SAME EXPLANATIONS, OPPOSITE EFFECTS: HOW DATA CONSTRUCTION SHAPES WEBPRM JUDGMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Web process reward models (WebPRMs) use explanations to help agents choose actions. When training labels influence how explanations are generated, models may learn to recognize labels rather than judge actions. We introduce PreAct, which reconstructs both candidates' explanations in WebPRM Collection using one generator and only pre-execution information, without access to preference labels. To evaluate how models use explanations, we introduce WebContrast, a benchmark of 300 pages across 119 websites that tests correct responses to goal changes and stability across explanation versions. On WebContrast, the same test explanations hurt action judgment on average after original-data training but help after PreAct training. Relative to each model's accuracy without explanations, their effect shifts from $-$**3.58 to +5.58** percentage points for Qwen3-8B and from $-$**24.25 to +5.33** for Llama-3.1-8B-Instruct. PreAct also improves mean action-selection accuracy on WebLINX and WebShop, and on Collection with pre-execution or no explanations. Yet evaluation with Collection's original explanations favors original-data training. Our findings show that data construction can change whether explanations help or hinder action judgment, while evaluation using the original explanations can obscure this difference.

1 INTRODUCTION

Web process reward models (WebPRMs) help web agents choose actions before their outcomes are known. Two prominent open-source families, Web-Shepherd (Chae et al., 2025) and WebArbiter (Zhang et al., 2026), evaluate candidate actions together with their *rationales*, explanations supplied to the reward model rather than its own reasoning. These explanations should improve choices for the current page and goal. High accuracy at selecting the labeled better action, however, does not demonstrate this benefit: it may also reflect label cues introduced during explanation generation. In the two families' shared training dataset, WebPRM Collection, actions labeled as better (*preferred*) receive explanations generated using post-execution screenshots, whereas alternatives (*rejected*) carry explanations from unexecuted proposals (Chae et al., 2025, Appendix B.3; Zhang et al., 2026, Section 3.2). This difference, illustrated in Figure 1, can make the label recognizable from how an explanation was produced, rather than from judging the action itself.

Our audit examines this risk. Classifiers reading only rationales predict preference labels with 97.70–99.94% accuracy on WebPRMBench and WebRewardBench, without the page, goal, or explicit action fields. Replacing original rationales with versions generated only from information available before execution (*pre-action information*) lowers accuracy across all six released WebPRMs we test by 21.67 percentage points on average, with tasks, pages, and actions unchanged. These findings establish label predictability and sensitivity to explanation construction, but not its training consequences. We therefore ask: *can changing explanation construction during training make the same test explanations more useful, and does evaluation with the original explanations recognize that improvement?*

We introduce *WebContrast* to measure whether explanations help models choose according to the user's goal. It contains 300 pages, each with two candidate actions and two goals requiring opposite choices. The page, actions, and rationales remain fixed across goals, so always favoring one candidate cannot answer both correctly. A complementary test changes only rationale versions while keeping the goal and correct action fixed. Because stability could arise from ignoring explanations,

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.

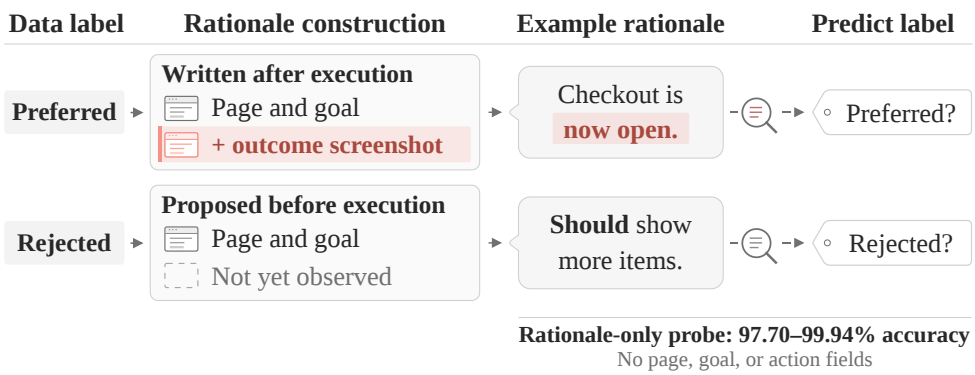


Figure 1: Label-dependent construction can leave recognizable traces in rationales. Phrases and predicted labels are illustrative; accuracy comes from full-rationale probes on WebPRMBench and WebRewardBench.

we also compare each model's rate of answering both goals correctly with and without them. Within each training experiment, all models receive identical test inputs, allowing us to examine how training changes the effect of the same explanations.

Using this evaluation, we study controlled training on Qwen3-8B and Llama-3.1-8B-Instruct. Each training action has an original rationale and one regenerated from pre-action information. We vary which version accompanies preferred actions, assigning the other to alternatives, while keeping states, actions, labels, and available texts fixed. Both models learn to favor the explanation version paired with preferred actions, and reversing the pairing reverses their preference. Under fixed pairings, the WebContrast explanations reduce mean accuracy. Mixing the pairings so that neither version consistently signals the label reduces this interference on both backbones. Changing the training association therefore affects how explanations influence task decisions, not just which version a model favors.

These findings motivate a construction principle: preference labels should supervise action judgment, not determine how candidate explanations are generated. We implement it through *PreAct*, which reconstructs both candidates' rationales using one generator, equal access to pre-action information, and no access to preference labels. The pages, actions, and training labels remain unchanged. On both backbones, the same WebContrast explanations lower mean accuracy after original-data training but raise it after PreAct training, relative to each model's own accuracy without explanations. The improvement mainly comes from fewer otherwise correct judgments being overturned. PreAct also improves mean action-selection accuracy on Collection with pre-action or no rationales, on WebLINX (Lù et al., 2024) without rationales, and on offline WebShop comparisons labeled using recorded branch rewards (Yao et al., 2022; Xia et al., 2025).

Finally, we ask whether evaluation with Collection's original explanations recognizes this improvement. It instead ranks original-data training higher on both backbones, favoring models that score higher with label-revealing rationales but achieve lower mean action-selection accuracy with pre-action or natural actor rationales, or without rationales. Higher original-rationale accuracy therefore need not indicate more effective explanation use. Our findings link how training explanations are constructed to how they affect later action judgment, and show that original-rationale accuracy can miss improvements in this behavior. PreAct addresses how training explanations are produced; WebContrast tests whether models benefit from explanations when choosing according to the user's goal. Model selection should therefore examine goal judgment and explanation benefit alongside original-rationale accuracy.

