OMNI2WEB: BENCHMARKING AUDIOVISUAL WEB EDITING WITH WEAK REFERENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Screen-recorded web editing requests contain weak deictic expressions such as "this" and "there," whose referents depend on speech, cursor trajectories, page state, and edit history. Such requests require intent recovery beyond the explicit specifications assumed by many existing web-editing benchmarks. We introduce Omni2Web, a bilingual benchmark of 918 instances spanning 13,907 edit steps. It defines three complementary tracks: Direct Editing evaluates webpage editing from recordings, Instruction Recovery measures explicit intent recovery, and Instruction Utility tests whether recovered instructions can drive a fixed code executor. We evaluate 17 open- and closed-source models. The best models attain 51.17 on the Edit Fidelity Score (EFS) for Direct Editing and 49.14 on the Instruction Recovery Score (IRS); under the fixed executor, the strongest recovered instructions reach 51.08 EFS, still far below the 89.69 EFS obtained with oracle instructions. Step-level analyses show that correct grounding does not guarantee successful edits, while some Omni models recover instructions that the fixed coding model executes substantially better than their direct edits. Controlled ablations further demonstrate the value of temporally aligned audiovisual evidence, while alternative judges preserve the leader and broad ordering. Together, these findings reveal substantial headroom in multimodal intent recovery and code execution and highlight the promise of pairing Omni rewriters with coding models.

1 INTRODUCTION

"Move this below that, make these two match, and undo the previous color change." For a viewer, this is actionable; as text alone, it is incomplete because cursor position, speech timing, and prior edits convey its referents. The operation spans language, cursor movement, and visual context.

Existing evaluations largely avoid this ambiguity. Instructed-editing benchmarks begin with explicit textual specifications (Chi et al., 2026); web and computer-use benchmarks execute actions from explicit goals (Deng et al., 2023; Koh et al., 2024; Xie et al., 2024; Lù et al., 2024). GUI-centered work emphasizes screen understanding, localization, or action prediction (Baechler et al., 2024; Hong et al., 2024), while audio-visual benchmarks test temporal alignment without realizing the evidence as a code edit (Zhou et al., 2025). None directly tests whether a model can recover weakly expressed intent from a recording and turn it into a complete, safe webpage modification.

A single end-to-end editing score would also obscure two distinct failures. A model may fail because it cannot determine what the user intends to change, or because it understands the request but cannot implement it correctly in HTML. The former concerns multimodal reference resolution; the latter concerns code execution. Successful editing must also preserve page content unrelated to the requested changes. Evaluation should separate intent recovery, implementation, and preservation.

We introduce **Omni2Web**, a bilingual benchmark for web editing from screen recordings with weak references. Its 918 instances span 129 source webpages, 13,907 edit steps, and 12 operation types. Each aligns speech with cursor trajectories and includes long sequences, localized modifications, and revisions. Annotations separately capture target grounding, edit fulfillment, revision handling, and damage to unrelated content. Figure 1 illustrates the task and evaluation design.

Omni2Web defines three complementary tracks. **Track A: Direct Editing** asks a model to produce the edited webpage directly from the source HTML and recording. **Track B: Instruction Recovery**

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

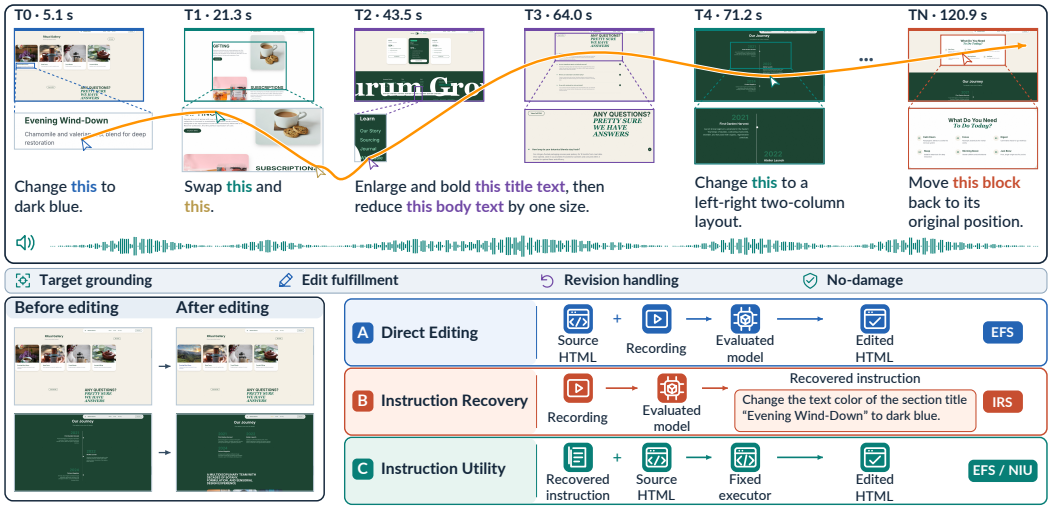


Figure 1: Overview of Omni2Web. The orange curve shows cursor movement across the recording. Tracks A–C evaluate direct editing, instruction recovery, and instruction utility.

asks it to rewrite the recording into explicit targets and editing actions. **Track C: Instruction Utility** passes these recovered instructions and the source HTML to a fixed coding model. Track C therefore pairs multimodal intent recovery with code generation, testing whether a fixed coding model can implement the instructions recovered by a multimodal model. Together, the three tracks provide complementary evidence about intent recovery and execution.

We evaluate 17 open- and closed-source models. Results leave substantial headroom: the best Direct Editing Edit Fidelity Score (EFS) is 51.17, the best Instruction Recovery Score (IRS) is 49.14, and the strongest recovered instructions score 51.08 EFS with 56.95 normalized instruction utility, versus 89.69 EFS for oracle instructions (*i.e.*, DeepSeek-v4-pro (Xu et al., 2026)). Step-level analyses reveal two complementary patterns: correct target grounding does not always translate into successful editing, while high No-damage can reflect inaction. Track C further shows that some Omni models recover useful instructions despite weak direct editing, allowing the fixed coding model to produce substantially stronger edits. Controlled ablations support the importance of synchronized audiovisual evidence, with native audiovisual input performing best for both tested Omni models. Shifting audio out of sync with video reduces IRS by up to 13.73 points, while adding timestamps to frame-and-transcript inputs improves it by up to 14.98 points. We contribute:

- We introduce Omni2Web, a bilingual benchmark comprising 918 screen-recorded editing requests, 13,907 edit steps, 129 source webpages, and 12 operation types, with synchronized audiovisual evidence and revision-aware final-state rubrics.
- We develop a three-track protocol that evaluates direct editing, component-level instruction recovery, and the downstream utility of recovered instructions under a fixed executor, while separately measuring preservation of unrelated page content.
- We benchmark 17 open- and closed-source models and conduct controlled modality, temporal-alignment, and executor studies. The results expose distinct grounding and execution bottlenecks, demonstrate the value of synchronized audiovisual evidence, and test evaluator reliability through alternative judges and human calibration.

2 Related Work

2.1 Web Editing and GUI Benchmarks

Web agents and instructed-editing benchmarks generally assume explicit textual goals (Deng et al., 2023; Koh et al., 2024; Chi et al., 2026; Dang et al., 2025). Video-based benchmarks differ in output: WebVR and IWR-Bench reconstruct complete webpages (Dai et al., 2026; Chen et al., 2026), VideoGUI predicts GUI actions (Lin et al., 2024), and Frontalk studies iterative front-end develop-

ment with multimodal feedback (Wu et al., 2025). Omni2Web reuses 129 WebVR source pages but collects new editing recordings and annotations. Starting from existing HTML, models must resolve weak speech–cursor references, apply multi-step revisions, and preserve unrelated content.

2.2 Multimodal Understanding and Grounded Interaction

Joint speech and pointing dates to "Put-That-There," which used gesture to resolve deictic expressions; later work established complementary roles for language, gesture, and visual context (Bolt, 1980; Oviatt, 1999). Modern work studies audio-visual reasoning (Goel et al., 2026; Li et al., 2026; Tao et al., 2026; Chao et al., 2026), embodied and video grounding (Shi & Yang, 2022; Ding et al., 2023; Wang et al., 2024), and GUI localization (Baechler et al., 2024; Hong et al., 2024; Cheng et al., 2024; You et al., 2024). These tasks output answers, locations, masks, or actions; Omni2Web requires recovered targets, actions, and revisions to yield final HTML, making multimodal grounding upstream to code editing.

3 Omni2Web

3.1 Task Instances and Annotations

Omni2Web evaluates single-turn, multi-step edits to an existing webpage. Each instance comprises the source page, synchronized media, transcript, and rubric:

$$\mathcal{X}_i = \left(H_i, V_i^{\mathrm{av}}, V_i^{\mathrm{sil}}, U_i, \tau_i, \mathcal{B}_i\right), \tag{1}$$

where H_i is the source HTML, V_i^{av} the original audiovisual screen recording, V_i^{sil} its silent-video counterpart, U_i the isolated audio, τ_i the manually written transcript, and $\mathcal{B}_i$ the step-level rubric. Auxiliary metadata is omitted from the notation. The model must use the spoken action together with the visual context to resolve expressions such as "this section" or "these two blocks," return a complete edited HTML document, and preserve unrelated parts of the source page.

Annotations decompose every instance into an ordered edit sequence. Each step has an operation type and two rubric items: one checks target grounding and the other checks edit fulfillment (or revision handling for a revision step). An instance-level no-damage item checks whether content outside the requested edits remains intact. The explicit reference instructions I_i^* are constructed deterministically from the paired rubric anchors in $\mathcal{B}_i$. We use M_i for the model-facing input assembled from V_i^{av}, V_i^{sil}, and τ_i under the protocol in Section 4.3.

3.2 Dataset Construction

We select 129 source webpages from WebVR (Dai et al., 2026). Annotators write multiple edit scripts per page, collectively covering the 12 operation types in Figure 2(a). The scripts specify the requested changes and their order. We review them before recording. Annotators then describe the requests naturally while pointing to the relevant page elements with the cursor. Weak references such as "here" and "these two" depend on speech, cursor pointing, and page context for their meaning. Annotators also manually write the transcripts τ_i and rubric anchors. The transcripts capture the spoken requests. The anchors specify each step's target and editing requirements. Before inclusion, we review the edit sequences and reference annotations for consistency in targets and edits.

A professional external team of 15 paid annotators collected and annotated the data. All received standardized training and provided informed consent for research use and public data release.

3.3 Text-Only Recoverability Probe

Some deictic requests remain recoverable without video when (τ_i, H_i) identifies a plausible target. We therefore measure text-only recoverability with a separate diagnostic probe. Gemini 3.1 Pro (Google DeepMind, 2026a) receives (τ_i, H_i) but no video or rubric anchors and predicts one target description per step. GPT-5.5 (OpenAI, 2026a) judges whether each prediction denotes the same page element or region as the corresponding human target anchor. Missing predictions score zero. This fixed evaluation pipeline is applied to all instances.

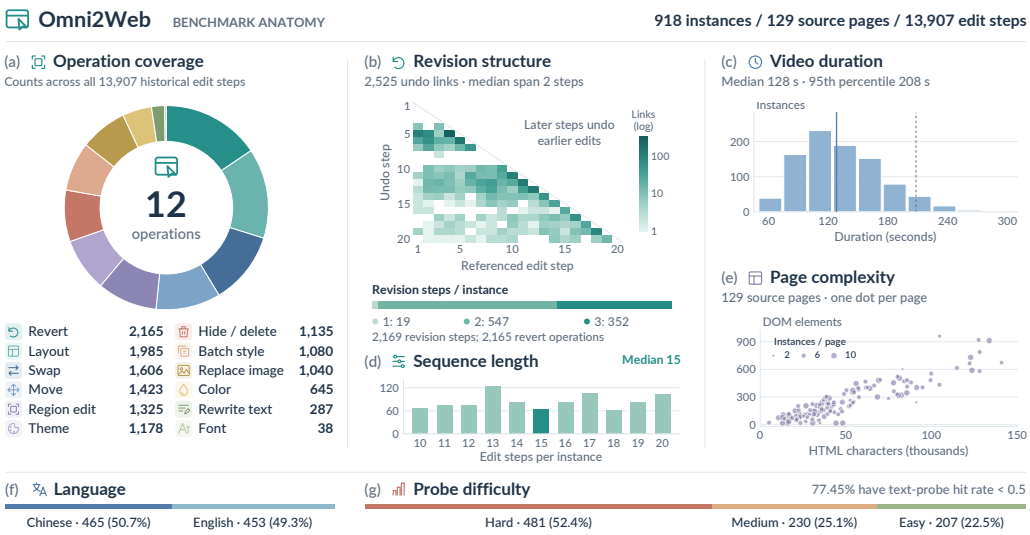


Figure 2: Dataset anatomy of Omni2Web. In (b), colors show link counts on a log scale; a revision can undo multiple steps. In (e), bubble area represents instances per source page.

