–instruction encounters. Each framework follows its own invocation policy, so its actual call count can differ. We report task-model, verifier/judge, and local-tool invocations separately because these calls have different costs. End-to-end latency is measured from request receipt to final output under a fixed cache state and hardware allocation; parallel calls are counted individually but contribute only their realized wall-clock overlap.

C.2 Per-Task Results on the 18 Benchmark Tasks

Related main text: Section 5.3 and Table 1.

We report detailed results for all 18 benchmark tasks. Each table includes GFS, EIC, MFC, and PAS on the shared 0–10 scale, together with their within-task arithmetic mean. The sample count in each caption is the number of distinct instructions; every method contributes 34 encounter outputs per instruction. Each displayed instruction score first averages the 34 users, after which the task value averages instructions; PAS uses only personalized instructions. The main paper then uses an equal-weight macro-average over the same 18 tasks, as specified in Section 4. The best and second-best results in each column are shown in bold and underlined, respectively. To keep the task-level tables within one column, Q3-O, Q3.5-O, and Q3.5-P abbreviate Qwen3-Omni, Qwen3.5-Omni, and Qwen3.5-Plus, respectively. Global Judge is the abbreviated table label for Global Judge + Regeneration.

Aggregation and Rounding. Tables 31–48 report task-level values using the user-first aggregation in Eqs. 43–45. Tables 1 and 2 use equal-weight macro-averages across all 18 tasks. Each metric is averaged before Avg. is formed; the final values are rounded to two decimal places using round-half-up. For example, ControlLLM EIC is 8.3178 before rounding and is reported as 8.32. The within-task Avg. column is not re-averaged to obtain the main results. Rounded task scores cannot recover unrounded encounter-level values. The heatmap and task-level comparisons use the same task aggregates. Caption counts denote unique instructions, not encounter outputs.

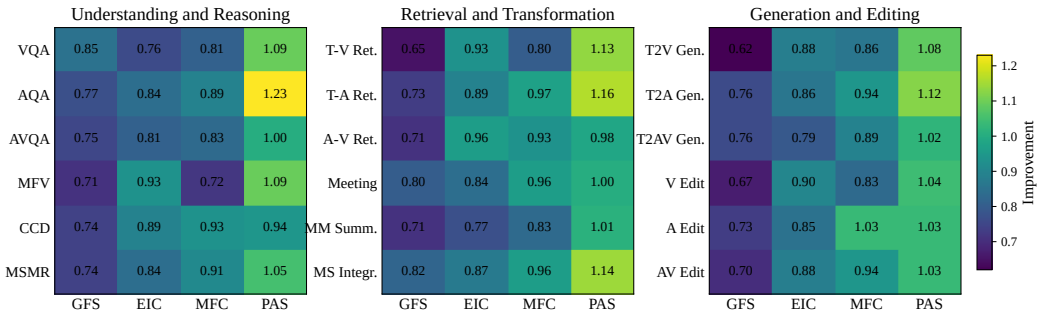


Figure 6: Per-task improvements of MUSE over direct Qwen3.5-Omni inference. Each cell reports the difference in GFS, EIC, MFC, or PAS. The three panels correspond to understanding and reasoning, retrieval and transformation, and generation and editing.

Figure 6 provides a compact overview of the task-level gains reported in the following tables. MUSE improves all four metrics on every task, with a mean gain of 0.89 points across the 72 task–metric combinations. PAS exhibits the largest average improvement (1.06), followed by MFC (0.89), EIC (0.86), and GFS (0.73). This pattern is consistent with the framework design: FPV directly strengthens user alignment, while FUV and CCV primarily improve instruction compliance and multimodal consistency. The retrieval and information-transformation group shows the largest mean gain (0.90), reflecting the benefit of structured evidence preservation and cross-source reconciliation. At the task level, the largest PAS improvement appears in Audio Question Answering (+1.23), the largest MFC improvement in Instruction-Guided Audio Editing (+1.03), and the largest EIC improvement in Audio–Visual Retrieval (+0.96). Positive gains are distributed across all three task groups rather than concentrated in a few categories, indicating that MUSE improves both general multimodal reliability and user-specific alignment across heterogeneous task structures.

Understanding and Reasoning Tasks.

Results for Visual Question Answering are reported in Table 31.

Table 31: Visual Question Answering (137 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.34	8.21	8.17	7.69	8.10
Qwen3.5-Omni	8.23	8.05	7.91	7.49	7.92
Qwen3-Omni	7.81	7.76	7.78	7.24	7.65
Gemini 3.1 Pro	8.12	7.85	8.01	7.45	7.86
Qwen3.5-Plus	7.73	7.56	7.56	7.23	7.52
Chameleon	8.48	8.27	8.24	7.76	8.19
Global Judge	8.32	8.12	8.17	7.68	8.07
HuggingGPT	8.49	8.22	8.23	7.68	8.16
ControlLLM	8.24	8.15	8.15	7.71	8.06
MUSE (GPT-4o)	**9.27**	**9.47**	**9.17**	**8.68**	**9.15**
MUSE (Gemini)	9.03	8.85	8.83	8.45	8.79
MUSE (Q3-O)	8.46	8.29	8.36	8.11	8.31
MUSE (Q3.5-P)	8.29	8.07	8.11	7.99	8.12
MUSE (Q3.5-O)	9.08	8.81	8.72	8.58	8.80

Results for Audio Question Answering are reported in Table 32.

Table 32: Audio Question Answering (172 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.28	8.05	8.02	7.54	7.97
Qwen3.5-Omni	8.20	7.99	7.97	7.47	7.91
Qwen3-Omni	7.83	7.75	7.87	7.30	7.69
Gemini 3.1 Pro	8.10	7.91	8.07	7.29	7.84
Qwen3.5-Plus	7.74	7.47	7.47	7.17	7.46
Chameleon	8.37	8.19	8.20	7.69	8.11
Global Judge	8.25	8.16	8.19	7.70	8.08
HuggingGPT	8.36	8.24	8.19	8.45	8.31
ControlLLM	8.79	8.60	8.61	7.56	8.39
MUSE (GPT-4o)	**9.15**	**9.41**	**9.05**	8.60	**9.05**
MUSE (Gemini)	9.00	8.83	8.78	8.41	8.76
MUSE (Q3-O)	8.41	8.24	8.27	8.13	8.26
MUSE (Q3.5-P)	8.19	8.04	8.02	7.90	8.04
MUSE (Q3.5-O)	8.97	8.83	8.86	**8.70**	8.84

Results for Audio-Visual Question Answering are reported in Table 33.

Table 33: Audio-Visual Question Answering (109 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.25	8.08	8.00	7.61	7.99
Qwen3.5-Omni	8.08	7.95	7.71	7.51	7.81
Qwen3-Omni	7.67	7.55	7.56	7.24	7.51
Gemini 3.1 Pro	7.96	7.83	7.73	7.33	7.71
Qwen3.5-Plus	7.61	7.51	7.35	7.25	7.43
Chameleon	8.30	8.15	8.01	7.71	8.04
Global Judge	8.13	8.08	8.00	7.69	7.98
HuggingGPT	8.32	8.21	8.04	8.44	8.25
ControlLLM	8.67	8.55	8.39	7.60	8.30
MUSE (GPT-4o)	**9.21**	**9.46**	**8.98**	**8.69**	**9.09**
MUSE (Gemini)	8.91	8.74	8.58	8.35	8.65
MUSE (Q3-O)	8.25	8.14	8.15	8.08	8.16
MUSE (Q3.5-P)	8.27	8.12	7.88	7.97	8.06
MUSE (Q3.5-O)	8.83	8.76	8.54	8.51	8.66

Results for Multimodal Fact Verification are reported in Table 34.

Table 34: Multimodal Fact Verification (198 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.21	8.11	7.98	7.65	7.99
Qwen3.5-Omni	8.07	7.96	7.84	7.44	7.83
Qwen3-Omni	7.69	7.57	7.67	7.38	7.58
Gemini 3.1 Pro	7.96	7.88	7.89	7.49	7.81
Qwen3.5-Plus	7.60	7.63	7.40	7.20	7.46
Chameleon	8.43	8.26	8.07	7.66	8.11
Global Judge	8.23	8.18	8.07	7.73	8.05
HuggingGPT	8.39	8.26	7.96	7.66	8.07
ControlLLM	8.18	8.06	7.90	7.64	7.95
MUSE (GPT-4o)	**9.15**	**9.06**	**8.89**	**8.59**	**8.92**
MUSE (Gemini)	8.92	8.76	8.65	8.40	8.68
MUSE (Q3-O)	8.25	8.22	8.17	8.15	8.20
MUSE (Q3.5-P)	8.22	8.12	8.00	7.91	8.06
MUSE (Q3.5-O)	8.78	8.89	8.56	8.53	8.69

Results for Cross-modal Conflict Detection are reported in Table 35.