2 Related Work

Process rewards for web agents. Reward models guide action selection by evaluating candidate decisions at each step of a task. Web-Shepherd (Chae et al., 2025) checks individual candidates against a list of criteria and converts the feedback into scores, whereas WebArbiter (Zhang et al.,

2026) compares candidates directly using a set of evaluation principles. They also provide reusable data and benchmarks: Web-Shepherd introduced WebPRM Collection and WebRewardBench, while WebArbiter reuses Collection for training and contributes WebPRMBench. Related work includes SimiLAR (Miao et al., 2025), which evaluates agent steps along multiple dimensions, and AgentRM (Xia et al., 2025), which studies how reward models generalize across agent tasks. Our audit focuses on the rationales supplied with candidate actions in Web-Shepherd and WebArbiter. These are explanations read by the evaluator, not the evaluator's own reasons for its judgment. We examine whether their construction reveals preference labels and changes how models learn to judge actions.

Biases in reward models and evaluators. Model judgments can depend on how a response is written or who produced it, rather than its quality alone. RM-Bench (Liu et al., 2025) separates sensitivity to response quality from writing-style bias. Style Outweighs Substance (Feuer et al., 2025) finds that judges can favor presentation over factual correctness and other task requirements. Panickssery et al. (2024) link self-preference to evaluators' ability to recognize their own outputs. reWordBench (Wu et al., 2025) tests judgment stability under meaning- or preference-preserving transformations and improves it through consistency training. These studies establish that presentation and provenance can bias evaluation. We examine a further consequence in WebPRMs: training on differently constructed rationales can change whether the same explanations help or hinder subsequent action judgment, even when all models receive identical test text.

Data construction and behavioral diagnosis. Data collection itself can create shortcuts. In natural language inference, Gururangan et al. (2018) show that cues left by annotation allow labels to be predicted from just one sentence, even though the task requires comparing two. Data construction can also help address biases: CROME (Srivastava et al., 2025) uses targeted synthetic response pairs to train reward models to distinguish quality differences without relying on irrelevant features. PreAct focuses on candidate rationales, regenerating both from the same pre-action context while preserving states, actions, and preference labels. To evaluate such changes, an overall accuracy score can be complemented by tests of which input changes affect a model's choices. CheckList (Ribeiro et al., 2020) tests both stability under changes that should not matter and appropriate responses to changes that should affect predictions. WebContrast applies this principle to the consequences of rationale construction: choices should follow goal changes that require a different action but remain stable when only the rationale version changes. Testing both on the same pages shows whether a training choice improves task responses as well as rationale stability.

3 METHODOLOGY

We define how rationales are constructed, checked for label cues, assigned during training, and evaluated for their effect on action judgment. PreAct supplies both reconstructed training data and the alternative rationales used in our controlled assignment experiments.

3.1 TASK SETUP

A WebPRM chooses between candidate browser actions using the user's goal, current page, interaction history, and any rationales supplied with the candidates. These rationales explain why the action-proposing model, or *actor*, suggests each action; they are not the WebPRM's own reasoning. The WebPRM compares candidate scores or outputs a *preference verdict* identifying its choice. We call the action labeled as better *preferred* and its alternative *rejected*. *Preference accuracy* is the proportion of pairs in which the model selects the preferred action. Tests without current rationales remove only the candidate explanations. Appendix A.1 specifies scoring and the handling of ties and invalid outputs.

3.2 PREACT: LABEL-BLIND RATIONALE RECONSTRUCTION

Native denotes the original dataset rationales. *PreAct* reconstructs both candidates' rationales through one procedure, regardless of their labels. The same generator explains each action separately using the goal, current page, prior actions, and that candidate action. It cannot see preference labels, the competing candidate, the original rationale, or execution outcomes. The labels remain available for training the WebPRM, but not for generating its input rationales. For the final training corpus, factual review corrects unsupported claims and action-binding errors before the text is

frozen. Only the candidate rationales change; the task, page, history, actions, and labels are preserved. Appendices A.3 and A.7 document the text versions and review procedures.

Native training uses original rationales for both candidates; *PreAct training* uses reconstructed rationales for both. Either model can be evaluated with original, reconstructed, or no current rationales. This separates the effect of training data from the effect of providing explanations at test time.

3.3 Label Probes and Source Assignments

We use lightweight text classifiers, or *probes*, to test whether candidate text predicts preference labels without the page, goal, or history. Using scikit-learn (Pedregosa et al., 2011), we fit logistic regression on term frequency–inverse document frequency (TF–IDF) features. A pair counts as correct when the preferred candidate receives the higher score. Five-fold cross-validation keeps candidates from the same state or step together; a further check separates complete trajectories. Appendix A.2 gives the inputs, features, and splits.

Probes test what the text reveals. To test what a WebPRM learns, we prepare a Native and a PreAct rationale for each training action, treating the two generation procedures as rationale *sources*. Unlike Native and PreAct training, which use the same source for both candidates, the experiments in Table 1 give one candidate a Native rationale and the other a PreAct rationale. Aligned and Reversed differ in which source accompanies the preferred action. Shuffled assigns half the states to each direction once before training and keeps those assignments fixed across updates and seeds. States, actions, labels, and available texts stay unchanged.

Table 1: Rationale-source assignments. Shuffled splits states equally between Aligned and Reversed.

Assignment	**Preferred action's rationale**	**Rejected action's rationale**
Aligned	Native	PreAct
Reversed	PreAct	Native
Shuffled	50% Native / 50% PreAct	50% PreAct / 50% Native

To measure learned source preference, we evaluate each trained model twice on the same held-out action pairs: once with the Aligned assignment and once with the Reversed assignment. Its *source gap* is Aligned-test accuracy minus Reversed-test accuracy. A positive gap means the model is more accurate when preferred actions have Native rationales; a negative gap means it is more accurate when they have PreAct rationales. This compares two test conditions for one model, not two differently trained models. Appendix A.1 specifies scoring; Appendix A.6 describes the mid-training intervention.

3.4 WebContrast: Goal Judgment and Rationale Robustness

WebContrast tests whether models follow goal changes, remain stable across rationale versions, and benefit from reading rationales. Following the behavioral-testing approach of CheckList (Ribeiro et al., 2020), it provides paired tests on 300 pages from 119 websites, derived from Mind2Web (Deng et al., 2023). For *goal-directed action judgment*, each page has two candidate actions, A and B, and two goals tested separately: one requires A and the other B. The page and actions stay fixed, so always favoring either candidate cannot answer both goals correctly.

WebContrast uses its own test rationales. Unlike PreAct reconstruction, it withholds the goals during generation so the explanations can stay unchanged when the goal switches. Two generators each explain A and B from the same page facts and the respective action, without labels, the competing action, or later outcomes. In the first input version, A uses generator 1's explanation of A, and B uses generator 2's explanation of B. In the second, A uses generator 2's explanation of A, and B uses generator 1's explanation of B. Thus, changing versions never moves an explanation to a different action.