For instance i with S_i steps, the probe hit rate is

$$h_i = \frac{1}{S_i} \sum_{j=1}^{S_i} \mathbf{1}\left[\widehat{t}_{ij} \equiv t_{ij}\right], \tag{2}$$

where $\widehat{t}_{ij}$ is the predicted target and t_{ij} is the human target anchor. We define Hard as $h_i < 0.2$, Medium as $0.2 \leq h_i < 0.5$, and Easy as $h_i \geq 0.5$. Low h_i means few recovered targets. These bands are diagnostic strata, not benchmark scores or measures of intrinsic human difficulty.

3.4 Dataset Statistics

Figure 2 summarizes 918 instances from 129 source webpages, covering 13,907 historical edit steps across 12 operation types. Sequences contain 10–20 steps, with a mean of 15.15. Every instance includes one to three revisions, totaling 2,169 revision-category steps. Their 2,525 undo links span a median of two steps (Figure 2(b)). Spoken languages are nearly balanced (465 Chinese, 453 English). The median video duration is 128.04 seconds. Under the text-only probe in Section 3.3, 481 instances are Hard, 230 are Medium, and 207 are Easy. Thus, 77.45% fall below a 0.5 recovery rate without video, indicating substantial dependence on visual and deictic evidence.

4 Evaluation Protocol

4.1 Three Evaluation Tracks

With the notation of Section 3.1, let f_θ^A and f_θ^B denote the evaluated model prompted for Tracks A and B, respectively, and e the fixed DeepSeek-v4-pro (Xu et al., 2026) executor. The tracks are

$$\begin{aligned} &\text{Track A:} && \widehat{H}_i^A = f_\theta^A(H_i, M_i), && \text{direct editing,} \\ &\text{Track B:} && \widehat{I}_i = f_\theta^B(M_i), && \text{instruction recovery,} \\ &\text{Track C:} && \widehat{H}_i^C = e(H_i, \widehat{I}_i), \quad H_i^{\text{oracle}} = e(H_i, I_i^*), && \text{instruction utility.} \end{aligned} \tag{3}$$

Track A produces an edited page, Track B recovers instructions without source HTML, and Track C measures their utility with a fixed executor that receives no media; the three contexts remain distinct.

4.2 TRACK-SPECIFIC SCORING

Tracks A&C. Tracks A and C share a media-free, rubric-anchored LLM judge (Zheng et al., 2023). Before aggregation, 2,525 earlier steps explicitly cancelled by later whole-step revisions are removed from the 13,907-step history, leaving 11,382 active steps; the cancelling revisions remain scored, while Track B retains the full history.

Given source H_i, prediction $\widehat{H}_i$, and rubric $\mathcal{B}_i$, the judge returns binary decisions for active target-grounding, edit-fulfillment, and revision items. Omissions score zero. An image check can only overturn passed replace-image items. Category averages give scores G_i, E_i, and R_i.

Separately, a structured HTML diff compares H_i with $\widehat{H}_i$. Let Δ_i be the extracted changes and $\mathcal{D}_i$ the collateral changes identified by the damage judge. Each damage receives a low, medium, or high penalty $w(d) \in \{0.05, 0.15, 0.30\}$. The No-damage score and EFS are

$$N_i = \max\left(0, 1 - \frac{\sum_{d\in\mathcal{D}_i} w(d)}{\max(\alpha|\Delta_i|, 1)}\right), \tag{4}$$

$$\mathrm{EFS}_i = 0.10G_i + 0.50E_i + 0.20N_i + 0.20R_i. \tag{5}$$

Here $|\Delta_i|$ is the number of extracted changes. We set the No-damage normalization coefficient to $\alpha = 0.15$, matching the medium penalty, while the unit floor stabilizes short or empty diffs. Category scores are averaged within each instance and EFS across instances. For Track C, instruction utility is normalized against the oracle executor run:

$$\mathrm{NIU} = \frac{\overline{\mathrm{EFS}}_{\mathrm{recovered}}}{\overline{\mathrm{EFS}}_{\mathrm{oracle}}}. \tag{6}$$

NIU is a ratio of dataset-level aggregates and is not clipped.

Track B. The model outputs an ordered JSON list of target/instruction pairs. One-to-one matching with I_i^* assigns binary target and action credit $(\pi_{ij}^T, \pi_{ij}^A) \in \{0,1\}^2$. Let $n_i^{\mathrm{ref}} = |I_i^*|$, $n_i^{\mathrm{pred}} = |\widehat{I}_i|$, and $m_i^c = \sum_j \pi_{ij}^c$ for $c \in \{T, A\}$. Component F1 and the Instruction Recovery Score (IRS) are

$$F_i^c = \frac{2m_i^c}{n_i^{\mathrm{ref}} + n_i^{\mathrm{pred}}}, \quad c \in \{T, A\}, \tag{7}$$

$$\mathrm{IRS}_i = 0.50F_i^T + 0.50F_i^A. \tag{8}$$

The semantic judge uses H_i only to verify element equivalence. Deterministic checks enforce one-to-one matches and valid indices. Duplicate, unmatched, and extra predictions increase n_i^{pred}, penalizing unsupported enumeration. Missing references and unusable outputs score zero. IRS is macro-averaged over instances. Across sensitivity tests, the Track A and C leaders persist under all 156 EFS configurations, while alternative IRS formulations retain $\rho \geq 0.983$ but can change the Track B leader (Appendices A.7 and A.8). Gemini 3.1 Pro judges all model-based Track A–C stages with fixed English prompts across languages and model families. Appendix A.11 lists them.

4.3 MODELS AND INPUT CONSTRUCTION

We evaluate 17 open- and closed-source models spanning omni-modal and vision-language (VL) input interfaces. The open-source group comprises MiniCPM-o 4.5 (Cui et al., 2026), Gemma 4 12B Unified (Team et al., 2026a), Qwen3.8-27B (Qwen Team, 2026), Nemotron 3 Nano Omni (Deshmukh et al., 2026), MiMo-V2.5 (Xiaomi, 2026), MiniMax-M3 (MiniMax, 2026), Qwen3.8-Max (Qwen Team, 2026), and Kimi-K3 (Team et al., 2026b). The closed-source group comprises Muse Spark 1.1 (Menghini et al., 2026), Seed2.0 Lite (ByteDance Seed, 2026a), Gemini 3.5 Flash (Google DeepMind, 2026b), Qwen3.5-Omni-Plus (Team, 2026), Seed2.1 Pro (ByteDance Seed, 2026b), Grok 4.6 (xAI, 2026), Gemini 3.1 Pro (Google DeepMind, 2026a), GPT-5.6 Sol (OpenAI, 2026b), and Claude Opus 5 (Anthropic, 2026). Table 1 marks each model's input modality. Cross-family results therefore compare end-to-end systems under native input interfaces, rather than architectures under identical information.

We assign each model one main input condition based on observed media support and use it consistently on Tracks A and B. Let $F_i = \mathrm{FPS}_1(V_i^{\mathrm{sil}})$ denote the timestamped frame sequence sampled

Table 1: Main results on the 918 instances for Direct Editing (Track A) and Instruction Recovery (Track B). Track B columns are component F1 scores and their average IRS; all values are percentages, with column bests in bold.

		Track A: Direct Editing					Track B: Instruction Recovery		
Model	**Modality**	**Grd.**	**Edit**	**No dmg.**	**Rev.**	**EFS**	**Target F1**	**Action F1**	**IRS**
Open-Source Models									
MiniCPM-o 4.5	Omni	–	–	–	–	–	17.87	31.93	24.90
Gemma 4 12B	Omni	–	–	–	–	–	1.88	11.67	6.78
Qwen3.8-27B	VL	12.25	7.56	**82.94**	26.74	26.94	19.82	39.85	29.84
Nemotron 3 Nano	Omni	0.75	0.04	71.27	11.46	16.64	1.74	8.26	5.00
MiMo-V2.5	Omni	31.44	17.93	43.59	39.81	28.79	33.95	40.67	37.31
MiniMax-M3	VL	31.12	18.83	33.96	50.87	29.49	21.25	37.38	29.32
Qwen3.8-Max	VL	43.51	27.59	46.52	55.77	38.60	34.84	41.90	38.37
Kimi-K3	VL	43.32	30.26	49.23	56.59	40.63	38.64	45.25	41.95
Closed-Source Models									
Muse Spark 1.1	Omni	22.29	10.14	45.99	37.91	24.08	18.45	32.37	25.41
Seed2.0 Lite	Omni	20.38	11.70	63.36	38.87	28.33	47.91	45.14	46.52
Gemini 3.5 Flash	Omni	**59.20**	**40.52**	53.71	**71.26**	**51.17**	47.04	47.88	47.46
Qwen3.5-Omni-Plus	Omni	20.79	11.86	75.46	32.28	29.56	**49.99**	48.29	**49.14**
Seed2.1 Pro	VL	37.54	23.67	37.40	57.03	34.48	33.21	44.96	39.08
Grok 4.6	VL	42.52	29.92	59.78	51.43	41.45	34.60	47.06	40.83
Gemini 3.1 Pro	Omni	26.93	16.46	74.09	41.67	34.08	42.19	44.18	43.19
GPT-5.6 Sol	VL	55.92	37.32	29.55	63.49	42.86	47.51	49.17	48.34
Claude Opus 5	VL	54.11	38.90	50.88	61.67	47.37	48.06	**49.76**	48.91

once per second. The model-facing input is

$$M_i = \begin{cases} V_i^{\mathrm{av}}, & \text{Omni,} \\ (V_i^{\mathrm{sil}}, \tau_i), & \text{VL with native-video delivery,} \\ (F_i, \tau_i), & \text{VL with frame delivery.} \end{cases} \tag{9}$$

5 Main Results

5.1 Main Leaderboard

Tables 1 and 2 report performance on all three tracks for the 918 instances. Track C reports the absolute Edit Fidelity Score (EFS) and normalized instruction utility (NIU), normalized by the 89.69 oracle-executor EFS. Higher values are better. MiniCPM-o 4.5 and Gemma 4 12B lack Track A scores because the direct-editing input exceeds their context limits.

Gemini 3.5 Flash leads Direct Editing with 51.17 EFS, whereas Qwen3.5-Omni-Plus has the highest Instruction Recovery point estimate, at 49.14 IRS, and leads Instruction Utility with 56.95 NIU. Omni models have the highest point estimates on all three tracks. Only Gemini exceeds 50 on Track A; no model exceeds 50 IRS, and the best recovered instructions reach 51.08 EFS under the fixed executor versus 89.69 for oracle instructions, leaving substantial headroom.

5.2 Direct Editing Performance

Correct grounding often fails to produce a correct edit. Gemini 3.5 Flash ranks first in Direct Editing. Yet only 42.59% of its 11,382 scored steps receive both grounding and edit credit; in another 16.68%, it finds the correct target but fails to execute the requested edit (Figure 3(a)). Edit credit without grounding credit never exceeds 3.83% for any model. Thus, execution after localization is the main bottleneck, as models often identify the target without implementing the requested change.

High No-damage can mask inaction. Qwen3.8-27B has the highest No-damage score (82.94) but only 7.56 in Edit Fulfillment; Qwen3.5-Omni-Plus scores 75.46 and 11.86, while Nemotron 3 Nano scores 71.27 and 0.04. These models leave much of the original page intact partly because they perform few requested edits. EFS therefore evaluates preservation jointly with grounding, edit fulfillment, and revision handling instead of rewarding No-damage in isolation.