Table 35: Cross-modal Conflict Detection (86 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.14	8.14	7.86	7.67	7.95
Qwen3.5-Omni	8.07	7.92	7.62	7.57	7.80
Qwen3-Omni	7.70	7.67	7.51	7.24	7.53
Gemini 3.1 Pro	7.87	7.78	7.73	7.37	7.69
Qwen3.5-Plus	7.63	7.44	7.27	7.22	7.39
Chameleon	8.32	8.14	7.99	7.68	8.03
Global Judge	8.21	8.06	7.96	7.66	7.97
HuggingGPT	8.30	8.23	7.98	7.63	8.04
ControlLLM	8.18	8.16	7.97	7.67	8.00
MUSE (GPT-4o)	**9.10**	**9.07**	**8.85**	**8.67**	**8.92**
MUSE (Gemini)	8.84	8.73	8.58	8.50	8.66
MUSE (Q3-O)	8.27	8.15	8.08	8.11	8.15
MUSE (Q3.5-P)	8.14	8.17	7.85	7.96	8.03
MUSE (Q3.5-O)	8.81	8.81	8.55	8.51	8.67

Results for Multi-source Multimodal Reasoning are reported in Table 36.

Table 36: Multi-source Multimodal Reasoning (215 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.15	8.04	7.88	7.46	7.88
Qwen3.5-Omni	7.98	7.94	7.78	7.41	7.78
Qwen3-Omni	7.68	7.52	7.57	7.20	7.49
Gemini 3.1 Pro	7.82	7.83	7.76	7.33	7.69
Qwen3.5-Plus	7.49	7.44	7.33	7.16	7.36
Chameleon	8.26	8.15	7.93	7.73	8.02
Global Judge	8.22	8.07	8.00	7.57	7.97
HuggingGPT	8.34	8.29	8.11	7.71	8.11
ControlLLM	8.18	8.09	8.00	7.64	7.98
MUSE (GPT-4o)	**9.02**	**8.97**	**8.87**	**8.55**	**8.85**
MUSE (Gemini)	8.87	8.89	8.63	8.33	8.68
MUSE (Q3-O)	8.18	8.24	8.12	8.09	8.16
MUSE (Q3.5-P)	8.10	8.08	8.01	7.90	8.02
MUSE (Q3.5-O)	8.72	8.78	8.69	8.46	8.66

Retrieval and Information Transformation Tasks.

Results for Text–Visual Retrieval are reported in Table 37.

Table 37: Text–Visual Retrieval (142 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.33	8.25	8.18	7.55	8.08
Qwen3.5-Omni	8.25	7.96	7.94	7.48	7.91
Qwen3-Omni	7.89	7.60	7.67	7.34	7.63
Gemini 3.1 Pro	8.04	7.97	8.07	7.39	7.87
Qwen3.5-Plus	7.69	7.62	7.47	7.32	7.53
Chameleon	8.40	8.21	8.15	7.77	8.13
Global Judge	8.20	8.16	8.21	7.68	8.06
HuggingGPT	8.55	8.35	8.29	7.84	8.26
ControlLLM	8.31	8.16	8.03	7.66	8.04
MUSE (GPT-4o)	**9.27**	**9.16**	**9.13**	**8.64**	**9.05**
MUSE (Gemini)	8.99	8.92	8.72	8.43	8.77
MUSE (Q3-O)	8.42	8.30	8.25	8.02	8.25
MUSE (Q3.5-P)	8.25	8.12	8.09	7.88	8.09
MUSE (Q3.5-O)	8.90	8.89	8.74	8.61	8.79

Results for Text–Audio Retrieval are reported in Table 38.

Table 38: Text–Audio Retrieval (164 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.29	8.16	8.02	7.55	8.01
Qwen3.5-Omni	8.22	7.94	7.87	7.52	7.89
Qwen3-Omni	7.94	7.55	7.81	7.25	7.64
Gemini 3.1 Pro	8.00	7.92	8.07	7.42	7.85
Qwen3.5-Plus	7.68	7.57	7.49	7.22	7.49
Chameleon	8.35	8.22	8.14	7.71	8.11
Global Judge	8.25	8.17	8.19	7.61	8.06
HuggingGPT	8.49	8.30	8.33	8.48	8.40
ControlLLM	8.78	8.60	8.59	7.63	8.40
MUSE (GPT-4o)	**9.06**	**9.06**	**8.96**	8.64	**8.93**
MUSE (Gemini)	8.93	8.88	8.73	8.44	8.75
MUSE (Q3-O)	8.41	8.31	8.29	8.05	8.27
MUSE (Q3.5-P)	8.22	8.08	7.97	7.97	8.06
MUSE (Q3.5-O)	8.95	8.83	8.84	**8.68**	8.83

Results for Audio–Visual Retrieval are reported in Table 39.

Table 39: Audio–Visual Retrieval (91 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.29	8.22	8.01	7.60	8.03
Qwen3.5-Omni	8.13	7.91	7.85	7.57	7.87
Qwen3-Omni	7.70	7.64	7.59	7.19	7.53
Gemini 3.1 Pro	7.93	7.84	7.86	7.38	7.75
Qwen3.5-Plus	7.56	7.46	7.36	7.21	7.40
Chameleon	8.37	8.09	8.03	7.71	8.05
Global Judge	8.14	8.10	8.05	7.66	7.99
HuggingGPT	8.33	8.26	8.06	8.45	8.28
ControlLLM	8.68	8.59	8.48	7.68	8.36
MUSE (GPT-4o)	**9.15**	**9.12**	**9.00**	**8.62**	**8.97**
MUSE (Gemini)	8.85	8.90	8.80	8.41	8.74
MUSE (Q3-O)	8.31	8.18	8.30	8.15	8.24
MUSE (Q3.5-P)	8.23	8.11	7.98	7.87	8.05
MUSE (Q3.5-O)	8.84	8.87	8.78	8.55	8.76

Results for Meeting Minutes Generation are reported in Table 40.

Table 40: Meeting Minutes Generation (203 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.25	8.08	8.02	7.67	8.01
Qwen3.5-Omni	8.27	8.04	7.89	7.67	7.97
Qwen3-Omni	7.94	7.65	7.82	7.36	7.69
Gemini 3.1 Pro	8.13	7.81	7.99	7.46	7.85
Qwen3.5-Plus	7.66	7.62	7.50	7.20	7.50
Chameleon	8.51	8.25	8.05	7.86	8.17
Global Judge	8.36	8.15	8.24	7.72	8.12
HuggingGPT	8.49	8.23	8.13	7.69	8.14
ControlLLM	8.28	8.11	7.98	8.25	8.16
MUSE (GPT-4o)	**9.26**	**9.04**	**8.98**	8.64	**8.98**
MUSE (Gemini)	9.01	8.86	8.71	8.50	8.77
MUSE (Q3-O)	8.43	8.32	8.31	8.14	8.30
MUSE (Q3.5-P)	8.23	8.05	8.06	8.01	8.09
MUSE (Q3.5-O)	9.07	8.88	8.85	**8.67**	8.87

Results for Multimodal Content Summarization are reported in Table 41.

Table 41: Multimodal Content Summarization (128 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.33	8.16	8.03	7.68	8.05
Qwen3.5-Omni	8.22	8.07	7.87	7.60	7.94
Qwen3-Omni	7.77	7.63	7.73	7.27	7.60
Gemini 3.1 Pro	8.03	7.86	7.94	7.37	7.80
Qwen3.5-Plus	7.73	7.59	7.36	7.31	7.50
Chameleon	8.40	8.20	8.15	7.80	8.14
Global Judge	8.23	8.07	8.24	7.74	8.07
HuggingGPT	8.49	8.23	8.13	7.71	8.14
ControlLLM	8.23	8.14	8.06	7.67	8.03
MUSE (GPT-4o)	**9.10**	**9.04**	**9.06**	**8.68**	**8.97**
MUSE (Gemini)	9.04	8.93	8.69	8.45	8.78
MUSE (Q3-O)	8.36	8.16	8.29	8.17	8.25
MUSE (Q3.5-P)	8.20	8.14	8.06	7.99	8.10
MUSE (Q3.5-O)	8.93	8.84	8.70	8.61	8.77

Results for Multi-source Information Integration are reported in Table 42.