Two-goal accuracy is the proportion of pages where both goals receive the correct action, rather than the average accuracy of the two goals. With rationales, we calculate this score separately for each input version and then average. Our central measure, the *rationale effect*, compares two-goal accuracy for the same model with and without candidate rationales:

$$\text{Rationale effect} = \text{Accuracy}_{\text{with rationales}} - \text{Accuracy}_{\text{without rationales}}. \quad (1)$$

Positive values mean net help; negative values mean net harm. We compute the effect in Eq. (1) within each model before comparing training groups. All groups within an experiment receive identical test inputs, so the comparison measures how training changes the influence of the same explanations.

For *robustness to rationale variation*, we hold the goal fixed and compare choices across the two input versions. The *rationale-version flip rate* is the fraction of page-and-goal cases where the selected action changes, among cases with definite choices under both versions. A lower rate means greater stability when the correct action is unchanged. To reduce position effects, both tests combine preference scores from candidate orders A/B and B/A before selecting an action. Orders and versions are repeated measurements of the same page. Appendix A.1 gives scoring and unresolved-choice rules; Appendix A.5 documents construction and review.

Finally, we ask whether a better rationale effect reflects useful explanation content or greater resistance to added text. For Native and PreAct models, we replace rationales with *neutral text*: task-unrelated prose of approximately matched word and token counts. Comparing accuracy with rationales versus neutral text measures their advantage over this control; comparing neutral text versus no rationales measures interference from unrelated text. Appendix A.8 gives the control's construction and length checks.

4 Experiments and Analysis

4.1 Experimental Setup

We audit the data shared by WebArbiter and Web-Shepherd, test six released checkpoints, and study training effects on Qwen3-8B (Qwen Team, 2025) and Llama-3.1-8B-Instruct (Llama Team, 2024). Within each backbone and seed, groups in a training experiment share initialization, states, actions, labels, and update budgets; only candidate rationales differ. We use supervised fine-tuning (SFT) with low-rank adaptation (LoRA; Hu et al., 2022) to predict the preferred-action verdict and report two-seed means. Appendices A.4 and A.7 give training and per-seed details. Accuracy is reported in percent and differences in percentage points (pp). We use paired bootstrap confidence intervals (Dror et al., 2018). Our 95% intervals resample whole tasks or pages while keeping each example's predictions together, as specified in Appendix A.1.

4.2 Label Cues in Rationale Construction

WebPRM Collection supplements human-demonstrated preferred actions with rationales generated from the action, current observation, and an after-execution screenshot. Rejected candidates instead carry rationales from unexecuted agent proposals (Chae et al., 2025, Appendix B.3); WebArbiter reuses this collection (Zhang et al., 2026, Section 3.2). Thus, preference labels are associated with different rationale-generation conditions. The text statistics in Figure 2, covering 9,460 preferred and 30,921 rejected records, show corresponding differences: outcome-related expressions occur in 40.31% versus 9.23%, alongside differences in pronouns and length. Appendix A.2 gives the word lists and counting rules.

To test whether rationale text predicts labels, we apply the probes from Section 3.3 to WebPRM-Bench's 4,600 action pairs and WebRewardBench's 3,104 pairs. The comparisons in Table 2(a) use actions, rationale openings, full rationales, or equal-length rationales. Full rationales recover labels with 97.70% and 99.94% accuracy; the first five words alone exceed 85%, and equal-length truncation barely changes the full-text result. Keeping each of WebRewardBench's 220 complete trajectories within one fold also leaves accuracy at 99.94%, as reported in Table 2(b). These signals survive both length matching and trajectory isolation, motivating us to test whether released WebPRMs also respond to rationale construction.

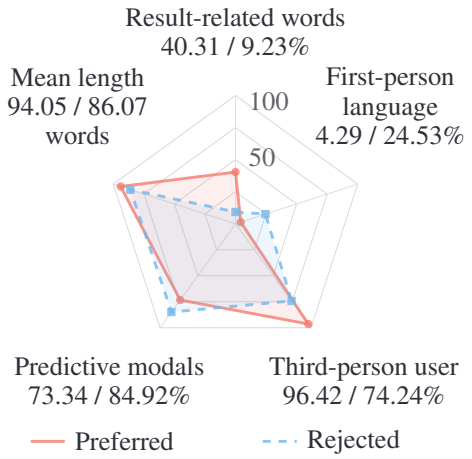


Figure 2: Preferred/rejected values for rationale features: rates (%) and mean length (words), on a 0–100 scale.

Table 2: Label-probe accuracy (%) with states (a) or full trajectories (b) kept within folds. PRM-B: WebPRMBench; Reward-B: WebRewardBench. CIs resample trajectories.

Probe input	**PRM-B**	**Reward-B**
Action only	64.50	57.93
First 5 words	85.41	86.98
Full rationale	**97.70**	**99.94**
Equal words	97.46	99.87

(a) State-isolated probes

Probe input	**Reward-B**	**95% CI**
Action only	59.86	[57.26, 62.43]
Full rationale	**99.94**	[99.84, 100.00]
Equal words	99.81	[99.64, 99.94]

(b) Trajectory-isolated probes

4.3 Released WebPRMs Respond to Rationale Construction

We replace Native rationales with PreAct rationales while fixing tasks, pages, histories, actions, and labels. Each checkpoint uses 60 pairs from its family's benchmark: WebPRMBench for WebArbiter and WebRewardBench for Web-Shepherd. We retain their respective verdict and checklist-scoring interfaces. All six checkpoints in Table 3 lose accuracy, by 21.67 points on average. On the same pairs, we also remove the task, page, and history, retaining only Native rationales, only actions, or neither. Rationales alone still yield 75.00–88.33% accuracy without the page or explicit actions. Together, these results show that released WebPRMs can predict labels from rationales and that their scores depend on how those rationales are constructed.

Table 3: Released-model preference accuracy (%). Candidate-only rationales are Native.

	Full context		Candidate only		
Checkpoint	Native	PreAct	Rationale	Action	None
WebArbiter-3B	83.33	68.33	80.00	50.00	50.00
WebArbiter-4B-Qwen3	91.67	65.00	85.00	55.00	50.00
WebArbiter-7B	85.00	65.00	81.67	51.67	50.00
WebArbiter-8B-Qwen3	98.33	75.00	88.33	58.33	50.00
WebShepherd-3B	93.33	71.67	75.00	46.67	43.33
WebShepherd-8B	93.33	70.00	78.33	63.33	46.67

Could the decrease simply reflect rewording, one generator, or length? The controls in Figure 3 test these alternatives on the two 8B models using the same full-context pairs and unchanged Native and PreAct references. Paraphrasing Native with content-preservation instructions leaves accuracy closer to Native than to PreAct, so ordinary rewording does not reproduce the drop. Generating PreAct with Llama-3.1-8B gives scores within five points of PreAct on both models, showing that the drop is not specific to one generator. Matching PreAct to within five words of Native changes little for WebArbiter; it raises Web-Shepherd's accuracy by 10.00 points but leaves a 13.33-point gap to Native. Length can therefore contribute without explaining the full decrease. Appendix A.3 provides expanded-panel results and text-quality checks. Since the source–label association also appears in training, we next test whether it teaches a source preference and changes how rationales affect action judgment.