Table 2: Instruction Utility (Track C) with the fixed DeepSeek-v4-pro executor. EFS and its components are absolute scores, whereas NIU is normalized by oracle EFS.

		Absolute EFS Components				Track C: Instruction Utility	
Model	**Modality**	**Grd.**	**Edit**	**No dmg.**	**Rev.**	**EFS**	**NIU**
Oracle Executor							
DeepSeek-v4-pro	Text	**95.79**	**91.69**	**74.01**	**97.35**	**89.69**	–
Open-Source Models							
MiniCPM-o 4.5	Omni	38.20	21.21	41.37	48.75	32.45	36.18
Gemma 4 12B	Omni	2.58	0.71	61.89	15.60	16.11	17.96
Qwen3.8-27B	VL	28.93	15.65	43.26	45.55	28.48	31.75
Nemotron 3 Nano	Omni	2.65	0.77	**69.30**	15.38	17.59	19.61
MiMo-V2.5	Omni	46.42	28.16	43.04	52.69	37.87	42.22
MiniMax-M3	VL	31.86	16.79	39.36	46.10	28.67	31.97
Qwen3.8-Max	VL	46.05	28.04	45.67	58.81	39.52	44.06
Kimi-K3	VL	47.59	30.36	44.61	60.26	40.91	45.62
Closed-Source Models							
Muse Spark 1.1	Omni	33.71	12.72	29.30	43.88	24.37	27.17
Seed2.0 Lite	Omni	55.30	37.68	52.10	63.18	47.42	52.87
Gemini 3.5 Flash	Omni	59.16	39.58	51.42	**68.92**	49.77	55.49
Qwen3.5-Omni-Plus	Omni	**61.39**	**41.72**	53.06	67.36	**51.08**	**56.95**
Seed2.1 Pro	VL	43.94	28.39	43.66	56.68	38.66	43.10
Grok 4.6	VL	44.72	28.12	45.53	54.81	38.60	43.04
Gemini 3.1 Pro	Omni	53.81	34.84	47.91	65.74	45.53	50.76
GPT-5.6 Sol	VL	56.70	37.94	45.98	61.78	46.19	51.50
Claude Opus 5	VL	56.78	39.34	46.13	62.49	47.07	52.48

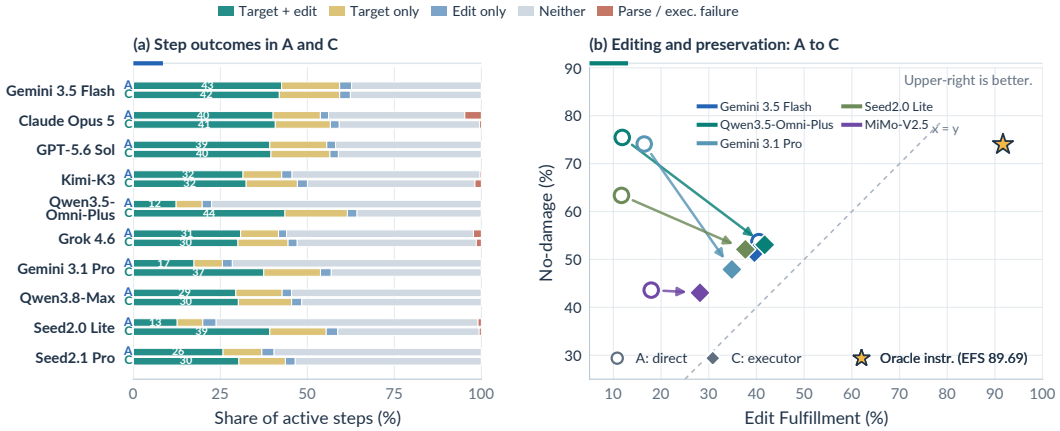


Figure 3: Track A/C diagnostics. (a) Step outcomes for the ten models with highest summed A/C EFS. (b) A-to-C changes for the five highest-scoring Omni models by summed A/C EFS. Open circles and filled diamonds denote Tracks A and C; the star denotes oracle instructions.

5.3 Instruction Recovery and Downstream Utility

Some Omni models recover intent better than they execute it. Among 15 complete models, Track A EFS and Track B IRS correlate moderately ($\rho = 0.686$), with Qwen3.5-Omni-Plus and Seed2.0 Lite rising from ninth and twelfth to first and fifth, respectively (Figure 4(a)). The fixed executor raises their EFS by 21.52 and 19.09 points. Track C exceeds Track A for 11 of 15 pairs and improves joint correctness by 3.77–14.77 points across all 12 edit types (Appendix A.2).

Lower No-damage is not necessarily a regression. Figure 3(b) shows that Qwen3.5-Omni-Plus increases Edit Fulfillment from 11.86 to 41.72 and EFS from 29.56 to 51.08, while No-damage decreases from 75.46 to 53.06. Seed2.0 Lite follows the same pattern: Edit Fulfillment rises from 11.70 to 37.68 and EFS from 28.33 to 47.42, while No-damage falls from 63.36 to 52.10. Both cases show more requested editing and higher EFS despite lower preservation.

Recovery predicts downstream utility, but a large oracle gap remains. Across 17 models, Track B IRS strongly correlates with Track C NIU (Pearson $r = 0.965$; Spearman $\rho = 0.951$;

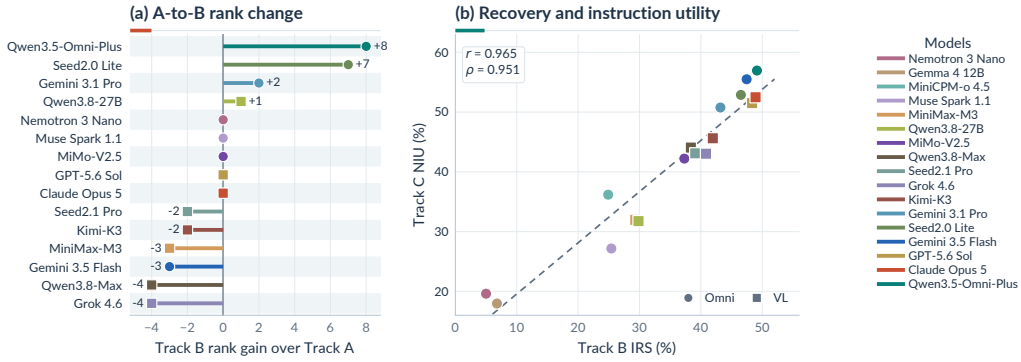


Figure 4: Cross-track relationships. (a) A-to-B rank change (positive favors B). (b) Track B IRS versus Track C NIU with a least-squares fit. Colors denote models; shapes denote Omni/VL input.

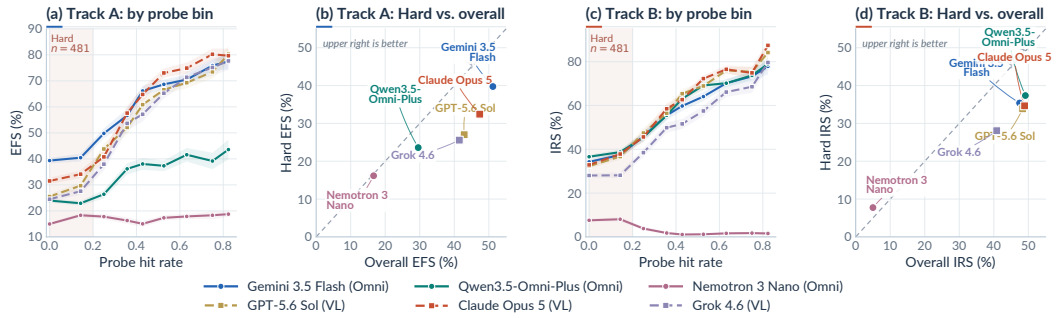


Figure 5: Scores by probe bin (a,c) and overall versus Hard (b,d; $h_i < 0.2$, $n = 481$).

Figure 4(b)). Qwen3.5-Omni-Plus scores 51.08 EFS and 56.95 NIU on Track C versus 89.69 oracle EFS with the same executor. On a 300-instance subset, self-execution leaves EFS gaps between oracle and recovered instructions of 36.96 for Gemini and 22.24 for Qwen. Thus, the external executor alone does not explain the gap (Appendix A.4).

6 Analysis and Robustness

6.1 Visual Dependence and Audiovisual Ablations

Omni models show their strongest advantage on Hard instances. On the 481 Hard instances, Gemini 3.5 Flash has the highest observed Track A score at 39.74 EFS, while Qwen3.5-Omni-Plus has the highest Track B score at 37.36 IRS (Figure 5 and Figure 6). Proximity to $y = x$ in Figure 5(b,d) indicates less Hard-subset degradation. Gemini 3.5 Flash drops by 11.43 points on Track A, versus 14.92–15.87 for the three leading VL models; on Track B, Qwen3.5-Omni-Plus drops by 11.78 versus Claude Opus 5's 14.27, increasing its observed margin from 0.23 overall to 2.72 on Hard. This supports the use of cursor–speech timing beyond text-recoverable cues.

Synchronized audiovisual input is strongest for both tested Omni models. Relative to transcript only, native audiovisual input improves Track A, B, and C by 18.41, 29.28, and 31.78 points for Gemini 3.5 Flash and by 5.97, 20.49, and 20.01 points for Qwen3.5-Omni-Plus. Relative to silent video with transcript, the gains span 9.39–16.80 and 7.13–19.82 points, respectively (Table 3). A complementary timing intervention separates synchronization from content: shifting audio by ± 5 seconds while preserving its content lowers IRS by 11.65 points for Gemini and 13.73 for Qwen across 300 instances. Conversely, adding timestamps to the same manual transcripts and 1 FPS frames raises IRS by 8.76 points for GPT-5.6 Sol and 14.98 for Grok 4.6 (Table 10). Both timing interventions affect target grounding more than action recovery, supporting the importance of temporal correspondence for resolving references (Appendix A.3).

Table 3: Input-modality ablation on the 918 instances. Tracks A/C report EFS and Track B reports Component-F1 IRS; bold marks each model's best input.

Model	Input	Track A	Track B	Track C
Gemini 3.5 Flash	Transcript only	32.76	18.18	18.00
	Silent video + transcript	34.37	38.06	34.96
	Audio-video	**51.17**	**47.46**	**49.77**
Qwen3.5-Omni-Plus	Transcript only	23.59	28.65	31.07
	Silent video + transcript	22.43	32.58	31.26
	Audio-video	**29.56**	**49.14**	**51.08**

Table 4: Judge robustness for six models on 300 Track A instances. Correlations use Gemini 3.1 Pro as reference; $|\Delta|$ is in EFS points.

Alternative judge	Pearson r	Spearman ρ	Kendall τ	Top model	Mean $\lvert\Delta\rvert$
GPT-5.5	0.983	0.943	0.867	Gemini 3.5 Flash	5.16
DeepSeek-v4-pro	0.998	1.000	1.000	Gemini 3.5 Flash	2.09

6.2 Judge Robustness

Judge replacement preserves the leader and broad model ordering. To test self-preference (Panickssery et al., 2024), we use a fixed evaluator-validation set of 300 instances for six models. Relative to Gemini 3.1 Pro, GPT-5.5 yields Spearman $\rho = 0.943$ and Kendall $\tau = 0.867$, while DeepSeek-v4-pro preserves the ordering exactly; Gemini 3.5 Flash remains first (Table 4).

Absolute scores vary more: mean absolute differences from the primary judge are 5.16 EFS for GPT-5.5 and 2.09 for DeepSeek-v4-pro. We find no consistent same-family advantage; relative to the other judges' mean, same-family changes are $+1.43$ for Gemini 3.5 Flash, -4.68 for Gemini 3.1 Pro, and -0.31 for GPT-5.6 Sol. Appendix A.5 reports the complete matrix. A blinded human evaluation of 1,451 steps finds strong output-level correlations for Track A grounding and edit fulfillment and Track B IRS (up to $r = 0.906$; Appendix A.6).