Table 42: Multi-source Information Integration (185 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.21	8.01	7.96	7.54	7.93
Qwen3.5-Omni	8.10	7.93	7.83	7.46	7.83
Qwen3-Omni	7.66	7.64	7.68	7.31	7.57
Gemini 3.1 Pro	7.89	7.86	7.85	7.37	7.74
Qwen3.5-Plus	7.61	7.46	7.41	7.25	7.43
Chameleon	8.40	8.18	8.10	7.75	8.11
Global Judge	8.20	8.10	8.10	7.64	8.01
HuggingGPT	8.46	8.26	8.16	7.82	8.18
ControlLLM	8.26	8.10	8.05	7.68	8.02
MUSE (GPT-4o)	**9.09**	**9.02**	**8.96**	8.59	**8.92**
MUSE (Gemini)	8.88	8.87	8.77	8.47	8.75
MUSE (Q3-O)	8.35	8.19	8.21	8.07	8.21
MUSE (Q3.5-P)	8.30	8.11	8.01	7.96	8.10
MUSE (Q3.5-O)	8.92	8.80	8.79	**8.60**	8.78

Generation and Editing Tasks.

Results for Text-to-Visual Generation are reported in Table 43.

Table 43: Text-to-Visual Generation (156 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.28	8.09	8.08	7.71	8.04
Qwen3.5-Omni	8.22	7.90	7.88	7.53	7.88
Qwen3-Omni	7.78	7.60	7.78	7.35	7.63
Gemini 3.1 Pro	7.99	7.80	7.84	7.40	7.76
Qwen3.5-Plus	7.61	7.49	7.55	7.29	7.49
Chameleon	8.39	8.17	8.24	7.72	8.13
Global Judge	8.24	8.04	8.16	7.76	8.05
HuggingGPT	8.27	8.13	8.04	7.56	8.00
ControlLLM	8.26	8.10	7.96	8.22	8.14
MUSE (GPT-4o)	**9.20**	**9.04**	**9.04**	**8.71**	**9.00**
MUSE (Gemini)	8.96	8.83	8.74	8.56	8.77
MUSE (Q3-O)	8.38	8.17	8.33	8.15	8.26
MUSE (Q3.5-P)	8.22	8.14	8.07	8.06	8.12
MUSE (Q3.5-O)	8.84	8.78	8.74	8.61	8.74

Results for Text-to-Audio Generation are reported in Table 44.

Table 44: Text-to-Audio Generation (119 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.20	7.98	8.00	7.56	7.94
Qwen3.5-Omni	8.13	7.96	7.86	7.55	7.88
Qwen3-Omni	7.82	7.66	7.77	7.24	7.62
Gemini 3.1 Pro	8.05	7.82	8.00	7.52	7.85
Qwen3.5-Plus	7.55	7.45	7.45	7.21	7.42
Chameleon	8.40	8.13	8.03	7.80	8.09
Global Judge	8.19	8.16	8.16	7.71	8.06
HuggingGPT	8.17	8.14	8.05	8.49	8.21
ControlLLM	8.71	8.55	8.55	8.20	8.50
MUSE (GPT-4o)	**9.22**	**9.02**	**8.92**	8.61	**8.94**
MUSE (Gemini)	8.85	8.89	8.78	8.44	8.74
MUSE (Q3-O)	8.30	8.18	8.32	8.18	8.25
MUSE (Q3.5-P)	8.14	8.05	8.01	7.98	8.05
MUSE (Q3.5-O)	8.89	8.82	8.80	**8.67**	8.80

Results for Text-to-Audio-Visual Generation are reported in Table 45.

Table 45: Text-to-Audio-Visual Generation (178 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.26	8.01	7.90	7.58	7.94
Qwen3.5-Omni	8.03	8.00	7.75	7.52	7.83
Qwen3-Omni	7.67	7.57	7.51	7.29	7.51
Gemini 3.1 Pro	7.80	7.70	7.77	7.37	7.66
Qwen3.5-Plus	7.44	7.47	7.28	7.23	7.36
Chameleon	8.28	8.14	7.97	7.78	8.04
Global Judge	8.14	8.04	7.87	7.68	7.93
HuggingGPT	8.14	8.06	7.89	8.45	8.14
ControlLLM	8.60	8.52	8.38	8.16	8.42
MUSE (GPT-4o)	**9.14**	**9.04**	**9.00**	**8.69**	**8.97**
MUSE (Gemini)	8.83	8.79	8.63	8.54	8.70
MUSE (Q3-O)	8.34	8.16	8.18	8.05	8.18
MUSE (Q3.5-P)	8.22	8.13	8.05	7.95	8.09
MUSE (Q3.5-O)	8.79	8.79	8.64	8.54	8.69

Results for Instruction-Guided Visual Editing are reported in Table 46.

Table 46: Instruction-Guided Visual Editing (147 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.25	8.18	8.08	7.71	8.06
Qwen3.5-Omni	8.19	7.97	7.90	7.61	7.92
Qwen3-Omni	7.77	7.62	7.73	7.31	7.61
Gemini 3.1 Pro	7.86	7.76	7.84	7.36	7.71
Qwen3.5-Plus	7.59	7.53	7.46	7.30	7.47
Chameleon	8.41	8.19	8.12	7.81	8.13
Global Judge	8.22	8.00	8.17	7.70	8.02
HuggingGPT	8.32	8.13	8.09	7.58	8.03
ControlLLM	8.28	8.08	8.03	8.23	8.16
MUSE (GPT-4o)	**9.19**	**9.05**	**9.04**	**8.75**	**9.01**
MUSE (Gemini)	8.98	8.82	8.72	8.56	8.77
MUSE (Q3-O)	8.31	8.17	8.22	8.09	8.20
MUSE (Q3.5-P)	8.23	8.07	8.02	8.00	8.08
MUSE (Q3.5-O)	8.86	8.87	8.73	8.65	8.78

Results for Instruction-Guided Audio Editing are reported in Table 47.

Table 47: Instruction-Guided Audio Editing (194 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.22	8.06	7.99	7.63	7.98
Qwen3.5-Omni	8.12	7.97	7.77	7.61	7.87
Qwen3-Omni	7.73	7.63	7.76	7.34	7.62
Gemini 3.1 Pro	8.02	7.89	7.92	7.44	7.82
Qwen3.5-Plus	7.68	7.57	7.47	7.21	7.48
Chameleon	8.30	8.10	8.17	7.67	8.06
Global Judge	8.20	8.04	8.11	7.70	8.01
HuggingGPT	8.26	7.95	7.92	8.51	8.16
ControlLLM	8.71	8.59	8.52	8.22	8.51
MUSE (GPT-4o)	**9.04**	**8.95**	**8.85**	8.61	**8.86**
MUSE (Gemini)	8.93	8.80	8.61	8.56	8.73
MUSE (Q3-O)	8.34	8.22	8.26	8.14	8.24
MUSE (Q3.5-P)	8.03	8.03	8.06	7.99	8.03
MUSE (Q3.5-O)	8.85	8.82	8.80	**8.64**	8.78

Results for Instruction-Guided Audio-Visual Editing are reported in Table 48.

Table 48: Instruction-Guided Audio-Visual Editing (85 samples).

Method	GFS	EIC	MFC	PAS	Avg.
GPT-4o	8.21	8.06	7.88	7.61	7.94
Qwen3.5-Omni	8.04	7.94	7.75	7.45	7.80
Qwen3-Omni	7.66	7.65	7.58	7.31	7.55
Gemini 3.1 Pro	7.79	7.78	7.78	7.34	7.67
Qwen3.5-Plus	7.58	7.58	7.27	7.14	7.39
Chameleon	8.21	8.10	7.96	7.68	7.99
Global Judge	8.09	8.04	8.04	7.58	7.94
HuggingGPT	8.15	8.11	7.96	8.42	8.16
ControlLLM	8.61	8.57	8.40	8.13	8.43
MUSE (GPT-4o)	**9.11**	**8.99**	**8.95**	**8.67**	**8.93**
MUSE (Gemini)	8.84	8.79	8.67	8.44	8.69
MUSE (Q3-O)	8.33	8.20	8.21	8.11	8.21
MUSE (Q3.5-P)	8.12	8.05	7.97	8.02	8.04
MUSE (Q3.5-O)	8.74	8.82	8.69	8.48	8.68

Across Tables 31–36, MUSE variants consistently occupy the strongest positions. The advantage is especially clear on audio-visual reasoning, fact verification, and conflict detection, where isolated unimodal inspection is insufficient.