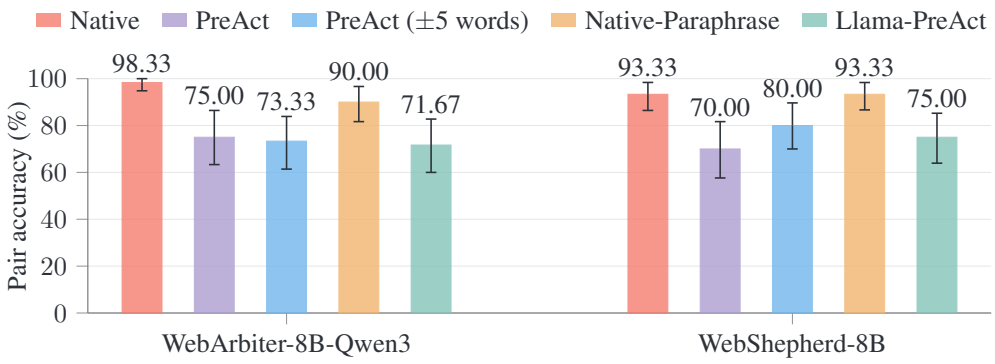


Figure 3: Controls for rationale length, wording, and generator. Error bars show 95% cluster-bootstrap confidence intervals.

4.4 Source–Label Pairing Changes the Effect of Explanations

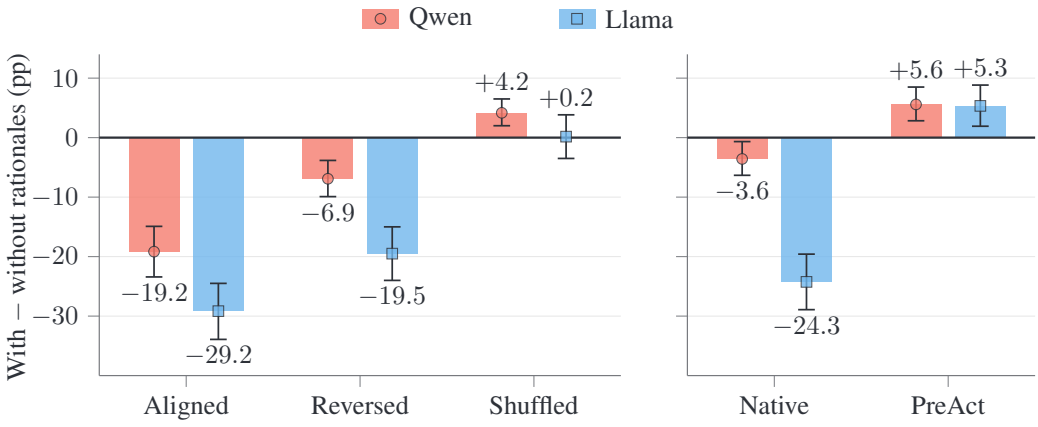
We hold the available rationale texts fixed and change only their source–label assignment. Aligned, Reversed, and Shuffled use the same 512 Collection states and train for 96 updates. Testing on 512 held-out states from 80 tasks shows that models learn the assigned preference: in Table 4, reversing the assignment reverses the source gap, while shuffling reduces it to about one point on both backbones. Thus, the training assignment can determine which rationale source a model favors.

Table 4: Source preference and WebContrast performance (%). Gap: Aligned−Reversed accuracy (pp); CI: paired 95% task-cluster interval. Two-goal accuracy omits current rationales. Qwen: Qwen3-8B; Llama: Llama-3.1-8B-Instruct.

		Source preference				WebContrast	
Model	**Training**	**Aligned**	**Reversed**	**Gap**	**95% CI**	**2-goal acc. ↑**	**Flip ↓**
Qwen	Aligned	**99.41**	6.35	+93.07	[90.50, 95.59]	78.50	30.02
	Reversed	8.89	**98.34**	−89.45	[−92.43, −86.04]	87.00	12.97
	Shuffled	81.05	79.69	**+1.37**	[−1.64, 4.37]	**89.67**	**2.00**
Llama	Aligned	**99.12**	5.57	+93.55	[90.90, 95.81]	49.33	57.90
	Reversed	4.39	**99.02**	−94.63	[−97.03, −91.65]	49.67	49.18
	Shuffled	86.43	85.35	**+1.07**	[−2.04, 4.25]	**61.50**	**18.61**

To test whether the training assignment also changes how rationales affect action judgment, we evaluate the same models on WebContrast, where always favoring one candidate cannot satisfy both goals. Each model receives identical pages, goals, actions, and rationale texts, and we compute each model's rationale effect using Eq. (1). As shown in Figure 4(a), adding rationales lowers accuracy by 19.17 points for Aligned Qwen and 29.17 for Aligned Llama. After Shuffled training, their effects are +4.17 and +0.17 points: helpful for Qwen and approximately neutral for Llama. Their net effects thus improve by 23.33 and 29.33 points. For both backbones and both seeds, rationales help more or harm less after Shuffled training than after either fixed assignment. This also holds when each goal is scored separately instead of requiring both answers to be correct; Appendix A.5 gives the paired intervals and sensitivity analyses.

Shuffled training also improves both abilities in Table 4: without rationales, it answers both goals correctly more often than either fixed-assignment group; with rationales, it changes choices less often across explanation versions. A mid-training intervention shows that an acquired source preference can also be weakened. Switching Qwen from Aligned to Shuffled after 32 updates reduces the final source gap from 92.19 to 8.12 points and raises no-rationale two-goal accuracy by 4.50 points relative to continued Aligned training (Appendix A.6). **Changing the training association changes how the same explanations affect action judgment.** This motivates us to apply the insight when constructing the training rationales themselves.



(a) Controlled source assignment (b) Native / PreAct training

Figure 4: Effect of adding rationales on WebContrast. (a) Controlled source–label assignments: Aligned, Reversed, and Shuffled. (b) Original-data (Native) versus PreAct training. Values show two-goal accuracy with minus without rationales (pp): positive indicates help, negative indicates harm. Bars average two seeds; error bars show 95% page-cluster intervals. Test inputs are fixed within each panel; prompts and inference interfaces differ between panels.