6.3 Metric and Aggregation Robustness

The Track A and C leaders are stable across tested EFS parameterizations. Across 156 weight and No-damage settings, the leaders never change; 150 retain Spearman $\rho \geq 0.9$ on Track A, and all do so on Track C. The largest shift, under a No-damage-heavy vector with $\alpha = 0.05$, moves Qwen3.8-27B from thirteenth to fifth on Track A, so middle ranks should not be overinterpreted. Alternative Track B formulations yield $\rho \geq 0.983$ but change the leader under precision-heavy or joint scoring (Appendices A.7 and A.8).

Page dependence and zero-filled outputs mainly affect close races. Page bootstrap ranks Gemini first on A in 99.99% of resamples and Qwen first on B and C in 55.67% and 98.63%, respectively. Qwen and Seed show positive A-to-C gains throughout. Valid-only scores retain all leaders and Track B's top three; no model–track exceeds 41 zero-filled outputs (Appendices A.9 and A.10).

7 Conclusion

Omni2Web evaluates weakly referential screen-recorded web editing through direct editing, instruction recovery, and downstream utility. Results across 17 models reveal substantial headroom in both multimodal understanding and execution: models often locate the correct element yet fail to implement the requested change. Controlled modality and temporal-alignment experiments further show that synchronized speech and visual context are important for recovering user intent. Meanwhile, some Omni models produce instructions that a coding model executes substantially better than their direct edits, demonstrating the promise of separating intent recovery from code generation. The remaining gap to oracle instructions leaves considerable room to improve both stages. Omni2Web shifts web-editing evaluation from complete specifications to incomplete, situated user intent.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing methods, cleaning and reformatting the dataset, translation, and providing feedback on research methodology and experiments; these tasks were led by the authors, with AI tools in an assisting role. We have not used generative AI tools to generate synthetic datasets, develop conceptual frameworks, propose or refine hypotheses, support qualitative or thematic analysis, or interpret results; these tasks were carried out entirely by the authors. Formulating mathematical claims and constructing or writing proofs are not applicable to this work. Additionally, we used generative AI tools to polish the writing. LLMs that are evaluated or used as judges in Omni2Web are part of the benchmark protocol described in Sections 3 and 4. The authors have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Claude Opus 5. https://www.anthropic.com/news/claude-opus-5, 2026. Accessed: 2026-08-21.

Gilles Baechler, Srinivas Sunkara, Maria Wang, Fedir Zubach, Hassan Mansoor, Vincent Etter, Victor Cărbune, Jason Lin, Jindong Chen, and Abhanshu Sharma. Screenai: A vision-language model for ui and infographics understanding. *arXiv preprint arXiv:2402.04615*, 2024.

Richard A Bolt. "put-that-there" voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pp. 262–270, 1980.

ByteDance Seed. Seed2.0. https://seed.bytedance.com/en/seed2, 2026a. Accessed: 2026-08-21.

ByteDance Seed. Seed2.1. https://seed.bytedance.com/en/seed2_1, 2026b. Accessed: 2026-08-21.

Jianghan Chao, Wenhui Tan, Yuchong Sun, Ruihua Song, Liyun Ru, et al. Jointavbench: A benchmark for joint audio-visual reasoning evaluation. In *International Conference on Learning Representations*, volume 2026, pp. 84035–84062, 2026.

Yang Chen, Minghao Liu, Yufan Shen, Yunwen Li, Tianyuan Huang, Xinyu Fang, Tianyu Zheng, Wenxuan Huang, Cheng Yang, Licheng Wen, et al. Iwr-bench: Can lvlms reconstruct interactive webpage from a user interaction video? In *International Conference on Learning Representations*, volume 2026, pp. 52158–52189, 2026.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Li YanTao, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9313–9332, 2024.

Wayne Chi, Valerie Chen, Ryan Shar, Aditya Mittal, Jenny Liang, Wei-Lin Chiang, Anastasios Angelopoulos, Ion Stoica, Graham Neubig, Ameet Talwalkar, et al. Editbench: Evaluating llm abilities to perform real-world instructed code edits. In *International Conference on Learning Representations*, volume 2026, pp. 94809–94835, 2026.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

Junbo Cui, Bokai Xu, Chongyi Wang, Tianyu Yu, Weiyue Sun, Yingjing Xu, Tianran Wang, Zhihui He, Wenshuo Ma, Tianchi Cai, et al. Minicpm-o 4.5: Towards real-time full-duplex omni-modal interaction. *arXiv preprint arXiv:2604.27393*, 2026.

Yuhong Dai, Yanlin Lai, Mitt Huang, Hangyu Guo, Dingming Li, Hongbo Peng, Haodong Li, Yingxiu Zhao, Haoran Lyu, Zheng Ge, et al. Webvr: Benchmarking multimodal llms for webpage recreation from videos via human-aligned visual rubrics. *arXiv preprint arXiv:2603.13391*, 2026.

Truong Hai Dang, Jingyu Xiao, and Yintong Huo. Envisioning future interactive web development: Editing webpage with natural language. In *2025 2nd IEEE/ACM International Conference on AI-powered Software (AIware)*, pp. 61–66. IEEE, 2025.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

Amala Sanjay Deshmukh, Kateryna Chumachenko, Tuomas Rintamaki, Matthieu Le, Tyler Poon, Danial Mohseni Taheri, Ilia Karmanov, Guilin Liu, Jarno Seppanen, Arushi Goel, et al. Nemotron 3 nano omni: efficient and open multimodal intelligence. *arXiv preprint arXiv:2604.24954*, 2026.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2694–2703. IEEE, 2023.

Christopher A Field and Alan H Welsh. Bootstrapping clustered data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(3):369–390, 2007.

Arushi Goel, Sreyan Ghosh, Vatsal Agarwal, Nishit Anand, Kaousheik Jayakumar, Lasha Koroshinadze, Yao Xu, Katie Lyons, James Case, Karan Sapra, et al. Mmou: A massive multi-task omni understanding and reasoning benchmark for long and complex real-world videos. *arXiv preprint arXiv:2603.14145*, 2026.

Google DeepMind. Gemini 3.1 Pro model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, 2026a. Accessed: 2026-08-21.

Google DeepMind. Gemini 3.5 Flash model card. `https://deepmind.google/models/model-cards/gemini-3-5-flash/`, 2026b. Accessed: 2026-08-21.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14281–14290. IEEE, 2024.

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 (Volume 1: Long Papers)*, pp. 881–905, 2024.

Caorui Li, Yu Chen, Yiyan Ji, Jin Xu, Zhenyu Cui, Shihao Li, Yuanxing Zhang, Zhenghao Song, Dingling Zhang, Ying He, et al. Omnivideobench: Towards audio-visual understanding evaluation for omni mllms. In *International Conference on Learning Representations*, volume 2026, pp. 138214–138236, 2026.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Qinchen Wu, Mingyi Yan, Zhengyuan Yang, Lijuan Wang, and Mike Zheng Shou. Videogui: A benchmark for gui automation from instructional videos. In *NeurIPS*, 2024.

Xing Han Lù, Zdeněk Kasner, and Siva Reddy. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*, 2024.

Cristina Menghini, Peter Ney, Hamza Kwisaba, Miles Turpin, Felix Binder, Jean-Christophe Testud, Aidan Boyd, Nathaniel Li, Ivan Evtimov, Klaudia Krawiecka, et al. Muse spark safety & preparedness report. *arXiv preprint arXiv:2606.12429*, 2026.

MiniMax. MiniMax-M3. `https://www.minimaxi.com/models/text/m3`, 2026. Accessed: 2026-08-21.

OpenAI. GPT-5.5 System Card. `https://openai.com/index/gpt-5-5-system-card/`, 2026a. Accessed: 2026-08-31.

OpenAI. GPT-5.6. `https://deploymentsafety.openai.com/gpt-5-6`, 2026b. Accessed: 2026-08-21.

Sharon Oviatt. Ten myths of multimodal interaction. *Communications of the ACM*, 42(11):74–81, 1999.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802, 2024.

Qwen Team. Qwen3.8. https://qwen.ai/blog?id=qwen3.8, 2026. Accessed: 2026-08-21.

Cheng Shi and Sibei Yang. Spatial and visual perspective-taking via view rotation and relation reasoning for embodied reference understanding. In *European Conference on Computer Vision*, pp. 201–218. Springer, 2022.

Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al. Qwen3-asr technical report. *arXiv preprint arXiv:2601.21337*, 2026.

Keda Tao, Yuhua Zheng, Jia Xu, Wenjie Du, Kele Shao, Hesong Wang, Xueyi Chen, Xin Jin, Junhan Zhu, Bohan Yu, et al. Lvomnibench: Pioneering long audio-video understanding evaluation for omnimodal llms. *arXiv preprint arXiv:2603.19217*, 2026.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026a.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026b.

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models. *arXiv preprint arXiv:2410.03290*, 2024.

Xueqing Wu, Zihan Xue, Da Yin, Shuyan Zhou, Kai-Wei Chang, Nanyun Peng, and Yeming Wen. Frontalk: Benchmarking front-end development as conversational code generation with multimodal feedback. *arXiv preprint arXiv:2601.04203*, 2025.

xAI. Grok 4.6. https://x.ai/news/grok-4-6, 2026. Accessed: 2026-08-25.

Xiaomi. MiMo-V2.5. https://mimo.xiaomi.com/mimo-v2-5/, 2026. Accessed: 2026-08-21.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, and Zhe Gan. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In *European Conference on Computer Vision*, pp. 240–255. Springer, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Ziwei Zhou, Rui Wang, Zuxuan Wu, and Yu-Gang Jiang. Daily-omni: Towards audio-visual reasoning with temporal alignment across modalities. *arXiv preprint arXiv:2505.17862*, 2025.

A APPENDIX

Contents

A.1 Discussion and Threats to Validity . . . 13
A.2 Operation-Level Analysis . . . 14
A.3 Controlled Modality and Temporal Alignment Ablations . . . 15
A.4 Same-Model Execution Control . . . 17
A.5 Alternative-Judge Ablation . . . 17
A.6 Human Calibration of the LLM Judge . . . 18
A.7 EFS Hyperparameter Sensitivity . . . 19
A.8 Track B Metric Sensitivity . . . 19
A.9 Source-Page Cluster Bootstrap . . . 20
A.10 Coverage and Zero-Fill Robustness . . . 21
A.11 Prompt Documentation . . . 22
A.11.1 Contestant and Executor Prompts . . . 22
Track A: Direct Editing . . . 22
Track B: Instruction Recovery . . . 23
Track C: Fixed Executor . . . 26
A.11.2 Scoring Prompts . . . 27
Tracks A and C Rubric Judge . . . 27
Track B Matching Judge . . . 28
No-Damage Judge . . . 30
Replacement-Image Content Judge . . . 31
A.11.3 Difficulty-Probe Prompts . . . 31
Text-Only Target Recovery . . . 31
Target-Equivalence Judge . . . 32

A.1 DISCUSSION AND THREATS TO VALIDITY

The three-track design provides complementary evidence about intent recovery and execution. Track A evaluates direct webpage editing from a screen recording, Track B isolates explicit-instruction recovery, and Track C tests whether a fixed code executor can turn those instructions into correct modifications. Some Omni models have limited direct-editing ability yet recover user intent useful to the downstream executor. Combining multimodal intent understanding with specialized code execution therefore offers a practical route for handling weakly referential editing requests. The tracks are complementary diagnostic views, not interchangeable leaderboards.

Model-based scoring only approximates editing quality. EFS and IRS convert complex webpage edits into reproducible, rubric-based scores, but cannot fully replace human evaluation (Zheng et al., 2023). We use Gemini 3.1 Pro (Google DeepMind, 2026a) as the primary judge and test robustness with GPT-5.5 (OpenAI, 2026a) and DeepSeek-v4-pro (Xu et al., 2026). Absolute scores differ, but the leading model and broad ordering remain stable. The EFS category weights and the 0.15 damage penalty are prespecified heuristics rather than data-derived quantities. Appendix A.7 varies both choices across 156 configurations. Appendix A.8 likewise finds high rank correlations across alternative IRS formulations, although some alternatives change the exact leader.

Track C depends on the fixed executor. All Track C results execute recovered instructions with DeepSeek-v4-pro (Xu et al., 2026), holding execution capability fixed across models. NIU, absolute Track C scores, and rankings may nevertheless change with the executor. The 89.69 EFS from oracle instructions is an executor-specific reference, not a theoretical task ceiling.