Across Tables 37–42, the largest gains appear on meeting minutes, summarization, and multi-source integration, indicating that structured verification helps preserve and reconcile evidence during information transformation.

Across Tables 43–48, MUSE remains strongest on generation and editing. Improvements in EIC and MFC are most pronounced when multiple output modalities or preservation constraints must be coordinated.

C.3 Robustness to Sparse and Noisy User Feedback

Related main text: Section 5.6.

Real users may provide feedback intermittently, and automatic feedback quantization may introduce label errors. We therefore perturb only the supervision available to FPV while keeping the backbone, FUV, CCV, tool pool, and chronological interaction order fixed. Feedback retention is varied from 100% to 10%. For label noise, a selected label $r_{u,t} \in [0, 10]$ is perturbed by $\delta \in \{-2, -1, 1, 2\}$ and clipped to the valid range.

Feedback Sparsity. Frozen FPV uses the warm-started predictor without online updates. Naive Online Adaptation updates the user adapter using every observed label without confidence filtering, replay, temporal weighting, or regularization. MUSE uses the complete update strategy.

The PAS trends under feedback sparsity and label noise are summarized in Figure 7. We plot PAS because the perturbations directly affect the supervision used by FPV, whereas GFS, EIC, and MFC remain comparatively stable.

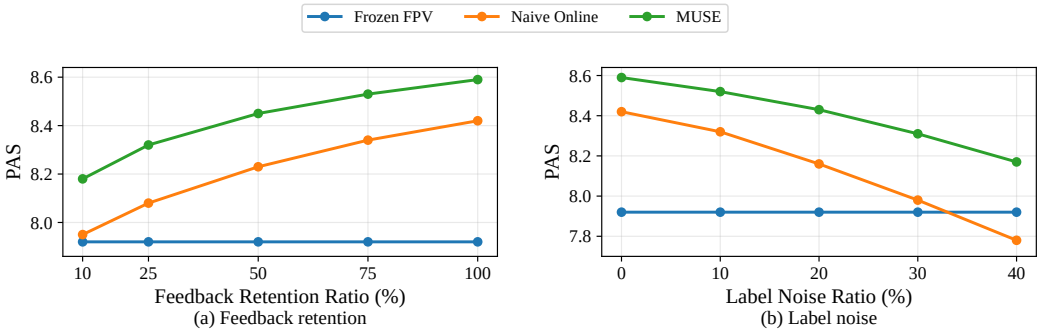


Figure 7: Robustness to incomplete and noisy user feedback. Left: PAS as the proportion of retained feedback decreases. Right: PAS as the proportion of corrupted satisfaction labels increases. MUSE remains more robust than naive online adaptation in both settings.

As shown in Figure 7(a), reducing feedback retention from 100% to 10% lowers MUSE PAS from 8.59 to 8.18, but MUSE remains above both comparison strategies. Naive online adaptation degrades more rapidly and approaches the frozen predictor when only 10% of feedback is retained, showing that isolated updates become unreliable when supervision is sparse. Figure 7(b) shows a similar pattern under corrupted labels. At 40% noise, naive adaptation falls below the frozen predictor (7.78 versus 7.92), whereas MUSE retains a PAS of 8.17. The slower degradation supports the combined use of confidence filtering, same-user replay, temporal weighting, and adapter regularization.

Prediction Robustness. MAE is computed on the same 0–10 satisfaction scale as PAS and FPV labels.

Table 49 directly compares satisfaction-prediction error and rank agreement under the most challenging sparse and noisy settings.

Table 49: FPV prediction robustness under sparse and noisy feedback.

Setting	Method	MAE	Spearman
25% feedback	Naive Online	0.73	0.68
	MUSE	**0.52**	**0.79**
10% feedback	Naive Online	0.88	0.57
	MUSE	**0.64**	**0.71**
20% noise	Naive Online	0.79	0.62
	MUSE	**0.55**	**0.77**
40% noise	Naive Online	1.09	0.44
	MUSE	**0.72**	**0.66**

As shown in Table 49, MUSE maintains lower MAE and higher rank correlation than naive online updating in every tested condition, confirming that the downstream PAS trend reflects more stable satisfaction prediction.

C.4 Memory Compression and Retrieval Strategy Analysis

Related main text: Sections 3.2 and 5.6.

We compare memory representations while keeping the backbone, verification modules, FPV architecture, feedback labels, and online-update budget fixed. No Memory removes historical context; Recent Interaction retains only the latest interaction; Full Memory directly encodes the complete available history; Retrieved Full Memory retrieves task-relevant records; Compact Memory uses the compressed profile; and Compact Memory + Retrieval is the complete dual-memory design.

The overall comparison of memory representations is presented in Table 50.

Table 50: Performance comparison of user-memory strategies.

Memory Strategy	GFS	EIC	MFC	PAS	Avg.
No Memory	8.63	8.73	8.65	7.94	8.49
Recent Interaction	8.72	8.75	8.68	8.21	8.59
Full Memory	8.78	8.78	8.68	8.34	8.65
Retrieved Full Memory	8.83	8.80	8.71	8.47	8.70
Compact Memory	8.84	8.81	8.71	8.50	8.72
Compact Memory + Retrieval	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

As shown in Table 50, directly encoding the complete history improves PAS over using only the latest interaction, but remains 0.25 below the complete strategy. Compact memory suppresses irrelevant and outdated details, while retrieval restores task-specific failures and successful corrections. Their combination yields the strongest PAS without weakening general reliability.

Compact-Memory Capacity. The capacity sensitivity results are reported in Table 51.

Table 51: Effect of the compact-memory token budget.

Tokens	GFS	EIC	MFC	PAS	Avg.
128	8.77	8.77	8.68	8.28	8.63
256	8.81	8.79	8.70	8.40	8.68
512	8.86	8.82	**8.72**	8.54	8.74
1,024	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**
2,048	8.86	8.81	8.71	8.53	8.73

The memory budget counts tokens across all serialized preference items; it is not the length of one encoder input. Each item obeys the 512-token encoder limit specified in Table 28. As shown in Table 51, performance improves until 1,024 tokens and then slightly declines. Smaller memories omit representative evidence, whereas the 2,048-token setting reintroduces redundant and occasionally conflicting historical details.

Retrieval-Source Ablation. Table 52 reports the contribution of each retrieved evidence source.

Table 52: Ablation of evidence sources retrieved from full interaction memory.

Retrieved Evidence	GFS	EIC	MFC	PAS	Avg.
Preferences only	8.70	8.73	8.66	8.37	8.62
Previous failures only	8.79	8.80	8.70	8.26	8.64
Successful corrections only	8.82	8.81	8.71	8.38	8.68
Recent interactions only	8.76	8.77	8.68	8.32	8.63
All relevant evidence	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

As shown in Table 52, preferences primarily improve PAS, while previous failures and successful corrections contribute more to GFS, EIC, and MFC. Combining all evidence types provides complementary information and yields the strongest overall result in this ablation. Task failures and corrections enter the task-memory view, whereas historical subjective preferences enter compact FPV memory and preference-aware revision; they are not added to FUV or CCV inputs.

C.5 User-adapter Isolation and Cross-User Interference

Related main text: Sections 3.5 and 5.6.

MUSE shares the frozen encoder, memory-attention module, and prediction head across users, while maintaining an independent lightweight adapter for each user. We compare a frozen Global Predictor, one Shared adapter for all users, Group adapters assigned by preference similarity, and Per-User adapters.

The four adapter-sharing strategies are compared in Table 53.

Table 53: Comparison of user-adapter sharing strategies.

Adaptation Strategy	**GFS**	**EIC**	**MFC**	**PAS**	**Avg.**
Global Predictor	8.61	8.75	8.66	7.92	8.49
Shared adapter	8.76	8.78	8.68	8.31	8.63
Group adapters	8.82	8.81	8.71	8.45	8.70
Per-User adapters	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

As shown in Table 53, per-user isolation provides the largest improvement in PAS because conflicting preferences do not update the same parameters. The smaller changes in GFS, EIC, and MFC indicate that the adapter mainly calibrates user-specific satisfaction rather than replacing hierarchical verification.