4.5 PreAct Reduces Misleading Use of the Same Rationales

PreAct applies this insight to rationale production: both candidates are explained through the same label-blind procedure in Section 3.2. We compare Native and PreAct training on the same 1,024 Collection states for 192 updates. On WebContrast, the same rationales change from mean net harm to net help in Figure 4(b): their effect changes from −3.58 to +5.58 points for Qwen and from −24.25 to +5.33 for Llama. The paired improvements average 9.17 and 29.58 points and are positive in both seeds of each backbone. Without rationales, both Qwen models score about 89%. Adding the same rationales lowers Native's accuracy but raises PreAct's, showing that training changes whether the model benefits from reading them. To understand this reversal, we count errors corrected and correct judgments overturned by rationales. **PreAct preserves the error-correcting benefit of rationales while substantially reducing how often they overturn correct judgments.** Across all pages, this harm rate falls from 7.42% to 1.42% for Qwen and from 31.50% to 4.92% for Llama. Appendix A.7 gives the help/harm decomposition and per-seed results.

Does this benefit require rationale content, or would adding unrelated text have the same effect? The control in Table 5 compares no rationales, neutral text, and real rationales on the same pages and models. After PreAct training, real rationales outperform neutral text by 7.33 points for Qwen and 7.75 for Llama, and also outperform no rationales. These unrelated texts do not reproduce their benefit. PreAct also reduces interference from neutral text, so the improvement includes greater resistance to added text.

Table 5: Neutral-text control on WebContrast. Two-goal accuracy (%), averaged over two seeds. R: the same WebContrast rationales; N: neutral text. Differences are in pp.

		Input / accuracy (%)			Difference (pp)	
Model	**Training**	**None**	**Neutral**	**Rationale**	**R−N**	**N−none**
Qwen	Native	89.00	74.33	85.42	+11.08	−14.67
	PreAct	88.67	86.92	94.25	+7.33	−1.75
Llama	Native	63.33	32.92	39.08	+6.17	−30.42
	PreAct	74.50	72.08	79.83	+7.75	−2.42

4.6 Original Evaluation Can Favor Less Effective Rationale Use

We next ask whether the improvement extends beyond WebContrast and whether the original evaluation recognizes it. The comparison in Table 6 uses the same Native and PreAct checkpoints. On the Collection holdout of 256 states from 80 tasks, original rationales preserve the existing evaluation; pre-action and no-rationale views test the same action pairs without those original explanations. For transfer, WebLINX (Lù et al., 2024) supplies 204 demonstration-based pairs without actor rationales. For WebShop (Yao et al., 2022), public search-tree logs released with AgentRM (Xia et al., 2025) test selection by observed branch outcomes: 116 unequal-reward pairs from 47 tasks are labeled by recorded branch rewards and evaluated with and without natural actor rationales. Appendix A.7 gives construction, weighting, and full-set results.

Table 6: Native versus PreAct training across rationale inputs and datasets. Preference accuracy (%), averaged over two seeds; WebShop uses unequal-reward logged branches. Qwen: Qwen3-8B; Llama: Llama-3.1-8B-Instruct.

		Qwen		Llama	
Dataset	**Input / metric**	**Native**	**PreAct**	**Native**	**PreAct**
Collection	Native rat. ↑	**99.02**	84.77	**99.32**	78.81
	PreAct rat. ↑	73.05	**83.40**	67.97	**80.96**
	No rat. ↑	75.88	**81.64**	60.74	**77.54**
WebLINX	No rat. ↑	79.04	**82.48**	51.23	**58.82**
WebShop	Natural rat. ↑	63.25	**64.32**	46.33	**64.37**
	No rat. ↑	63.95	**65.17**	54.94	**66.88**

Across the comparisons in Table 6, PreAct improves Collection accuracy with pre-action rationales by 10.35 points for Qwen and 12.99 for Llama, with gains also without rationales and on WebLINX. Both WebShop views show higher mean accuracy; for Llama, the mean loss from adding natural rationales also falls from 8.62 to 2.51 points. The mean gains without rationales on Collection and WebLINX show that PreAct's benefits do not require reconstructed explanations at test time. They extend beyond matching an explanation format to improved offline action selection. Yet evaluation with Collection's original rationales reverses the ranking: Native exceeds 99% on both backbones and outperforms PreAct. Choosing by this score would favor the recipe with lower mean accuracy in this table's other views and greater interference from the same WebContrast rationales. **Higher original-rationale scores need not indicate more effective explanation use.** Model comparisons should therefore report goal judgment and the effect of adding explanations alongside original-rationale accuracy.

5 Conclusion

Training-data construction can change whether the same explanations help or hinder action judgment. Controlled experiments identify source–label pairing as an influential training choice. PreAct reduces explanation-induced errors and reverses their mean effect on WebContrast, yet original-rationale evaluation favors the more interference-prone recipe. Candidate rationales should therefore be generated through a common pre-action procedure that does not depend on preference labels, while evaluation should test whether choices follow the user's goal and whether explanations improve them. PreAct provides one such construction procedure; WebContrast provides paired tests of these behaviors.

AI Use Statement

Generative AI was used only for auxiliary review and auditing of selected qualitative outputs. These AI-assisted checks were used to flag potential issues for further author inspection and were not treated as independent human judgments or ground-truth annotations. Their scope is documented in

Appendices A.5, and A.7. All final decisions, revisions, claims, code, and data artifacts remain the responsibility of the authors.

REFERENCES

Hyungjoo Chae, Seonghwan Kim, Junhee Cho, Seungone Kim, Seungjun Moon, Gyeom Hwangbo, Dongha Lim, Minjin Kim, Yeonjun Hwang, Minju Gwak, Dongwook Choi, Minseok Kang, Gwanhoon Im, ByeongUng Cho, Hyojun Kim, Jun Hee Han, Taeyoon Kwon, Minju Kim, Beong-woo Kwak, Dongjin Kang, and Jinyoung Yeo. Web-Shepherd: Advancing PRMs for Reinforcing Web Agents. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025.*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/5bac36213824c6e0b0a173df8e2276a6-Abstract-Conference.html.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samual Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a Generalist Agent for the Web. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/5950bf290a1570ea401bf98882128160-Abstract-Datasets_and_Benchmarks.html.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 1383–1392. Association for Computational Linguistics, 2018. doi: 10.18653/V1/P18-1128. URL https://aclanthology.org/P18-1128/.

Benjamin Feuer, Micah Goldblum, Teresa Datta, Sanjana Nambiar, Raz Besaleli, Samuel Dooley, Max Cembalest, and John P. Dickerson. Style Outweighs Substance: Failure Modes of LLM Judges in Alignment Benchmarking. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=MzHNftnAM1.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 107–112. Association for Computational Linguistics, 2018. doi: 10.18653/V1/N18-2017. URL https://doi.org/10.18653/V1/N18-2017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. RM-Bench: Benchmarking Reward Models of Language Models with Subtlety and Style. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=QEHrmQPBdd.