The benchmark's scope limits external validity. Omni2Web contains 918 instances constructed from 129 source webpages and covers 12 webpage-editing operation types. It focuses on multi-step modifications to existing HTML pages and does not represent the full range of open-ended software engineering, live website interaction, or general GUI-agent tasks. Online API results are snapshots of particular model versions and evaluation dates. Zero-filling refused, empty, or otherwise unusable outputs, together with valid-only robustness and coverage reporting, reduces selective bias but cannot eliminate model-version drift or output nondeterminism.

A.2 Operation-Level Analysis

We align each of the 13,907 reference steps with one of the 12 operation types and compare the 15 models that have results on all three tracks. For Tracks A and C, a step is jointly correct when both its target-grounding and edit-fulfillment or revision decisions pass; for Track B, both the recovered target and action must match. We first compute micro accuracy over steps of each operation within a model and then macro-average across models. These diagnostic rates are not EFS or IRS: they exclude the instance-level No-damage term and, unlike Component-F1 IRS, have no prediction-level precision term because extra predictions cannot be assigned to a reference operation. Table 5 reports the cross-model macro-average, while Table 6 gives the corresponding per-model rates for the two leading Omni systems.

Table 5: Macro-averaged step-level joint correctness (%) by operation over the 15 models evaluated on all three tracks. B retains the full history; A/C use active steps. Δ is Track C minus Track A.

Operation	B steps	A/C steps	Track A	Track B	Track C	Δ C–A
Batch style	1,080	909	17.32	19.99	21.09	+3.77
Change color	645	643	24.99	35.00	34.18	+9.20
Change font	38	37	27.39	42.11	42.16	+14.77
Change layout	1,985	1,553	25.43	41.43	33.05	+7.61
Change theme	1,178	884	19.46	28.30	26.13	+6.67
Hide/delete	1,135	996	19.59	22.65	25.41	+5.82
Move	1,423	958	11.89	16.03	16.88	+4.99
Region edit	1,325	983	19.92	28.36	27.60	+7.68
Replace image	1,040	890	22.51	47.26	29.42	+6.91
Revert	2,165	2,164	31.55	28.79	40.56	+9.01
Rewrite text	287	287	36.77	42.53	42.21	+5.44
Swap	1,606	1,078	19.06	32.03	26.80	+7.74

Table 6: Per-model step-level joint correctness (%) by operation for Gemini 3.5 Flash and Qwen3.5-Omni-Plus.

Operation	Gemini 3.5 Flash			Qwen3.5-Omni-Plus		
	A	B	C	A	B	C
Batch style	33.33	26.85	27.50	6.60	28.98	29.15
Change color	57.08	52.71	52.41	18.20	60.62	60.50
Change font	59.46	57.89	59.46	18.92	55.26	51.35
Change layout	45.20	50.03	43.40	16.03	55.11	45.07
Change theme	40.61	39.56	41.52	6.56	38.46	38.80
Hide/delete	35.54	31.89	36.35	10.34	34.98	39.46
Move	22.86	20.80	24.43	1.77	23.12	24.22
Region edit	45.78	37.89	43.95	9.26	38.79	45.98
Replace image	40.00	57.88	38.76	15.28	69.04	44.27
Revert	53.33	31.04	53.10	17.51	43.74	53.23
Rewrite text	44.95	42.16	44.25	25.09	48.43	49.13
Swap	40.17	44.65	43.97	10.11	49.38	44.25

The fixed-executor pipeline improves joint correctness for every operation. Track C exceeds Track A by 3.77–14.77 points across all 12 types. The largest increase is on font changes, although that category contains only 37 steps; among operations with at least 1,000 steps, Revert gains 9.01 points. This breadth indicates that the pipeline advantage is not confined to one editing category.

The two leading Omni models exhibit different operation profiles. Gemini 3.5 Flash (Google DeepMind, 2026b) is strongest in direct editing for Change Font (59.46), Change Color (57.08), and Revert (53.33). Qwen3.5-Omni-Plus (Team, 2026) reaches 25.09 only on Rewrite Text in Track A, but its Track C score exceeds its Track A score for all 12 operations, with gains of 22.44–42.30 points. This pattern is consistent with stronger instruction recovery and fixed-executor performance than direct code execution.

Recovery-to-execution gaps vary across operations. The Track A and Track C operation rankings are strongly aligned ($\rho = 0.958$), and Move is the lowest-scoring operation on every track (11.89, 16.03, and 16.88 joint accuracy on A, B, and C). The largest gaps occur for Replace Image (47.26

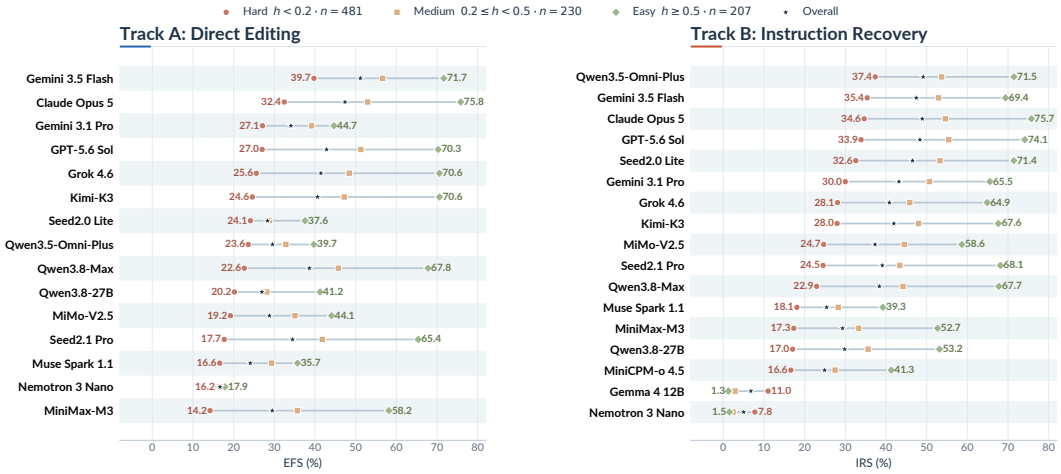


Figure 6: All-model performance across probe-defined visual-dependence strata. Lines connect each model's Hard, Medium, and Easy scores; stars mark overall averages.

on B versus 29.42 on C), Layout (41.43 versus 33.05), and Swap (32.03 versus 26.80). For these operations, models receive target-and-action credit more often than the downstream pages receive joint correctness credit.

A.3 Controlled Modality and Temporal Alignment Ablations

Input modality. Table 3 summarizes the aggregate results in the main paper; Tables 7 and 8 report every component score. Every row uses the same 918 instances and Gemini 3.1 Pro (Google DeepMind, 2026a) judge. The transcript-only condition supplies the manual step-aligned transcript but no video; the silent-video condition supplies the video without its audio channel together with the transcript; and the audio-video condition supplies the original audio-bearing video without an injected transcript. Track C changes only the input used to recover instructions and retains DeepSeek-v4-pro (Xu et al., 2026) as the fixed executor.

Table 7: Full controlled input-modality results for Tracks A and B on the 918 instances. Input abbreviations are T, V+T, and A+V. Scores are percentages; bold marks each model's best condition.

Model	Input	A Grd.	A Edit	A No dmg.	A Rev.	A EFS	B Target F1	B Action F1	B IRS
Gemini 3.5 Flash	T	30.88	20.50	49.35	47.77	32.76	4.90	31.46	18.18
	V+T	37.04	23.48	41.55	53.10	34.37	30.90	45.23	38.06
	A+V	**59.20**	**40.52**	**53.71**	**71.26**	**51.17**	**47.04**	**47.88**	**47.46**
Qwen3.5-Omni-Plus	T	14.97	6.51	64.43	29.76	23.59	8.46	**48.85**	28.65
	V+T	13.70	6.64	66.98	21.71	22.43	23.77	41.39	32.58
	A+V	**20.79**	**11.86**	**75.46**	**32.28**	**29.56**	**49.99**	48.29	**49.14**

Table 8: Full controlled input-modality results for Track C on the 918 instances. Input abbreviations are T, V+T, and A+V. Scores are percentages; bold marks each model's best condition.

Model	Input	Grd.	Edit	No dmg.	Rev.	EFS
Gemini 3.5 Flash	T	15.06	7.18	39.04	25.45	18.00
	V+T	41.59	24.71	41.02	51.23	34.96
	A+V	**59.16**	**39.58**	**51.42**	**68.92**	**49.77**
Qwen3.5-Omni-Plus	T	30.39	18.81	47.97	45.17	31.07
	V+T	33.56	19.48	40.31	50.53	31.26
	A+V	**61.39**	**41.72**	**53.06**	**67.36**	**51.08**

Native audiovisual input is strongest for both tested Omni models. It has the highest score in all three tracks among the tested input conditions, supporting the value of preserving synchronized speech and visual context.

Temporal alignment. To isolate temporal alignment from modality content, we sample 300 benchmark instances without using model predictions, scores, or judge decisions. The sample spans all 129 source pages, both languages, all three difficulty bands, all 12 operations, and 4,554 reference steps. For each instance, we circularly shift the audio by five seconds while leaving the video stream unchanged; 150 use a -5-second shift and 150 use a $+5$-second shift, preserving all speech content. We evaluate Gemini 3.5 Flash and Qwen3.5-Omni-Plus on the aligned and shifted recordings with the same Track B prompt and judge. Confidence intervals use 10,000 paired source-page cluster bootstrap replicates.

Table 9: Track B temporal-alignment intervention on 300 instances. Δ is aligned minus shifted IRS; intervals are paired source-page cluster bootstrap 95% CIs.

Model	Aligned	Shifted	Δ	95% CI
Gemini 3.5 Flash	47.65	35.99	11.65	[9.37, 13.92]
Qwen3.5-Omni-Plus	48.39	34.66	13.73	[11.54, 16.00]

Temporal misalignment substantially reduces instruction recovery for both models. IRS drops by 11.65 points for Gemini 3.5 Flash and 13.73 points for Qwen3.5-Omni-Plus. Target F1 falls by 18.26 and 20.81 points, respectively, compared with Action F1 decreases of 5.04 and 6.66 points, showing that disrupted timing primarily impairs grounding the spoken request to the referenced page region.

Timestamped transcripts. We further evaluate whether explicit temporal information improves VL instruction recovery on the same fixed 300-instance set. We detect speech intervals from the isolated audio, use Qwen3-ASR-Flash (Shi et al., 2026) to audit the recognized content in each interval, and align the manually annotated step-level transcript with the corresponding intervals. The resulting input preserves the manual transcript verbatim and only prefixes each instruction with its start and end times, such as `[t=15s--23s]`. Both conditions use the same 1 FPS video frames, transcript text, Track B prompt, and scoring protocol.

Table 10: Track B results with plain and timestamped transcripts on the fixed 300-instance subset (300 cases for GPT-5.6 Sol and 298 completed cases for Grok 4.6). Plain+F denotes the plain manual transcript with 1 FPS frames; TS+F denotes the timestamped manual transcript with the same frames. Gain is TS+F minus Plain+F.

Model	Input	Target F1	Action F1	IRS
GPT-5.6 Sol	Plain+F	48.29	49.49	48.89
	TS+F	**59.31**	**55.99**	**57.65**
	Gain	**+11.02**	**+6.50**	**+8.76**
Grok 4.6	Plain+F	33.23	47.71	40.47
	TS+F	**56.68**	**54.23**	**55.45**
	Gain	**+23.45**	**+6.52**	**+14.98**

Explicit temporal alignment primarily improves target grounding for VL models. For GPT-5.6 Sol (OpenAI, 2026b), timestamps raise IRS by 8.76 points, with a larger gain in Target F1 (+11.02) than Action F1 (+6.50); the paired source-page cluster bootstrap gives a 95% confidence interval of [6.55, 11.04] for the IRS gain. Together with the degradation under shifted audio, this result shows that temporal correspondence between speech and page state is a central task signal, beyond the speech content alone.