Cross-User Interference. After adapting to target user u, we update the system using feedback from another user v and re-evaluate u. We report absolute changes in the target user's PAS and four-metric Avg.; lower values indicate greater score stability.

The measured cross-user degradation for each sharing strategy is reported in Table 54.

Table 54: Cross-user interference under adapter-sharing strategies.

Strategy	$\lvert\Delta\rvert$**PAS**	$\lvert\Delta\rvert$**Avg.**
Shared adapter	0.31	0.18
Group adapters	0.14	0.08
Per-User adapters	**0.02**	**0.01**

Table 54 shows smaller reported score changes for per-user adapters than for shared updates. Under frozen shared modules and isolated user state, another user's update does not alter the target user's deterministic FPV parameters. The residual 0.02/0.01 score changes therefore cannot by themselves establish parameter leakage; fixed-input, fixed-seed records are needed to separate scoring variation from any implementation-level state sharing.

Conflicting Preferences. Preference Discrimination Accuracy (PDA) is the percentage of paired cases in which the predictor assigns a higher satisfaction score to the output aligned with the evaluated user.

Preference discrimination under deliberately conflicting user profiles is reported in Table 55.

Table 55: Preference discrimination accuracy (%) for conflicting preferences.

Strategy	Length	Detail	Structure	Visual	Audio	Average
Global Predictor	61.8	60.9	62.4	58.7	59.4	60.6
Shared adapter	71.3	69.8	72.1	68.5	67.9	69.9
Group adapters	79.4	78.1	80.0	76.8	75.9	78.0
Per-User adapters	**87.6**	**86.9**	**88.2**	**84.7**	**83.8**	**86.2**

As shown in Table 55, parameter isolation also improves direct discrimination between opposing preferences rather than merely increasing the average satisfaction score.

C.6 Verifier Selection and Coverage Analysis

Related main text: Sections 3.1 and 5.5.

Unlike the component ablation in the main paper, this experiment keeps FUV, CCV, and FPV active and varies only the internal FUV selection policy. A Single Global Verifier performs holistic assessment; Random Selection samples a fixed verifier set; Modality-Matched Selection activates all verifiers associated with the involved modalities; All Verifiers invokes the complete FUV pool; and Task-Conditioned Selection uses the task profile, explicit instructions, historical failures, and predicted risks.

The overall effect of the verifier-selection policy is reported in Table 56.

Table 56: Performance under different verifier-selection strategies.

Selection Strategy	GFS	EIC	MFC	PAS	Avg.
Single Global Verifier	8.22	8.10	8.11	7.68	8.03
Random Selection	8.47	8.40	8.33	8.17	8.34
Modality-Matched Selection	8.70	8.68	8.58	8.39	8.59
All Verifiers	8.83	8.79	8.67	8.52	8.70
Task-Conditioned Selection	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

As shown in Table 56, task-conditioned selection exceeds modality-only selection by 0.16 points in the displayed average scores. Exhaustive verification is also weaker because irrelevant verifiers introduce redundant or unsupported findings.

Verifier Budget. When more verifiers are applicable than the specified budget, the scheduler retains those with the highest task relevance and estimated risk coverage.

The sensitivity to the maximum number of selected FUV verifiers is shown in Table 57.

Table 57: Effect of the maximum number of selected FUV verifiers.

$K_{\max}$	GFS	EIC	MFC	PAS	Avg.
1	8.51	8.45	8.38	8.25	8.40
2	8.70	8.68	8.57	8.42	8.59
3	8.82	8.79	8.68	8.53	8.71
4	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**
5	8.87	8.82	**8.72**	8.58	**8.75**
All	8.83	8.79	8.67	8.52	8.70

As shown in Table 57, the four- and five-verifier settings share the highest displayed average of 8.75, with four verifiers using the smaller budget. A smaller budget leaves important risks unchecked, whereas exhaustive invocation adds low-relevance reports without increasing effective coverage.

Task-Group Results. For Task-Conditioned Selection, each group value in Table 58 averages the six task scores in that group for each metric and then averages the four metrics. Overall averages all 18 tasks and agrees with Table 1 at 8.75. Task-Conditioned Selection uses the default MUSE task results in Appendix C.2; other selection strategies use their respective measured results.

Table 58: Task-group performance of verifier-selection strategies.

Selection Strategy	Understanding	Retrieval	Generation	Overall
Single Global Verifier	8.05	8.08	7.96	8.03
Random Selection	8.36	8.39	8.27	8.34
Modality-Matched Selection	8.61	8.62	8.54	8.59
All Verifiers	**8.73**	8.72	8.66	8.70
Task-Conditioned Selection	8.72	**8.80**	**8.74**	**8.75**

Task-Conditioned Selection has the highest displayed retrieval and generation scores. Its understanding score is 0.01 below All Verifiers, a difference too small at this precision to support a ranking. In generation and editing, the scheduler distinguishes prompt realization, perceptual quality, controllability, and preservation requirements within the same modality.

Historical-Risk-Aware Selection. The benefit of incorporating retrieved historical failures is reported in Table 59.

Table 59: Effect of historical failure retrieval on verifier selection.

Setting	GFS	EIC	MFC	PAS	Avg.
Without failure retrieval	8.84	8.80	8.70	8.48	8.71
With failure retrieval	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

As shown in Table 59, the largest reported difference after historical failure retrieval is in PAS. Verifier selection uses objective task-failure and correction records; historical subjective preferences remain in FPV memory. The PAS difference alone does not establish that user-preference signals entered verification.

C.7 Supplementary Fusion Variants and Experimental Results

Related main text: Section 3.6.

This subsection specifies supplementary variants of score calibration, reliability weighting, cold-start masking, and hard-constraint gating under the stated Qwen3.5-Omni configuration and two-round revision budget. The main-paper default uses positive-ϵ fusion and a score threshold, with neither explicit cold-start masking nor a hard-constraint gate. Its four score maps use agent-produced scores paired with actual user-satisfaction ratings from the separate 800-example collection, as described in Appendix C.10.

Scope of the Experiments. Tables 60–63 compare the supplementary fusion configurations described below. The main-paper evaluation uses the default positive-ϵ fusion rule.

Compared Fusion Variants. We compare four configurations. **Equal-Weight Fusion** assigns a fixed weight of 0.25 to each signal. **Uncalibrated Reliability Fusion** uses module reliability but directly divides the original 0–10 scores by 10. **Calibrated Fusion without Hard Gate** adds learned calibration and FPV cold-start masking but relies only on the fused threshold. **Masked + Gated Fusion** additionally applies the hard-constraint gate defined in Appendix A.6.

Downstream quality under the four fusion variants is reported in Table 60.

Table 60: Downstream performance of supplementary fusion variants.

Fusion Variant	GFS	EIC	MFC	PAS	Avg.
Equal-Weight Fusion	8.64	8.58	8.48	8.35	8.51
Uncalibrated Reliability Fusion	8.73	8.68	8.57	8.43	8.60
Calibrated Fusion w/o Hard Gate	8.83	8.79	8.67	8.54	8.71
Masked + Gated Fusion	**8.88**	**8.83**	**8.73**	**8.59**	**8.76**

In Table 60, the masked and gated variant has the highest average among the compared configurations. Calibration makes the module scores comparable, while the hard gate prevents detected critical violations from being overridden by high scores on other dimensions.

All four metrics increase across the reported configurations. Relative to calibrated fusion without a hard gate, the masked and gated variant improves GFS, EIC, MFC, and PAS by 0.05, 0.04, 0.06, and 0.05 points, respectively. MFC has the largest increase; these comparisons describe the evaluated configurations without establishing a separate causal effect for every component.

Decision Quality and Calibration. Decision Accuracy is the percentage of evaluated outputs for which the automatic accept-or-revise decision matches the panel reference decision. In the voting-based supplementary decision study, one expert organizes the review session and the participating graduate-student and student users cast the accept/revise votes; the recorded vote result provides the reference decision. False Acceptance Rate (FAR) is the percentage of outputs with an annotated unresolved failure that are incorrectly accepted. Expected Calibration Error (ECE) is computed on these benchmark decisions, not on the 800-example development collection; lower values are better. Avg. Rounds reports the average number of revision rounds under the default two-round budget.

Decision accuracy, false acceptance, calibration, and revision efficiency are compared in Table 61.