Llama Team. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/ARXIV.2407.21783.

Xing Han Lù, Zdenek Kasner, and Siva Reddy. WebLINX: Real-World Website Navigation with Multi-Turn Dialogue. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33007–33056. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/lu24e.html.

Bingchen Miao, Yang Wu, Minghe Gao, Qifan Yu, Wendong Bu, Wenqiao Zhang, Yunfei Li, Siliang Tang, Tat-Seng Chua, and Juncheng Li. Boosting Virtual Agent Learning and Reasoning: A Step-Wise, Multi-Dimensional, and Generalist Reward Model with Benchmark. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/miao25b.html.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM Evaluators Recognize and Favor Their Own Generations. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024.*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/7f1f0218e45f5414c79c0679633e47bc-Abstract-Conference.html.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011. URL https://dl.acm.org/doi/10.5555/1953048.2078195.

Qwen Team. Qwen3 Technical Report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/ARXIV.2505.09388.

Marco Túlio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 4902–4912. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.ACL-MAIN.442. URL https://doi.org/10.18653/V1/2020.ACL-MAIN.442.

Pragya Srivastava, Harman Singh, Rahul Madhavan, Gandharv Patil, Sravanti Addepalli, Arun Suggala, Rengarajan Aravamudhan, Soumya Sharma, Anirban Laha, Aravindan Raghuveer, Karthikeyan Shanmugam, and Doina Precup. Robust Reward Modeling via Causal Rubrics. *CoRR*, abs/2506.16507, 2025. doi: 10.48550/ARXIV.2506.16507. URL https://doi.org/10.48550/ARXIV.2506.16507.

Zhaofeng Wu, Michihiro Yasunaga, Andrew Cohen, Yoon Kim, Asli Celikyilmaz, and Marjan Ghazvininejad. reWordBench: Benchmarking and Improving the Robustness of Reward Models with Transformed Inputs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 3383–3409. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.167. URL https://doi.org/10.18653/V1/2025.EMNLP-MAIN.167.

Yu Xia, Jingru Fan, Weize Chen, Siyu Yan, Xin Cong, Zhong Zhang, Yaxi Lu, Yankai Lin, Zhiyuan Liu, and Maosong Sun. AgentRM: Enhancing Agent Generalization with Reward Modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 19277–19290. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.945. URL https://doi.org/10.18653/V1/2025.ACL-LONG.945.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

Yao Zhang, Shijie Tang, Zeyu Li, Zhen Han, and Volker Tresp. WebArbiter: A Principle-Guided Reasoning Process Reward Model for Web Agents. *CoRR*, abs/2601.21872, 2026. doi: 10.48550/ARXIV.2601.21872. URL https://doi.org/10.48550/ARXIV.2601.21872.

A Additional Experimental Details and Results

The appendix follows the evidence chain: scoring and probes (A.1–A.2), released-model controls (A.3), controlled training and WebContrast (A.4–A.5), mid-training intervention (A.6), real-data reconstruction and transfer (A.7), the neutral-text control (A.8), and reproducibility (A.9).

A.1 Scoring, Denominators, and Statistical Uncertainty

Evaluation units and scoring rules are summarized in Table A1. Candidate orders, rationale versions, and repeated predictions are measurements of the same example, not independent examples. No-rationale tests remove only current candidate explanations.

Table A1: Evaluation units and scoring rules.

Result	Evaluation set	Scoring rule	Resampling unit
Table 2: text probes	4,600 / 3,104 pairs	Preferred candidate has a strictly higher classifier score	Source state/step; complete trajectory in the additional WebRewardBench check
Table 3 / Figure 3: released models	60 pairs per model	Family-specific verdict or checklist score	Source cluster within the frozen panel
Table 4: source assignment	512 states, 80 tasks	Both AB and BA orders must select the preferred action	Task
Table 4: WebContrast	300 pages, 119 websites	Order-symmetrized verdict margin; both goals correct	Page; website sensitivity analysis
Figure A1: source gap	128 states, 80 tasks	Complete-verdict likelihood; average order correctness, then tasks	Task
Figure A1: goal judgment	WebContrast (the same 300 pages)	Order-symmetrized verdict margin; no current rationales	Page; website sensitivity analysis
Table 6: Collection / WebLINX	256 states / 204 reference pairs	Verdict correctness, averaged over AB/BA	Task
Table 6: WebShop	116 non-tied pairs, 47 tasks	Higher logged-reward branch selected; tasks weighted equally	Task
Figure 4 / Table A13: rationale effect	WebContrast (the same 300 pages)	With-minus-without two-goal accuracy; average versions and seeds within page	Page; website sensitivity analysis
Table 5: neutral-text control	WebContrast (the same 300 pages)	Rationale, neutral text, and no-rationale accuracy; paired within-model differences and training interactions	Page; website sensitivity analysis

For WebContrast scoring, let d_{AB} be the log-probability difference between selecting the first and second candidates when actions appear in order A, B; d_{BA} is the corresponding difference in the reversed order. We map both orders to the same action identities and use

$$m = \frac{d_{AB} - d_{BA}}{2}. \tag{A1}$$

We select A when the margin in Eq. (A1) exceeds 10^{-4} and B when it is below -10^{-4}. An unresolved margin or an invalid or truncated verdict does not count as a correct goal response. Both goals must be answered correctly for a page to receive credit. When rationales are present, this requirement is applied separately within each fixed rationale version before averaging; success in either version is not sufficient. Appendix A.5 also reports results from the generated hard verdicts.

For rationale-version sensitivity, let F be confirmed flips, U unresolved comparisons, and N all planned comparisons. We report the conditional flip rate $F/(N-U)$ in Table 4 and Table A16, calculated separately for each training seed and then averaged. Appendices A.5 and A.7 retain the full-denominator bounds $[F/N, (F+U)/N]$ and resolved coverage $(N-U)/N$. The conditional rate is not the midpoint of the bounds. For WebContrast's 300 pages, each seed contributes 600 goal-specific version comparisons.

The paired analyses retain each example's predictions under both conditions during resampling. For trained models, seeds 20260901 and 20260917 are fixed; their scores are averaged within each resampled unit. We use 10,000 bootstrap resamples for the WebContrast and transfer analyses. These intervals quantify variation across tasks, pages, or websites conditional on those seeds. They do not estimate uncertainty across all possible training runs. Differences are computed before rounding. Table-wise intervals are exploratory and not jointly multiplicity-corrected; the simultaneous band in Appendix A.6 covers only its intervention curve.

A.2 Data Construction and Text Probes

The construction audit follows the rationale-generation procedure documented in Chae et al. (2025, Appendix B.3) and the downstream use of WebPRM Collection in Zhang et al. (2026, Section 3.2). The former permits an after-action screenshot when supplementing rationales for demonstrated actions, while rejected candidates are policy proposals. Thus, our cross-family audit covers different evaluation architectures that share data lineage, not two wholly independent data-production pipelines.