The main leaderboard measures end-to-end performance under native model interfaces. We do not provide timestamp augmentation to VL models in the main results because constructing it requires separate audio processing and an external ASR system, whose capability is not native to the

Table 11: Execution control on the same 300 instances. Recovered and Oracle report the EFS of the final HTML generated from model-recovered and benchmark-reference instructions, respectively. Gap is Oracle minus Recovered; Ratio is Recovered divided by Oracle.

Instruction model	Executor	Recovered	Oracle	Gap	Ratio
Gemini 3.5 Flash	DeepSeek-v4-pro	49.32	89.35	40.03	55.20%
	Gemini 3.5 Flash	53.48	90.45	36.96	59.13%
Qwen3.5-Omni-Plus	DeepSeek-v4-pro	50.32	89.35	39.03	56.32%
	Qwen3.5-Omni-Plus	45.20	67.43	22.24	67.02%

evaluated VL model. Adding this information instead evaluates an augmented alignment-plus-VL pipeline. The main results therefore use the plain manual transcript with visual input, while this controlled experiment shows that an external temporal-alignment module can substantially improve VL instruction recovery.

A.4 Same-Model Execution Control

Track C uses a fixed DeepSeek-v4-pro code executor, so its recovered-to-oracle gap may depend on the executor choice. To test whether the gap is solely caused by the external executor, we reuse the fixed 300-instance sample from Appendix A.3 and evaluate Gemini 3.5 Flash and Qwen3.5-Omni-Plus with same-model execution.

For each instruction-recovery model, we compare two instruction sources and two executors. The first source is the editing instructions recovered by the model from the original audiovisual recording; the second is the benchmark's human-authored oracle instructions. Each instruction set is executed both by the fixed DeepSeek-v4-pro executor and by the instruction-recovery model itself, using the same source HTML, executor prompt, output budget, and EFS protocol within each comparison. The reported EFS therefore evaluates the final HTML generated after executing the corresponding instructions, rather than the instruction text itself. Missing or unusable outputs follow the main zero-fill policy.

Executor choice alone does not explain the recovered-to-oracle gap. For Gemini 3.5 Flash, self-execution raises recovered-instruction EFS from 49.32 to 53.48 and oracle-instruction EFS from 89.35 to 90.45, leaving a 36.96-point gap. For Qwen3.5-Omni-Plus, the scores for recovered and oracle instructions are 50.32 and 89.35 with DeepSeek-v4-pro, and 45.20 and 67.43 with self-execution, leaving gaps of 39.03 and 22.24 points. Substantial gaps persist under both executors, but this control does not by itself attribute the remaining gap to understanding or execution.

A.5 Alternative-Judge Ablation

The alternative-judge analysis uses a fixed 300-instance evaluator-validation set, with 100 instances from each visual-dependence band under seed 20260824 (Zheng et al., 2023; Panickssery et al., 2024). Each judge evaluates the same 300-instance grid and 4,482 reference steps under the zero-fill policy. The selected judge is used throughout both the rubric and No-damage stages; the experiment does not replace only one component of the scoring pipeline. Table 12 reports all component scores underlying the summary in Table 4.

Table 13: Blind human calibration of the automatic judge. Agreement and Cohen's κ (Cohen, 1960) are computed over binary item decisions; Pearson r and bias are computed over output-level component scores. Bias is automatic minus human score in percentage points.

Track	Component	Items	Agreement	κ	Pearson r	Bias
A	Target grounding	650	75.69%	0.502	0.867	−1.00
A	Edit fulfillment	528	76.52%	0.459	0.840	−10.65
A	Revision	122	68.85%	0.337	0.361	0.00
B	Target-F1	798	81.58%	0.630	0.888	−10.16
B	Action-F1	798	80.14%	0.606	0.851	−8.60
B	IRS	—	—	—	**0.906**	−9.38

Table 12: Full Track A judge-ablation results on the same 300 instances. Scores are percentages; parentheses give each model's rank under that judge for the corresponding column.

Model	Judge	Grd.	Edit	No dmg.	Rev.	EFS
Gemini 3.5 Flash	Gemini 3.1 Pro	62.78 (2)	**36.58 (1)**	60.95 (4)	**73.39 (1)**	**51.44 (1)**
	GPT-5.5	56.94 (2)	32.67 (2)	67.00 (4)	76.44 (2)	**50.72 (1)**
	DeepSeek-v4-pro	52.01 (2)	33.88 (2)	65.43 (4)	**70.11 (1)**	**49.29 (1)**
GPT-5.6 Sol	Gemini 3.1 Pro	**64.03 (1)**	36.21 (2)	32.95 (6)	70.89 (2)	45.28 (2)
	GPT-5.5	**57.61 (1)**	**32.77 (1)**	40.88 (6)	69.39 (6)	44.20 (3)
	DeepSeek-v4-pro	**54.83 (1)**	**34.06 (1)**	38.33 (6)	67.61 (2)	43.73 (2)
Kimi-K3	Gemini 3.1 Pro	50.99 (3)	30.44 (3)	53.87 (5)	62.78 (3)	43.65 (3)
	GPT-5.5	48.33 (3)	27.39 (3)	58.53 (5)	71.06 (5)	44.44 (2)
	DeepSeek-v4-pro	44.30 (3)	28.48 (3)	60.12 (5)	62.78 (3)	43.38 (3)
Gemini 3.1 Pro	Gemini 3.1 Pro	28.78 (4)	15.47 (4)	**76.92 (1)**	42.83 (4)	34.56 (4)
	GPT-5.5	30.62 (4)	14.42 (4)	**79.11 (1)**	**77.94 (1)**	41.68 (4)
	DeepSeek-v4-pro	23.53 (4)	15.02 (4)	**79.07 (1)**	55.44 (4)	36.81 (4)
Qwen3.5-Omni-Plus	Gemini 3.1 Pro	27.62 (5)	14.04 (5)	74.77 (2)	38.06 (5)	32.35 (5)
	GPT-5.5	26.33 (5)	13.05 (5)	77.47 (2)	73.28 (4)	39.31 (5)
	DeepSeek-v4-pro	21.59 (5)	13.51 (5)	78.84 (2)	46.56 (5)	34.01 (5)
Nemotron 3 Nano	Gemini 3.1 Pro	1.06 (6)	0.00 (6)	72.10 (3)	9.44 (6)	16.42 (6)
	GPT-5.5	6.61 (6)	0.42 (6)	74.08 (3)	75.22 (3)	30.73 (6)
	DeepSeek-v4-pro	1.52 (6)	0.45 (6)	73.78 (3)	29.83 (6)	21.10 (6)

A.6 Human Calibration of the LLM Judge

We constructed a paired human-calibration pool from 50 benchmark instances, with one Track A and one Track B output per instance. Sampling used fixed seeds and did not use model scores, rankings, or judge decisions. The pool balances four model families: in the order Claude Opus 5 (Anthropic, 2026), Gemini 3.5 Flash (Google DeepMind, 2026b), Qwen3.5-Omni-Plus (Team, 2026), and Qwen3.8-27B (Qwen Team, 2026), the Track A allocation is 12/13/12/13 and the Track B allocation is 13/13/12/12. Its 50 instances come from 50 distinct source pages and comprise 24 English and 26 Chinese examples, with 12 Easy, 12 Medium, and 26 Hard. All 12 operations are represented: eight common types occur in all 50 source histories, while Hide/Delete, Change Color, Rewrite Text, and Change Font occur in 49, 37, 15, and 3, respectively. The only availability requirement was that the assigned model output had already been generated and could be inspected.

The human evaluator inspected the source page, request, and model output, judging target grounding and edit or revision fulfillment for Track A, and target/action recovery plus step alignment for Track B. Automatic decisions remained hidden until the corresponding human decisions were submitted, and all analyses use the first answers frozen before judge reveal. The evaluator completed all the outputs, covering 1,451 steps and 2,902 valid binary decisions: 653 steps for Track A and 798 for Track B. This calibration targets the rubric components decided by the LLM judge; the deterministic No-damage component is outside its scope.

Human and automatic judgments are broadly aligned. On Track A, output-level correlations reach $r = 0.867$ for grounding and $r = 0.840$ for edit fulfillment. Agreement is higher on Track B: Target

and Action exceed 80% item agreement, step alignment agrees in 91.39% of cases, and IRS reaches $r = 0.906$. The judge is more conservative on edit fulfillment and Track B, but preserves relative variation across outputs. Together with the alternative-judge results, this human check supports using the automatic evaluator as a consistent and reproducible basis for model comparison.

A.7 EFS Hyperparameter Sensitivity

We evaluate whether the EFS rankings depend on its category weights or the No-damage normalization coefficient α. The sweep varies $\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$ and uses 26 weight vectors that sum to one, including a balanced vector and configurations in which Edit Fulfillment has the largest or tied-largest weight. Their Cartesian product gives 156 configurations; the baseline is $(w_G, w_E, w_N, w_R) = (0.10, 0.50, 0.20, 0.20)$ with $\alpha = 0.15$. All configurations reuse the same predictions and judgments and retain the primary zero-fill policy.

Table 14: Ranking sensitivity to EFS weights and the No-damage coefficient α. Spearman ρ compares each configuration with the baseline ranking.

Track	Variation	Configs	Leader stable	Median ρ	Min. ρ	Max. rank shift
A	Weights only	26	26/26	0.982	0.743	6
A	α only	6	6/6	0.993	0.986	2
A	Joint	156	156/156	0.977	0.639	8
C	Weights only	26	26/26	0.998	0.988	2
C	α only	6	6/6	0.999	0.995	1
C	Joint	156	156/156	0.998	0.971	3

The top-ranked models are stable across the tested configurations. Gemini 3.5 Flash remains first on Track A, while Qwen3.5-Omni-Plus remains first on Track C. Varying only α has little effect: the minimum Spearman correlation is 0.986 on Track A and 0.995 on Track C. Rank changes beyond the leader occur more often, so we treat the top model as stable within this sweep without claiming that the full ranking is invariant.

Weight sensitivity is concentrated in Track A's middle and lower ranks. Track C remains highly stable under weight-only changes, with a minimum Spearman correlation of 0.988. On Track A, the No-damage-heavy vector $(0.10, 0.40, 0.40, 0.10)$ lowers ρ to 0.743 and moves Qwen3.8-27B from rank 13 to rank 7. Combining this vector with $\alpha = 0.05$ gives the joint minimum $\rho = 0.639$; nevertheless, 150 of 156 joint configurations retain $\rho \geq 0.9$, and none changes the leader. Thus, the top-ranked models are stable across the tested EFS parameterizations, while Track A's full ordering reflects the metric's tradeoff between preservation and requested edits.

A.8 Track B Metric Sensitivity

The primary IRS averages target and action F1, so unsupported predicted steps reduce precision without changing recall. For component $c \in \{T, A\}$, let

$$P_i^c = \frac{m_i^c}{n_i^{\text{pred}}}, \qquad R_i^c = \frac{m_i^c}{n_i^{\text{ref}}}, \tag{10}$$

$$F_{\beta,i}^c = \frac{(1+\beta^2) P_i^c R_i^c}{\beta^2 P_i^c + R_i^c}, \tag{11}$$

$$\text{IRS}_\beta = \frac{1}{2N} \sum_{i=1}^{N} \left(F_{\beta,i}^T + F_{\beta,i}^A \right). \tag{12}$$

Here $N = 918$. We set $P_i^c = 0$ when no steps are predicted and $F_{\beta,i}^c = 0$ whenever $P_i^c R_i^c = 0$. The primary metric uses $\beta = 1$; $F_{0.5}$ emphasizes precision and therefore penalizes unsupported predictions more strongly, whereas F_2 emphasizes recall. We compare these variants with recall-only scoring and a stricter Joint-F1 that credits a step only when both its target and action are correct.

Table 15: Track B ranking sensitivity. Correlations and rank shifts are measured against the primary Component-F1 IRS ranking.