Table 61: Decision quality and calibration of supplementary fusion variants.

Fusion Variant	Acc.	FAR	ECE	Rounds
Equal-Weight Fusion	84.6	8.1	0.086	1.78
Uncalibrated Reliability Fusion	87.5	6.3	0.064	1.64
Calibrated Fusion w/o Hard Gate	90.7	4.4	0.037	1.49
Masked + Gated Fusion	**93.5**	**1.8**	**0.028**	**1.41**

In Table 61, the masked and gated variant has the highest decision accuracy and the lowest FAR, ECE, and average number of rounds among the compared configurations.

Relative to calibrated fusion without the hard gate, the masked and gated configuration increases decision accuracy by 2.8 percentage points (90.7% to 93.5%), reduces FAR by 2.6 points (4.4% to 1.8%), and reduces ECE from 0.037 to 0.028. Its mean revision count is also lower, from 1.49 to 1.41. These are observed end-to-end differences between configurations; the lower round count is not a guarantee that a veto alone reduces revisions on an identical fixed sequence of candidates.

Cold-Start Masking. Cold start denotes a user state with no eligible prior feedback. The epsilon-only baseline retains a positive FPV weight through smoothing in both numerator and denominator. The activation-mask variant sets the FPV weight to zero and uses the unsmoothed weights defined in Appendix A.6. Thus, this comparison changes both FPV activation and the weighting of the remaining signals.

The cold-start masking results are reported in Table 62.

Table 62: Comparison of optional FPV masking during cold start.

Cold-Start Strategy	Acc.	FAR	ECE
Epsilon-only FPV weight	88.9	5.7	0.058
Explicit FPV activation mask	**91.6**	**3.2**	**0.036**

Table 62 compares the positive-ϵ default with optional explicit masking at the first user interaction. The masked variant has higher decision accuracy and lower FAR and ECE in this comparison. Masking removes FPV from the first decision; it remains a supplementary variant rather than a component of the default rule.

The explicit mask removes the generic FPV estimate from the cold-start decision. Its reported difference from epsilon-smoothed fusion cannot be attributed to masking alone because the other weights also change. Once valid feedback becomes available, FPV contributes according to its maturity rather than a manually specified interaction threshold.

Hard-Violation Analysis. Table 63 reports the false acceptance rate for different critical error types. The largest FAR reductions occur for missing primary modalities and explicit instruction violations. These results are consistent with the purpose of the additional hard gate: preventing detected critical failures from being obscured by strong scores on unaffected dimensions.

Table 63: False acceptance rates (%) with and without the additional hard gate.

Hard Violation	No Gate	Hard Gate
Explicit instruction violation	6.8	**1.5**
Confirmed factual error	5.4	**1.3**
Missing primary modality	7.1	**0.8**
Severe cross-modal conflict	4.8	**1.4**
Safety or privacy violation	2.6	**0.0**
Average	5.3	**1.0**

Together, these experiments compare configurations using calibration, reliability weighting, optional cold-start masking, and optional hard gating. The default method deliberately retains the soft positive-ϵ fusion because it applies the same score-based rule across all 18 tasks without introducing a task-specific critical-error taxonomy or deterministic veto. The masked/gated configuration is therefore treated as an optional deployment policy rather than selected as the benchmark default; its higher displayed numbers describe that additional policy, not the definition of MUSE used in Tables 1–3.

C.8 Computational Fairness and Latency

Related main text: Sections 5.1 and 5.7.

The controlled agent-framework comparison holds the Qwen3.5-Omni task backbone, registered task-tool backends, decoding limits, maximum of two verification–revision iterations, and retry policy constant. Each framework retains its own invocation policy; Table 64 records the resulting calls and latency. FUV calls executed in parallel are counted separately even though their wall-clock latency overlaps. Latency is measured end-to-end from request receipt to final output over five runs, with cache state and hardware held fixed.

Average model and tool calls, per-encounter wall-clock latency, and a full-evaluation serial-time estimate are summarized in Table 64. The last quantity multiplies average latency by all $34 \times 2{,}709 = 92{,}106$ participant–instruction encounters for one method. It is an execution-equivalent serial estimate rather than measured wall-clock project duration, because requests and verifier calls can be parallelized.

Table 64: Average calls and wall-clock latency under the native execution policies. Serial-time estimates cover one full 92,106-encounter evaluation for each method.

Method	Task Calls	Verif./Judge	Local Tools	Total Calls	Latency/Enc.	Serial Time
Direct Qwen3.5-Omni	1.00	0.00	0.00	1.00	6.2 s	158.6 h
Global Judge + Regen.	1.44	1.44	0.00	2.88	13.9 s	355.6 h
Chameleon	1.08	0.00	3.64	4.72	16.4 s	419.6 h
HuggingGPT	1.11	0.00	4.15	5.26	18.7 s	478.4 h
ControlLLM	1.13	0.00	4.58	5.71	20.3 s	519.4 h
MUSE	1.38	5.50	3.33	10.21	24.3 s	621.7 h

Table 64 confirms that MUSE incurs additional verification calls and latency relative to direct inference. Across one complete 92,106-encounter evaluation, the corresponding serial-time estimate is 621.7 hours for MUSE versus 158.6 hours for direct Qwen3.5-Omni; actual elapsed time can be lower under parallel execution.

The task-level MUSE call and latency breakdown is reported in Table 65.

Table 65: Average MUSE calls and latency for each benchmark task. Parallel verifier calls are counted individually.

Task	Calls	Latency	Task	Calls	Latency
Visual QA	8.4	19.2 s	Text–Visual Retrieval	7.6	17.3 s
Audio QA	8.8	20.5 s	Text–Audio Retrieval	7.9	18.1 s
Audio-Visual QA	10.6	25.8 s	Audio–Visual Retrieval	8.8	20.2 s
Fact Verification	10.9	26.4 s	Meeting Minutes	10.6	25.1 s
Conflict Detection	11.2	27.1 s	Multimodal Summarization	9.8	23.4 s
Multi-source Reasoning	11.5	28.0 s	Information Integration	11.0	26.7 s
Text-to-Visual	10.2	24.0 s	Text-to-Audio	9.6	22.8 s
Text-to-Audio-Visual	12.4	30.2 s	Visual Editing	10.8	25.6 s
Audio Editing	10.3	24.7 s	Audio-Visual Editing	13.1	32.0 s

As shown in Table 65, the most expensive tasks are joint audio-visual generation and editing, while single-modality retrieval and question answering require fewer calls.

C.9 Verifier, Judge, and FPV Backbone Sensitivity

Related main text: Section 5.4.

We keep the task-generation backbone fixed as Qwen3.5-Omni and vary only the FUV and CCV backends. The maximum overall spread is moderate, while MFC is more sensitive than PAS because verifier strength mainly affects evidence localization and cross-modal adjudication.

Sensitivity to the FUV and CCV judge backbones is reported in Table 66.

Table 66: Sensitivity to FUV and CCV backbones.

FUV Backbone	CCV Backbone	GFS	EIC	MFC	PAS	Avg.
Qwen3-Omni	Qwen3-Omni	8.70	8.65	8.50	8.51	8.59
Gemini 3.1 Pro	Gemini 3.1 Pro	8.80	8.75	8.61	8.55	8.68
Qwen3.5-Omni	Qwen3.5-Omni	8.88	8.83	8.72	8.59	8.75
GPT-4o	GPT-4o	8.93	8.89	8.82	8.65	8.82
Qwen3.5-Omni	GPT-4o	8.91	8.87	8.80	8.63	8.80
GPT-4o	Qwen3.5-Omni	8.90	8.86	8.78	8.62	8.79

Table 66 indicates moderate backbone sensitivity rather than dependence on a single judge, with stronger judges primarily improving MFC.

The contribution of specialist non-LLM verification tools is isolated in Table 67.

Table 67: Sensitivity to prompt-only and hybrid FUV implementations.

FUV Setting	GFS	EIC	MFC	PAS	Avg.
Prompt-only FUV	8.79	8.72	8.61	8.58	8.68
Hybrid FUV	**8.88**	**8.83**	**8.72**	**8.59**	**8.75**

As shown in Table 67, the hybrid FUV design outperforms prompt-only verification, supporting the use of specialist grounding, alignment, and quality tools.

The effect of replacing the frozen FPV encoder is reported in Table 68.

Table 68: Sensitivity to the frozen FPV text encoder.