Table A2: Raw lexical measurements underlying Figure 2.

Feature	**Preferred: 9,460 records**	**Rejected: 30,921 records**
Outcome-lexicon hit	40.31%	9.23%
First-person expression	4.29%	24.53%
Third-person user expression	96.42%	74.24%
Prospective modal	73.34%	84.92%
Mean word count	94.05	86.07

We use four fixed, rule-based lexicons to describe narrative perspective, prospective modality, and outcome-related wording. The first-person list is "I, I'm, I'll, my, we, our"; third-person user expressions are "the user, user needs, user wants"; modal words are "should, would, may, might, likely, can, could, will." The outcome list combines completion expressions and descriptions of a resulting state: "successfully, has opened, has navigated, is now, was added, were added, has been, completed, is displayed, now shows." These explicit lists define the descriptive features in Figure 2; they are separate from the learned TF–IDF features used by the label probes below.

Matching lowercases each rationale and searches for the listed forms with word boundaries, without stemming, synonym expansion, negation handling, or whitespace normalization. Multiword entries use literal single spaces. A rationale counts once for a feature if at least one listed expression appears, regardless of repeated hits. We count every preferred record and every rejected record across the ten local data shards, without deduplication or task weighting. The outcome rates are therefore 3,813/9,460 preferred records and 2,854/30,921 rejected records. Mean word counts use the regular expression `[A-Za-z]+(?:'[A-Za-z]+)?`, which counts ASCII alphabetic words with an optional internal straight apostrophe; hyphenated words are split and numbers are excluded. A lexicon hit marks wording rather than verified information leakage; the same phrase may describe a legitimate past event.

The probes use scikit-learn (Pedregosa et al., 2011). Rationale features are word unigrams and bigrams; action features use character-within-word n-grams of lengths 2–5. TF–IDF uses a minimum document frequency of 2, at most 50,000 features, lowercase text, and sublinear term frequency. L2-regularized logistic regression uses $C = 1$, the liblinear solver, and at most 2,000 iterations. Vocabulary and classifier fitting occur within each training fold. Five deterministic grouped folds contain 1,150 state/step groups for WebPRMBench and 776 for WebRewardBench. Equal-length

truncation uses the shorter candidate's whitespace-token count within each pair. Classifier ties are not counted as correct.

The trajectory-isolated check in Table 2(b) retains the same 3,104 pairs and keeps all steps of each episode in one fold. Across source strata, full-rationale accuracy ranges from 92.22% to 100.00% on WebPRMBench and from 99.82% to 100.00% on WebRewardBench. These strata share underlying data and are not independent replications. Full prefix-length sweeps are retained in the reproducibility records; the manuscript reports the five-word and full-text comparisons.

A.3 Released-Model Controls and Text Quality

All Table 3 conditions use the same frozen pairs within each model. Information ablations remove the task, page, URL, and history as well as the omitted candidate fields. Web-Shepherd uses a task-independent checklist without context and samples five feedbacks per candidate. Figure 3 retains Table 3's Native and PreAct references; Table A3 reports its 60-pair controls, including a second sampling seed for Web-Shepherd. These seeds change evaluation sampling, not model training.

Table A3: Released-model controls (accuracy, %; differences, pp). Length: ±5-word PreAct; Para.: Native paraphrase; Llama: Llama-PreAct; Δlength: Native − Length.

Model / seed	**Native**	**PreAct**	**Length**	**Para.**	**Llama**	**Δlength [95% CI]**
WA-8B / 20260901	98.33	75.00	73.33	90.00	71.67	+25.00 [13.33, 37.29]
WS-8B / 20260901	93.33	70.00	80.00	93.33	75.00	+13.33 [1.69, 25.00]
WS-8B / 20260917	91.67	58.33	65.00	91.67	76.67	+26.67 [15.79, 38.60]

For the length control, an editor receives the existing PreAct text, its action, and a target word count, but not the Native text, preference label, or later outcome. All 240 edited rationales are within five whitespace-separated words of their Native counterparts; the mean absolute difference is 2.15 words. Texts and membership were frozen before scoring, without using PRM predictions to select edits or examples. AI-assisted review addressed flagged passages and material binding or factual errors. It did not independently verify every inherited statement in the initial PreAct text.

To isolate surface rewriting from semantic changes, we constructed meaning-preserving controls for every evaluated rationale. Each rewritten rationale was required to preserve the source rationale's action binding, task-relevant facts, temporal status, epistemic modality, and outcome claims. Two independent reviewers verified bidirectional content preservation for every candidate rationale, with disagreements adjudicated before model evaluation. Texts that failed any preservation criterion were revised and re-audited until they passed; no examples were selected or removed based on PRM predictions. All evaluated paraphrase pairs passed the final preservation audit.

For the second-generator control, both generators received exactly the same pre-action information, and the resulting rationales were likewise verified to express the same task-relevant propositions. Thus, the paraphrase condition changes surface realization while holding semantic content fixed, and the second-generator condition changes generator identity while holding both information access and verified semantic content fixed. The ±5-word condition additionally matches rationale length without changing these verified propositions. All texts and panel membership were frozen before scoring.

These controls therefore isolate wording, generator identity, and length from task-relevant semantic content.

The expanded panels use separate frozen text versions and overlap with the smaller panels, so their sample counts are not added. Nine of ten family-by-source strata show a decrease. The exact-text non-overlap check rules out confinement to exactly reused text, but not broader task or semantic overlap.

A.4 Controlled Training and Per-Seed Source Preferences

All source-assignment groups use 512 training states, expanded to 1,024 rows by reversing candidate order. The source library contains a Native and a PreAct rationale for each action. Aligned, reversed,

Table A4: Expanded-panel and exact-text non-overlap checks (%, pp), with 95% cluster CIs.

Model / subset	Pairs / clusters	Native	PreAct	Decrease [95% CI]	Correct→wrong / wrong→correct
WA-8B	204 / 183	92.65	71.08	21.57 [15.12, 28.28]	50 / 6
WS-8B	200 / 126	92.50	66.00	26.50 [19.80, 33.33]	57 / 4
WA-8B (non-overlap)	102 / 89	85.29	70.59	14.71 [4.85, 24.75]	21 / 6

and shuffled groups change which of these versions is selected, not the state, action, or label. The shuffled allocation is fixed by a state hash before training and is shared across training seeds. Two training seeds therefore do not constitute two independent randomizations of the source allocation. The 512 evaluation states from 80 tasks are held out from the training states.

Table A5: Training settings.