Metric	Leader	ρ	Max shift	Same top 3
Component F1 (primary)	Qwen3.5-Omni-Plus	1.000	0	Yes
Recall only	Qwen3.5-Omni-Plus	0.998	1	Yes
Component $F_{0.5}$	Gemini 3.5 Flash	0.983	3	No
Component F_2	Qwen3.5-Omni-Plus	0.998	1	Yes
Joint F1	Claude Opus 5	0.983	2	Yes

The broad ranking is stable, but the exact leader depends on how precision is weighted. Every alternative has Spearman $\rho \geq 0.983$ relative to the primary ranking. Qwen3.5-Omni-Plus remains first under recall-only and F_2, whereas $F_{0.5}$ favors the more concise outputs of Gemini 3.5 Flash and Joint-F1 places Claude Opus 5 first. The primary Component-F1 definition balances recovered coverage with unsupported-output penalties without requiring target and action to pass jointly.

A.9 Source-Page Cluster Bootstrap

The 918 instances derive from 129 source webpages, and instances from the same page reuse its layout, components, and visual style. They may therefore be correlated rather than independent observations. Treating all instances as independent could overstate the effective sample size and make model differences appear more stable than they are across distinct webpages. We use a source-page cluster bootstrap (Field & Welsh, 2007) to test whether track leaders and the Track A-to-C changes persist across alternative compositions of source webpages, rather than being driven by a small number of pages.

Each replicate samples 129 page IDs with replacement. A page drawn k times contributes all of its instances k times, while unselected pages contribute none. The same draw is used for every model and track, preserving paired comparisons. Across 10,000 replicates, we recompute instance-macro EFS or IRS, ranks, pairwise gaps, and C–A differences. In Table 16, p_+ is the fraction of positive differences; "Stable $+$" or "Stable $-$" requires at least 97.5% in the corresponding direction.

Table 16: Selected source-page cluster bootstrap comparisons over 10,000 resamples. Δ is the first score minus the second, or Track C minus Track A; p_+ is the percentage of positive resamples.

Comparison	Δ	p_+	Direction
Track A: Gemini 3.5 Flash - Claude Opus 5	+3.80	99.99%	Stable +
Track B: Qwen3.5-Omni-Plus - Claude Opus 5	+0.23	60.30%	Uncertain
Track C: Qwen3.5-Omni-Plus - Gemini 3.5 Flash	+1.31	98.63%	Stable +
A→C: Seed2.0 Lite	+19.09	100.00%	Stable +
A→C: Gemini 3.5 Flash	-1.40	1.09%	Stable -
A→C: Qwen3.5-Omni-Plus	+21.52	100.00%	Stable +
A→C: Grok 4.6	-2.85	0.00%	Stable -

The Direct Editing leader is stable, whereas the leading Track B pair remains close. Gemini 3.5 Flash ranks first on Track A in 99.99% of resamples. Qwen3.5-Omni-Plus ranks first on Track B in 55.67% of resamples, and its 0.23-point lead over Claude Opus 5 has uncertain direction. It retains the highest Track C EFS in 98.63% of resamples; its 1.31-point lead over Gemini 3.5 Flash is positive in 98.63% of the paired resamples.

The pipeline benefit is model dependent. Of the 15 models with both Track A and Track C results, six have positive C–A differences in at least 97.5% of resamples and two have a negative difference at the same threshold; the remaining seven have uncertain direction. The large gains for Qwen3.5-Omni-Plus and Seed2.0 Lite remain positive in every resample, whereas Grok 4.6 consistently favors direct editing. As a separate sensitivity check, averaging pages equally preserves the Track A, B, and C leaders, with Spearman correlations of 0.982, 1.000, and 0.995 against the instance-macro rankings.

A.10 Coverage and Zero-Fill Robustness

Zero-filled failures do not determine the leading models. The primary aggregation retains all 918 instances and assigns zero to refused, empty, unparsable, or otherwise unusable outputs. When we recompute each model's score over valid outputs only, all three tracks retain their leaders and the Track B top-three set remains unchanged. No available model–track contains more than 41 zero-filled instances. Table 18 reports the complete per-model coverage.

Table 17: Robustness to zero-filled unusable outputs on the 918 instances. Valid-only scores omit non-valid outputs.

Track	Models	Affected	Zero-filled	ρ	$\Delta_{\max}$
A	15	6	133	0.989	2.21
B	17	8	87	0.998	1.30
C	17	8	87	0.998	1.27

Table 18: Per-model output coverage. Entries are valid / zero-filled outputs out of 918; "–" denotes unavailable Track A runs.

Model	Track A	Track B	Track C
MiniCPM-o 4.5	–	918 / 0	918 / 0
Gemma 4 12B	–	913 / 5	913 / 5
Qwen3.8-27B	918 / 0	917 / 1	917 / 1
Nemotron 3 Nano	918 / 0	918 / 0	918 / 0
MiMo-V2.5	893 / 25	915 / 3	915 / 3
MiniMax-M3	881 / 37	879 / 39	879 / 39
Qwen3.8-Max	918 / 0	918 / 0	918 / 0
Kimi-K3	915 / 3	902 / 16	902 / 16
Muse Spark 1.1	918 / 0	918 / 0	918 / 0
Seed2.0 Lite	910 / 8	911 / 7	911 / 7
Gemini 3.5 Flash	918 / 0	918 / 0	918 / 0
Qwen3.5-Omni-Plus	918 / 0	918 / 0	918 / 0
Seed2.1 Pro	918 / 0	918 / 0	918 / 0
Grok 4.6	899 / 19	906 / 12	907 / 11
Gemini 3.1 Pro	918 / 0	918 / 0	918 / 0
GPT-5.6 Sol	918 / 0	918 / 0	918 / 0
Claude Opus 5	877 / 41	914 / 4	913 / 5

A.11 PROMPT DOCUMENTATION

This appendix documents the model-facing prompts used by the evaluation pipeline. Each box contains only text supplied to the model; titles, delivery information, and other documentation remain outside the boxes. Bracketed uppercase text inside a box denotes content substituted at runtime. The boxes reproduce the original prompts used for the reported evaluation.

A.11.1 CONTESTANT AND EXECUTOR PROMPTS

Track A uses three retained input variants. The `omni` message accompanies an audio-bearing video; `vl_asr` accompanies a silent video and includes its transcript; and `vl_asr_frames` accompanies sampled frames and includes the transcript.

Track A: `omni` user-message text

````
You are a web page editing model. You are given the original HTML of a
web page and one web page editing request from the user. Understand
what the user actually wants to change, and edit the page accordingly.

Requirements:
1. Output the complete HTML.
2. Make only the change the user requested; do not make extra changes,
and do not explain.
3. Keep the rest of the page's content and structure as unchanged as
possible.
4. If the request contains multiple steps, an undo, or a change of
mind, follow the final intent.
5. Return one ```html code block whose content is the complete page.

The user's editing request is in the attached video: the user points
at parts of the page with a cursor or a finger while speaking the
request out loud, using deictic words such as "this", "here", "these
two". Watch the picture to see WHICH part of the page is being pointed
at, and listen to the speech to hear WHAT to change | you need both,
together with the HTML below, to resolve each reference to a concrete
element.

Original HTML:
```html
[ORIGINAL_HTML]
```
````

Track A: `vl_asr` user-message text

````
You are a web page editing model. You are given the original HTML of a
web page and one web page editing request from the user. Understand
what the user actually wants to change, and edit the page accordingly.

Requirements:
1. Output the complete HTML.
2. Make only the change the user requested; do not make extra changes,
and do not explain.
3. Keep the rest of the page's content and structure as unchanged as
possible.
4. If the request contains multiple steps, an undo, or a change of
mind, follow the final intent.
5. Return one ```html code block whose content is the complete page.
````

````
The user's editing request is split across two inputs. (1) The
attached video is SILENT and only shows the user pointing at parts of
the page with a cursor or a finger. (2) The text below is a
transcription of what the user said at the same time, one line per
step, in the order requested. The text tells you WHAT to change; the
video tells you WHICH part of the page each step points at. Use both.
Transcription of the user's speech:
[ASR_TRANSCRIPT]

Original HTML:
```html
[ORIGINAL_HTML]
```
````

Track A: `vl_asr_frames` user-message text

````
You are a web page editing model. You are given the original HTML of a
web page and one web page editing request from the user. Understand
what the user actually wants to change, and edit the page accordingly.

Requirements:
1. Output the complete HTML.
2. Make only the change the user requested; do not make extra changes,
and do not explain.
3. Keep the rest of the page's content and structure as unchanged as
possible.
4. If the request contains multiple steps, an undo, or a change of
mind, follow the final intent.
5. Return one ```html code block whose content is the complete page.

The user's editing request is split across two inputs. (1) The
attached [N_IMAGES] image(s) are not separate screenshots: they are
consecutive frames sampled once per second from ONE silent screen
recording of this page, in which the user points at parts of the page
with a cursor. The images are in chronological order | one image per
second of the recording | and every image has its true timestamp
burned into its top-left corner; read those timestamps off the image
rather than computing them. (2) The text below is a transcription of
what the user said while that recording was made, one line per step,
in the order requested. The text tells you WHAT to change; the images
tell you WHICH part of the page each step points at. Use both.
Transcription of the user's speech:
[ASR_TRANSCRIPT]

Original HTML:
```html
[ORIGINAL_HTML]
```
````

Track B likewise uses an audio-bearing video for `omni`, a silent video plus transcript for `vl_asr`, and sampled frames plus transcript for `vl_asr_frames`. No source HTML is supplied in the main Track B protocol.

Track B: `omni` user-message text

```
You are given a video in which a user points at parts of a web page
and speaks editing requests using deictic references ("this", "here",
"these two"). You are NOT asked to edit the HTML. Rewrite the user's
request into explicit, self-contained instructions.

Output ONLY a JSON array, one object per editing step, in the order
the user requested them:
[{"step": 1, "target": "...", "instruction": "..."}, ...]

- "target": which part of the page the user is pointing at, described
precisely enough that someone who never saw the video could find it on
the page | quote the visible text, heading, label or section name you
can read on screen, plus its position if needed. Never use "this" /
"here" / "that".
- "instruction": what to change, as one explicit imperative sentence.
No deictic words. If the user undoes a step or changes their mind,
state the FINAL intent.
- Write "target" and "instruction" in THE SAME LANGUAGE THE USER
SPEAKS in the video. Do not translate the user's words into English.
The only exception is text you read off the page itself (headings,
labels, button text) | quote that verbatim in whatever language it
appears on screen.
- Emit exactly as many steps as the user actually requested. Do not
pad or merge.
- Do not output HTML.
```

Track B: `vl_asr` user-message text

```
You are given a SILENT screen recording of a web page and a
transcription of what the user said while it was recorded.

In the recording the user moves the mouse cursor to point at things on
the page. In the transcription the user refers to those things only by
weak deictic words ("this", "here", "these two") | which cannot be
resolved from the text alone. Resolve every one of them by aligning
each spoken step with where the cursor is at that moment in the
recording.

You are NOT asked to edit the HTML. Rewrite the user's request into
explicit, self-contained instructions.

Output ONLY a JSON array, one object per editing step, in the order
the user requested them:
[{"step": 1, "target": "...", "instruction": "..."}, ...]

- "target": which part of the page the user is pointing at, described
precisely enough that someone who never saw the video could find it on
the page | quote the visible text, heading, label or section name you
can read on screen, plus its position if needed. Never use "this" /
"here" / "that".
- "instruction": what to change, as one explicit imperative sentence.
No deictic words. If the user undoes a step or changes their mind,
state the FINAL intent.
```

```
- Write "target" and "instruction" in THE SAME LANGUAGE AS THE
TRANSCRIPTION below. Do not translate the user's words into English.
The only exception is text you read off the page itself (headings,
labels, button text) | quote that verbatim in whatever language it
appears on screen.
- Emit exactly as many steps as the user actually requested. Do not
pad or merge.
- Do not output HTML.

The recording is attached as a silent video file.

Transcription of the user's speech (one line per step, in the order
requested):
[ASR_TRANSCRIPT]
```