Encoder	MAE	Spearman	PAS	Avg.
MiniLM-L6	0.69	0.69	8.48	8.69
RoBERTa-base	0.57	0.76	8.57	8.74
BERT-base-uncased	0.55	0.77	8.59	8.75
DeBERTa-v3-base	**0.53**	**0.79**	**8.61**	**8.77**

Table 68 shows limited variation across the frozen encoders under this setup. This sensitivity analysis alone does not rank encoder choice against the separate contributions of the adapter or memory.

Although DeBERTa-v3-base yields the strongest FPV prediction metrics, its displayed overall gain over BERT-base-uncased is only 0.02 points. We therefore retain BERT-base-uncased as the fixed default and interpret the encoder swap as a robustness check rather than a separate model-selection claim.

C.10 Calibration Procedure and Decision Settings

Related main text: Section 3.6.

Each heterogeneous module score is calibrated independently using a monotonic logistic function

$$\mathrm{Cal}_i(s) = \sigma(a_i s + b_i), \qquad a_i = \mathrm{softplus}(\alpha_i) > 0. \tag{46}$$

The 800 real-user examples used for system development are separate from the 2,709-instruction benchmark and precede all benchmark interactions. For each development example, the agent's task wrapper, FUV, CCV, and FPV produce four raw module scores $\{s_n^{(i)}\}_{i\in\{A,1,2,3\}}$, which are paired with the answer-linked participant rating $r_n \in [0, 10]$ (normalized to $r_n/10$) to fit one monotonic map per signal. The same development collection initializes FPV for 100 epochs; it is not a held-out calibration set and provides no independent estimate of FPV calibration error. The four maps are fixed before evaluation, no benchmark label is used to refit them, and ECE in Appendix C.7 is computed on benchmark decisions. The specified decision settings are $\tau_s = 0.75$, $N_0 = 4$, $\tau_c = 0.60$, $|\mathcal{W}_t| = 4$, and $\epsilon = 0.5$; the same values are used for all 34 users, all 18 tasks, and all 92,106 encounters.

The default decision rule is defined in Section 3.6, and the separately specified masking and hard-gate variants are compared in Appendix C.7. No per-task or per-user threshold retuning is performed during benchmark evaluation.

C.11 Cost–Quality Pareto Analysis

Related main text: Section 5.7.

We compare quality, total invocations, average revision rounds, wall-clock latency, and token consumption under verifier-count and revision-budget ablations. For a request, let T_m sum the actual input and output model tokens from the initial request until the user regards the result as satisfactory and needs no further revision, or until the allowed revision budget is exhausted without an effective

answer. The extended revision analysis stops after at most five unsuccessful rounds; ablations with smaller budgets stop at their own caps and retain their last output. Audio and video are counted using the tokens actually consumed by the respective model. The relative token index is the mean T_m divided by the corresponding mean for direct backbone inference, normalized to $1.0\times$. Revision rounds contribute their actual tokens without a separate round multiplier; model-free tools have no model-token charge, although their invocations and latency are reported separately.

The resulting operating points are visualized in Figure 8.

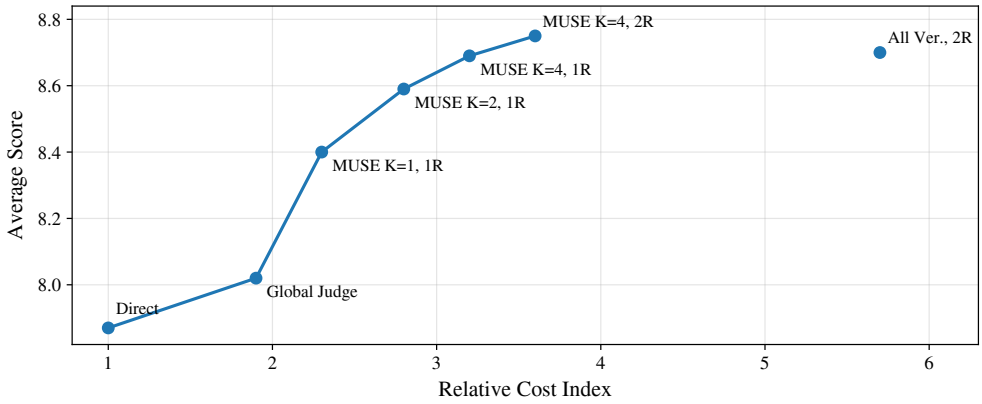


Figure 8: Quality versus relative input/output token use through the response judged sufficiently satisfactory. Direct inference is normalized to $1.0\times$; the connecting line is a visual guide.

As shown in Figure 8, moving from direct inference to one-round MUSE with $K = 2$ increases the average score from 7.87 to 8.59 at a relative cost index of $2.8\times$. Increasing to $K = 4$ produces a smaller positive gain, and a second revision round reaches the highest quality of 8.75 at an index of $3.6\times$. Invoking all verifiers raises the index to $5.7\times$ while reducing the score to 8.70. The nondominated operating points shown comprise direct inference, Global Judge, the $K = 1$, $K = 2$, and $K = 4$ one-round settings, and the complete two-round setting. They span increasing quality and token consumption; the connecting line is a visual guide between evaluated configurations. Calls and latency are shown separately in Appendix C.8.

C.12 Qualitative Case Study

Related main text: Sections 3 and 5.7.

Figure 9 presents a representative meeting-minutes generation example with audio, video, and textual meeting evidence. The user requests concise bullet-point minutes that emphasize budget decisions and action items and has a persistent preference for placing action items first.

The direct output contains three failures. It omits the requested action items, states that the budget was increased by 8% despite audio and slide evidence indicating an 8% reduction, and produces an unnecessarily long summary. FUV isolates these defects: the text verifier marks the missing action items, the audio verifier identifies the spoken budget cut, and the visual verifier identifies the revised timeline in the shared slide. CCV then confirms that the budget statement is a cross-modal factual contradiction rather than an acceptable wording difference. FPV uses the user's stable preference for concise, action-item-first minutes from compact memory to assess personalized satisfaction.

The scheduler preserves correct content while instructing the task tool to restore action items, correct the budget decision, move action items to the beginning, and shorten the remaining summary. The revised output includes responsible participants and deadlines, reports that the budget was reduced by 8% for Q3, and follows the preferred structure. The case illustrates that the three verification levels play complementary roles: FUV localizes defects, CCV adjudicates cross-modal evidence, and FPV determines how verified output characteristics relate to user satisfaction.

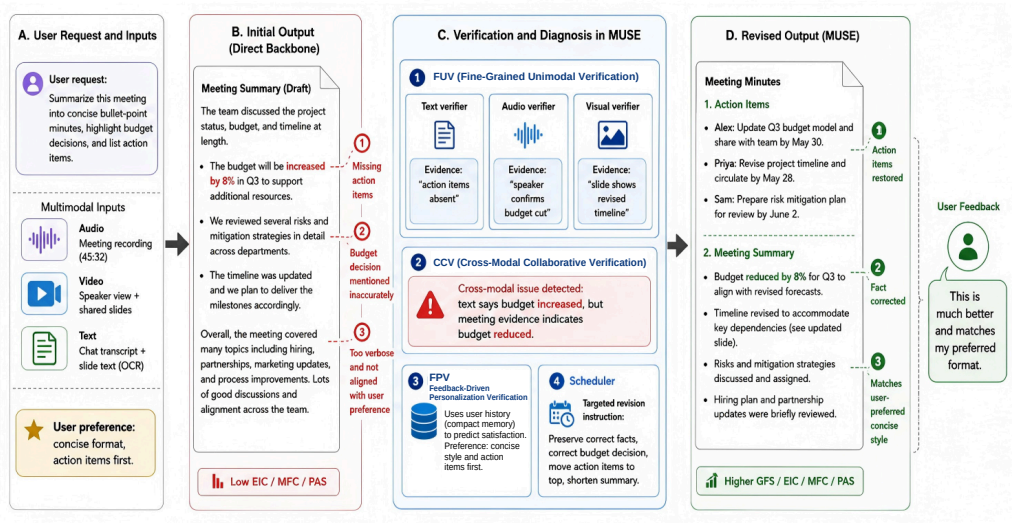


Figure 9: Qualitative case study on multimodal meeting-minutes generation. The direct backbone omits the requested action items, reverses the budget decision, and violates the user's preference for concise, action-item-first minutes. FUV identifies modality-specific evidence, CCV adjudicates the cross-modal contradiction, and FPV uses the persistent formatting preference supplied through compact memory to assess user alignment. The scheduler then issues a targeted revision that restores the action items, corrects the budget fact, and reorganizes the output, improving GFS, EIC, MFC, and PAS.

D Persistent User Memory

Related main text: Sections 3.2 and 3.5.

This section discusses privacy, consent, data governance, and failure modes associated with persistent user-specific state. The study uses feedback from actual system use by 34 participants. Online memory and adapter updates follow each user's interaction order, as described in Appendix E.1.

Table 69: Privacy and governance controls for persistent user memory (Part I).

Aspect	Required Control
Consent	Separate consent for research use and persistent personalization; renew it for materially different purposes.
User identifiers	Use salted pseudonymous identifiers and exclude raw account identifiers from released data.
PII removal	Combine automated detection with human review for direct identifiers.
Raw media	Release only when licensing and consent permit; otherwise provide identifiers, timestamps, annotations, or de-identified features.
Encryption	Encrypt persistent state at rest and use current transport-layer encryption in transit.

Table 70: Privacy and governance controls for persistent user memory (Part II).

Aspect	Required Control
Access control	Enforce user-level access, least privilege, and auditable retrieval and update logs.
Replay isolation	Draw replay samples only from the same user, without cross-user batch completion or memory mixing.
Deletion	Propagate deletion to full and compact memory, replay buffers, retrieval indexes, cached embeddings, and the user adapter.
Retention	Define separate limits for raw content, structured memory, embeddings, replay samples, and adapter parameters.
Fallback	Disable personalization when identity, consent, provenance, or memory relevance is uncertain.

Stored State and Data Minimization. MUSE maintains full interaction memory, compact preference memory, a same-user replay buffer, a retrieval index, and a lightweight user adapter. These states may encode recurring preferences, rejected outputs, satisfaction patterns, and inferred decision boundaries. A deployment should minimize stored raw content, retain structured summaries when possible, and attach provenance, collection time, purpose, confidence, and expiration metadata to every persistent item.

Consent and User Control. Persistent personalization should be opt-in and purpose-specific. Consent for research use of interaction data should be separated from consent for long-term personalization.

Users should be informed about which information is retained, how it affects future outputs, and which derived components are updated. They should be able to inspect, correct, disable, export, and delete persistent state. Reuse across a materially different task or modality should require renewed consent. Incidental statements should not be converted into durable preferences when personalization intent is ambiguous.

Benchmark Release Policy. The benchmark release policy restricts interaction histories to records authorized for research use and requires de-identification before release. Direct identifiers such as names, email addresses, phone numbers, account identifiers, and precise addresses are removed or replaced by salted pseudonymous identifiers. Raw media that cannot be redistributed is replaced by permitted identifiers, timestamps, derived annotations, or features. The histories arise from actual use by five experts, 17 graduate students, and 12 other students. The five experts also served as evaluation organizers and expert reviewers. Participants provided research-use consent, and source-permission records are held for third-party media. Personal identifiers are removed from shared records, and each user's memory and adapter are kept separate.

Security and Retention. A deployment-ready implementation should encrypt persistent state at rest and in transit, apply user-level access control, and maintain auditable read, write, retrieval, and deletion logs. Full memory, compact profiles, replay samples, cached embeddings, retrieval indexes, and adapter parameters require explicit retention periods and inactivity-based expiration. A deletion request must propagate to every derived state. Operational user memory should remain separated from data used for population-level training.

The concrete governance requirements for each persistent state component are summarized in Tables 69 and 70.

Tables 69 and 70 make explicit that consent, deletion, and retention must propagate beyond visible conversations to compact summaries, embeddings, replay samples, retrieval indexes, and user-specific parameters.

Failure Modes. Persistent memory introduces failures beyond ordinary generation errors. Stale preferences may remain active after user needs change; isolated evidence may be overgeneralized across tasks; sarcastic, contradictory, or malicious feedback may poison the replay buffer; account sharing or identity-resolution errors may contaminate another user's state; and context that is technically relevant may be socially inappropriate in the current setting. Deleting visible conversations without deleting compact summaries, embeddings, replay samples, and adapter effects can also create a false impression of erasure.

Mitigation and System Boundaries. MUSE reduces several risks through confidence filtering, recency weighting, task-conditioned retrieval, user-isolated replay, adapter regularization, and contradiction-aware feedback handling. These mechanisms are not a complete privacy solution. Deployment should additionally use consent-aware retrieval gates, preference expiration, abrupt-shift detection, provenance-aware deletion, and periodic user confirmation of high-impact preferences. For deployment, independently of the default fusion rule, disable personalization when identity is uncertain, consent has expired, or memory conflicts with the current explicit request; use task evidence from the wrapper, FUV, and CCV.

E Online Evaluation and Additional Protocol Details

Related main text: Sections 4–5.

This appendix clarifies the distinction between the 2,709-instruction content catalog and the 92,106 participant–instruction encounters used for each main-comparison method, together with the common chronological history supplied to compared methods.

E.1 Instruction Catalog and Per-User Interaction Order

The instruction catalog contains 2,709 distinct sample IDs across 18 tasks. The capability-group counts are 917, 913, and 879, and the personalization counts are 1,340 personalized and 1,369 non-personalized instructions. These entries define the content to be executed, not the number of method outputs.

The main evaluation uses a separate anonymous interaction log. For every main-comparison method, participant $u \in \{1, \ldots, 34\}$, and within-user interaction index $t \in \{1, \ldots, 2709\}$, the log links the anonymous participant index, interaction index, sample ID, generated output, and the answer-linked history available before that interaction. Thus every main-comparison method is executed on exactly $34 \times 2{,}709 = 92{,}106$ encounters. The catalog row order and sample-ID suffixes are not used as chronology; the per-user interaction index in the evaluation log is.

For controlled comparisons, all methods receive the same instruction and the same prefix of that participant's historical answer–feedback records at a given (u, t). A historical feedback label remains attached to the exact answer that received it and is not reassigned as feedback for another method's newly generated answer. The new output is benchmark-evaluated first; only information whose interaction index precedes the next decision may enter subsequent memory or adapter state. This preserves temporal causality while keeping the comparison history-aligned across methods. The chronology requires only anonymous indices and therefore does not require release of participant identity information.

E.2 Decision Configurations and Result Interpretation

The default MUSE configuration uses learned monotonic calibration and positive-ϵ reliability weighting, and accepts when $S_t \geq \tau_s$. It uses $\epsilon = 0.5$ and $\tau_s = 0.75$, with the hard gate and cold-start mask both disabled. FPV therefore has a positive weight even before valid user history is available; its later influence changes with maturity and prediction reliability. The 800-example development collection initializes FPV and fits the four score maps before benchmark evaluation, after which all maps and global decision settings remain fixed.

Table 71: Decision settings for the default and specified supplementary configurations.

Setting or result family	Learned Cal.	Hard gate	Cold mask
Full-MUSE default, Tables 1–3	Yes	No	No
Equal-weight fusion	Yes	No	No
Uncalibrated reliability fusion	No	No	No
Calibrated fusion without hard gate	Yes	No	Yes
Masked + gated fusion	Yes	Yes	Yes
Matched-history cold-start curve	Yes	No	Yes
Epsilon-only cold start	Yes	No	No
Masked cold start	Yes	No	Yes
Hard-violation comparison	Yes	No / Yes	No

If no answer reaches the acceptance criterion before the iteration limit, the final generated answer is returned and recorded as below threshold rather than accepted; supplementary gate variants apply the same terminal rule. Consequently, the gate blocks acceptance during revision but does not by itself guarantee a flawless last output. The supplementary cold-start mask also changes weight normalization, so masked-versus-epsilon comparisons should be interpreted as configuration-level comparisons rather than an isolated causal effect of masking.

E.3 Answer-Specific Feedback and Evaluation Histories

Participant feedback is answer-specific. Its FPV record stores the answer and verification state that received it; a historical label cannot be transferred to another method's new output for the same request. This separates *adaptation supervision* from *benchmark evaluation*.

At a given participant and interaction index, all main-comparison methods receive the same earlier answer–feedback history and produce separate outputs. Each output receives GFS, EIC, MFC, and, when applicable, PAS benchmark scores. These scores compare methods and do not become FPV feedback labels. After scoring, the next actual answer-linked historical record becomes available for later interactions. This preserves a common history without leaking current labels.