Track B: `vl_asr_frames` user-message text

```
You are given a SILENT screen recording of a web page and a
transcription of what the user said while it was recorded.

In the recording the user moves the mouse cursor to point at things on
the page. In the transcription the user refers to those things only by
weak deictic words ("this", "here", "these two") | which cannot be
resolved from the text alone. Resolve every one of them by aligning
each spoken step with where the cursor is at that moment in the
recording.

You are NOT asked to edit the HTML. Rewrite the user's request into
explicit, self-contained instructions.

Output ONLY a JSON array, one object per editing step, in the order
the user requested them:
[{"step": 1, "target": "...", "instruction": "..."}, ...]

- "target": which part of the page the user is pointing at, described
precisely enough that someone who never saw the video could find it on
the page | quote the visible text, heading, label or section name you
can read on screen, plus its position if needed. Never use "this" /
"here" / "that".
- "instruction": what to change, as one explicit imperative sentence.
No deictic words. If the user undoes a step or changes their mind,
state the FINAL intent.
- Write "target" and "instruction" in THE SAME LANGUAGE AS THE
TRANSCRIPTION below. Do not translate the user's words into English.
The only exception is text you read off the page itself (headings,
labels, button text) | quote that verbatim in whatever language it
appears on screen.
- Emit exactly as many steps as the user actually requested. Do not
pad or merge.
- Do not output HTML.

The recording is attached as [N_IMAGES] consecutive frames sampled
once per second from that ONE recording | not as separate screenshots.
They are in chronological order, one image per second, and every image
has its true timestamp burned into its top-left corner; read those
timestamps off the image rather than computing them.
```

```
Transcription of the user's speech (one line per step, in the order
requested):
[ASR_TRANSCRIPT]
```

Track C Fixed-Executor Prompt

```
You are a web page editing model. You are given the original HTML of a
web page and one web page editing request from the user. Understand
what the user actually wants to change, and edit the page accordingly.

Requirements:
1. Output the complete HTML.
2. Make only the change the user requested; do not make extra changes,
and do not explain.
3. Keep the rest of the page's content and structure as unchanged as
possible.
4. If the request contains multiple steps, an undo, or a change of
mind, follow the final intent.
5. Return one ```html code block whose content is the complete page.

The user's web page editing request:
[STRUCTURED_CHANGE_LIST]

Original HTML:
```html
[ORIGINAL_HTML]
```
```

A.11.2 Scoring Prompts

The following no-video judge prompt begins with the complete ordered history, marking each step as ACTIVE or CANCELLED, followed by the confirmed whole-step revision links. Each history line has the form [step k] [status] instruction; cancelled requests provide context only, and the rubric block contains only active items.

Tracks A and C Rubric-Judge Prompt

```
FINAL-STATE EVALUATION: Only the final HTML is observable. Score only
the supplied active rubric items.
No intermediate execution trace is required. A correct restored state
may have no diff from the original.
Cancelled requests below are historical context, NOT requirements or
permissions for the final page.
Restoration references apply only to the affected target/properties,
never to the whole page.
HISTORY (reference only):
[STEP_STATUS_HISTORY]
REVISION LINKS: [CONFIRMED_WHOLE_STEP_LINKS]

You are an expert evaluator of web page editing.

Your task is to judge, item by item, whether the contestant model's
modification of the web page is correct, according to the **rubric**
below.
The rubric already contains, for every editing step, the target
element, the expected result of the edit and a detailed description of
the changes that are forbidden.

How to judge:
- Carefully compare the code diff between [ORIGINAL HTML] and
[MODIFIED HTML]
- Locate the target element via before_anchor in the rubric
```

````
- Check whether the modification described by after_expected_change is
present in the code
- Check whether the problem described by forbidden_change occurred

Each rubric item contains:
- item_id: rubric item id
- category: category (target_grounding = locating the target,
edit_fulfillment = carrying out the edit, no_damage = no collateral
damage, revision = undo / change of mind)
- question: the question you must answer
- before_anchor: features of the element before the edit (to help you
locate it)
- after_expected_change: the expected result of the edit
- forbidden_change: changes that must not happen

Output in strict JSON format (do not include any other text):

```json
{
  "items": [
    {"item_id": "r1", "category": "target_grounding", "met": true,
    "reason": "brief justification"}
  ]
}
```

Requirements:
- You must give a judgment for every item in the rubric; none may be
omitted
- met=true means the item passes, met=false means it does not pass
- reason must cite the concrete HTML code change as evidence

[RUBRIC]
[RUBRIC_JSON]

[ORIGINAL HTML]
```html
[ORIGINAL_HTML]
```

[MODIFIED HTML]
```html
[MODIFIED_HTML]
```
````

Track B Matching-Judge Prompt

```
You are an expert evaluator of *instruction rewriting* for web page
editing.

A user recorded a video in which they point at parts of a web page and
speak editing requests
using deictic references ("this", "here", "these two"). A contestant
model watched that video
and rewrote the request into explicit, self-contained editing steps.
Your job is to score that
rewrite against the reference answer.

You are given:
```

```
- [REFERENCE STEPS]: the ground-truth steps, written by a human
annotator from the same video.
  Each has a `target` (which part of the page) and an `instruction`
  (what to change).
- [CONTESTANT STEPS]: what the contestant model produced. The step
numbering may differ, steps
  may be missing, merged, split, extra, or out of order.
- [SOURCE HTML]: the page before any edit, so you can check whether a
target description really
  points at the element the reference means.

Do this, in order:

1. **Match** each reference step to at most one contestant step. Match
by *what is being done to
   what*, not by step number and not by wording. A contestant step may
   be used for at most one
   reference step. If no contestant step plausibly corresponds to a
   reference step, leave it
   unmatched (`matched_contestant_step: null`) and score both criteria
   0.

2. For each matched pair, judge two criteria independently, each 0 or
1:

   - `grounding_met`: does the contestant's `target` identify the
   **same element or region** of
     the page as the reference's `target`? Use [SOURCE HTML] to
     resolve both descriptions to
     actual elements. Score 1 when they resolve to the same
     element/region even if worded
     completely differently (a quoted visible string, a heading name,
     a position, a CSS-ish
     description are all acceptable ways to say it). Score 0 when it
     resolves to a different
     element, when it is so vague that it could be many elements, or
     when it still contains a
     deictic reference ("this one", "here") that carries the location.
   - `instruction_met`: if a competent web developer followed the
   contestant's `instruction`
     on that element, would the result be the change the reference's
     `instruction` describes?
     Score 1 when the *outcome* matches. Score 0 when the outcome
     differs, when the instruction
     is too vague to act on, or when it still relies on the video to
     be understood.

**Do not deduct for wording, style, verbosity, language, or
ordering.** Only the identified
element and the resulting change matter. Being more specific than the
reference is fine.
When the reference `instruction` names a value (a color, a count, a
position) the contestant
must reach the same value; a different value is 0.

Output ONLY this JSON, no other text:

```json
{
  "steps": [
```
```

```
    {"gt_step": 1, "matched_contestant_step": 2, "grounding_met": 1,
    "instruction_met": 0,
     "reason": "brief justification, quote the deciding words from
     both sides"}
  ],
  "extra_contestant_steps": [4],
  "notes": "optional, one short line"
}
```

Requirements:
- Emit exactly one entry per reference step, in reference order. Never omit one.
- `extra_contestant_steps`: step numbers of contestant steps you matched to nothing.
- `reason` must cite the concrete wording that decided the call.

[REFERENCE STEPS]
[REFERENCE_STEPS]

[CONTESTANT STEPS]
[CONTESTANT_STEPS]

[SOURCE HTML]
```html
[ORIGINAL_HTML]
```

No-Damage Judge Prompt

```
You are an expert at detecting web page integrity problems. I will
give you a list of changes a web page underwent between before and
after an edit, plus a description of the editing task.
Decide, for each change, whether it is a "reasonable result of the
editing task" or "unintended collateral damage".

[EDITING TASK DESCRIPTION]
[EDIT_TASK_DESCRIPTION]

[CHANGE LIST]
[STRUCTURED_CHANGE_LIST]

Output in strict JSON format (do not include any other text):

```json
{
  "changes": [
    {"id": 1, "is_damage": false, "severity": null, "reason": "brief
    reason"},
    {"id": 2, "is_damage": true, "severity": "high", "reason": "brief
    reason"}
  ],
  "damage_count": 0,
  "overall_assessment": "one-sentence summary"
}
```

severity levels:
```

```
- "high": content loss (text disappeared, image deleted) or structural
breakage (a section is gone)
- "medium": style / attribute anomaly (color changed outside the
target area, a link was modified)
- "low": tiny change (slight style adjustment that does not hurt
readability)

Judging principles:
- If the change is explicitly requested by the editing task
description, or is a reasonable side effect of that operation ->
is_damage=false
- If the change happens in an area completely unrelated to the editing
task -> is_damage=true
- If you are not sure -> lean towards is_damage=true (judge strictly)
- Adding/removing text inside the edited area is reasonable, but text
disappearing far away from the edited area is damage
```

The replacement-image judge receives one ordered multimodal user message: the text label `[ORIGINAL IMAGE]`, the original-image attachment, the text label `[NEW IMAGE]`, the new-image attachment, and then the instruction text below. If the original image cannot be retrieved, its label and attachment are both omitted.

Replacement-Image Judge: final text part

````
You are an expert at verifying image content. You will see two images
and one description.

[TASK] Decide:
1. Is the new image really different from the original image (not the
same image)?
2. Does the new image match the content description below?

[CONTENT DESCRIPTION]
[IMAGE_CONTENT_DESCRIPTION]

Output in strict JSON format (do not include any other text):

```json
{
  "is_different": true/false,
  "matches_description": true/false,
  "image_content": "briefly state what the new image actually shows
  (under 20 words)",
  "reason": "your justification (under 30 words)"
}
```

Notes:
- If the new image is a solid color / gradient / abstract pattern
rather than a real photo, matches_description must be false
- If the description asks for a specific subject (e.g. "sculpture",
"architecture"), the new image must contain that subject to count as a
match
- Judge strictly: when in doubt, answer false
````

A.11.3 Difficulty-Probe Prompts

The following prompts support the benchmark difficulty analysis.

Text-Only Target-Recovery Prompt

```
You are a senior expert in web page structure analysis.

Below, you are given the source HTML of a web page and the spoken
editing instructions that a user issued step by step while facing this
page in a screen recording (ASR speech transcription, one sentence per
step, numbered in order). While speaking, the user points to a region
on the screen with the mouse or a gesture, but you cannot see the
video; you have only these textual instructions and the HTML.

Your task: Using only the HTML structure and instruction text, infer
which specific element or region of the page is denoted by the
demonstrative expressions in each instruction (such as "this", "here",
"these two", or "this region").

Requirements:
1. For each step, state the target element you infer and uniquely
describe it in concise natural language: what region of the page it is
(semantic name/function), the key visible text it contains, and its
approximate position on the page (which screen, top, left, right,
etc.). Do not paste HTML code, tag names, or class/id values; explain
in ordinary language which region it is, in no more than one or two
sentences.
2. You may use the context of the entire instruction sequence
(preceding and following steps) to assist your judgment.
3. If the target cannot be determined from the text and HTML alone,
mark the confidence as "low" and still provide your most likely guess
(do not leave it blank).

Source HTML:
<<<HTML
[ORIGINAL_HTML]
HTML>>>

Step-by-step instructions ([N_STEPS] steps in total):
[ASR_TRANSCRIPT]

Output only JSON in the following format (do not output any extra
text):
{"steps": [{"step": 1, "target": "a concise natural-language
description locating the target element", "confidence":
"high|medium|low"}, ...]}
```

Target-Equivalence Judge Prompt

```
You are checking whether the "target-element localization" for a web
page editing task is correct.

You are given three things:
- The user's spoken instruction for this step (containing a
demonstrative expression);
- [GROUND TRUTH], the target element actually indicated in this step
(human annotation);
- [MODEL LOCALIZATION], the target element inferred by a model using
only text and HTML.

Determine whether [MODEL LOCALIZATION] and [GROUND TRUTH] refer to the
same element or region on the web page.
```

```
Decision rule: Count it as a hit (hit=1) whenever the two semantically
refer to the same target, even if the wording or level of descriptive
detail differs. If they refer to different regions, the localization
is clearly incorrect, or the answer is irrelevant to the target, then
hit=0.

User instruction for this step: [ASR_TRANSCRIPT]
[GROUND TRUTH]: [GROUND_TRUTH_TARGET]
[MODEL LOCALIZATION]: [MODEL_LOCATED_TARGET]

Output only JSON: {"hit": 0 or 1, "reason": "briefly explain the
decision"}
```