Setting	Source assignment	Reconstruction
Backbones	Qwen; Llama	Same
Training states / order-expanded rows	512 / 1,024	1,024 / 2,048
Updates	96	192
Training seeds	20260901, 20260917	Same
Target	Preferred-action verdict	Same
LoRA rank / alpha / dropout	64 / 128 / 0.05	Same
LoRA targets	All supported linear targets	Same
Learning rate / schedule	10^{-4} / cosine	Same
Warm-up ratio	0.10	Same
Microbatch / accumulation	1 / 32	Same
Maximum input length	8,192 tokens	Same
Precision / attention	BF16 / SDPA	Same
Gradient checkpointing / packing	Enabled / disabled	Same
Loss on input prompt	Disabled	Same

Both experiments use the optimization settings in Table A5 and backbone-specific chat templates, including the configured Qwen3 thinking mode. Targets supervise only the final preference verdict. Groups match updates and example identities, but rationale lengths differ, so input-token totals and FLOPs are not exactly matched. We evaluate fixed final checkpoints, not the best test checkpoint. Per-seed results in Table A6 use Table 4's strict two-order rule.

The reported Qwen source probe uses 12,288 stable-backend predictions with no invalid or truncated outputs. Checks cover singleton and batched requests, reordered batches, and a cold restart. Earlier-backend predictions are not mixed into the reported results.

A.5 WebContrast: Construction, Coverage, and Sensitivity

WebContrast contains 300 distinct source-task IDs and pages from 119 websites, derived from Mind2Web (Deng et al., 2023). We deduplicate page IDs, source HTML hashes, and normalized observations. The 600 candidate actions comprise 458 clicks, 78 fill actions, and 64 option selections. Each page supplies two valid goals that require different actions from the same fixed pair. The label check asks whether the visible page and goal identify the intended action, not whether a future page transition is guaranteed to succeed.

Table A6: Per-seed source-probe accuracy (%) and source gap (pp).

Backbone	Assignment	Seed	Aligned	Reversed	Gap (pp)
Qwen	Aligned	20260901	99.41	5.27	+94.14
Qwen	Aligned	20260917	99.41	7.42	+91.99
Qwen	Reversed	20260901	6.64	98.83	−92.19
Qwen	Reversed	20260917	11.13	97.85	−86.72
Qwen	Shuffled	20260901	80.08	80.08	0.00
Qwen	Shuffled	20260917	82.03	79.30	+2.73
Llama	Aligned	20260901	98.83	4.49	+94.34
Llama	Aligned	20260917	99.41	6.64	+92.77
Llama	Reversed	20260901	4.88	98.63	−93.75
Llama	Reversed	20260917	3.91	99.41	−95.51
Llama	Shuffled	20260901	88.09	84.38	+3.71
Llama	Shuffled	20260917	84.77	86.33	−1.56

For each page, the construction records the two goals, action targets, checked page facts, and four rationales. Each of two generators writes one rationale per action using the approved facts and action description, without access to either goal, preference labels, the competing action, later outcomes, or PRM scores. Automated checks and AI-assisted review compare page facts, action bindings, and the correct action for each goal against saved HTML. These checks cover target identity, visibility and enabled status, numerical and entity consistency, and the facts needed to distinguish the goals. The review records preserve unresolved cases and revision history and accompany the benchmark. This is not an independent human annotation study or a record of 300 live browser executions.

All WebContrast experiments use the same frozen set of 300 pages. The model comparisons in Table 4, Table A13, Figure A1, and Figure 4 use this shared benchmark.

Rationale effects. Figure 4(a), Table A7, and Table A8 reuse the six training groups' frozen predictions on all 300 pages, with the averaging and bootstrap rules in A.1. The main metric requires both goals to be correct. An individual-goal check instead averages the two goals' correctness. Neither analysis filters pages using model outputs.

Table A7: Controlled-training rationale effects on WebContrast. Accuracies are percentages; effects and paired page-cluster intervals are in percentage points.

Backbone	Assignment	Without rat. (%)	With rat. (%)	Effect (pp)	95% page CI
Qwen	Aligned	78.50	59.33	−19.17	[−23.42, −14.92]
	Reversed	87.00	80.08	−6.92	[−9.92, −3.83]
	Shuffled	89.67	93.83	+4.17	[2.00, 6.50]
Llama	Aligned	49.33	20.17	−29.17	[−33.92, −24.50]
	Reversed	49.67	30.17	−19.50	[−24.00, −15.00]
	Shuffled	61.50	61.67	+0.17	[−3.50, 3.83]

The two-goal effects for seeds 20260901/20260917 are: Qwen Aligned −28.00/−10.33, Reversed −15.50/+1.67, and Shuffled +3.50/+4.83; Llama Aligned −27.33/−31.00, Reversed −18.17/−20.83, and Shuffled −0.33/+0.67. All four contrasts in Table A8 remain positive in each seed. Website-cluster intervals for the two-goal contrasts are [18.87, 27.78], [8.42, 13.87], [23.68, 35.10], and [14.12, 25.09] in table order. These results support reduced harm from the fixed explanations, but not a positive effect for Shuffled Llama. They also do not identify source preference as the sole mediator of action-judgment differences.

Table A8: Improvement in rationale effect after Shuffled training (pp), with paired 95% page-cluster intervals.

Backbone	Comparison	Two-goal effect difference	Individual-goal effect difference
Qwen	Shuffled−Aligned	+23.33 [19.00, 27.75]	+11.62 [9.46, 13.83]
	Shuffled−Reversed	+11.08 [8.25, 13.83]	+5.50 [4.08, 6.88]
Llama	Shuffled−Aligned	+29.33 [23.83, 34.75]	+14.42 [11.67, 17.12]
	Shuffled−Reversed	+19.67 [14.08, 25.17]	+9.25 [6.42, 12.04]

An input audit verifies identical pages, goals, actions, and rationale text across the controlled and Native/PreAct diagnostic suites. The suites use different system prompts and original inference interfaces, while all groups within each suite share the same complete evaluation inputs. We therefore estimate training-by-rationale effects within each suite and do not treat the panels of Figure 4 as one fully matched five-group experiment. These reanalyses were conducted after the earlier results were available.

Table A9: Shuffled minus aligned training on WebContrast (pp).

Backbone	Goal gain [page CI]	Goal gain [website CI]	Flip reduction [page CI]
Qwen	+11.17 [8.00, 14.50]	+11.17 [8.17, 14.14]	+27.75 [24.42, 31.25]
Llama	+12.17 [7.67, 16.83]	+12.17 [7.49, 16.61]	+38.50 [34.25, 42.83]

Flip reduction in Table A9 is computed from the confirmed-flip lower bound over all planned comparisons. The website-cluster intervals are [24.12, 31.36] points for Qwen and [34.10, 42.83] for Llama. Of the full set, 299 pages contain at least one action with distinct rationale texts, and 296 contain distinct versions for both actions.

The readout and coverage checks in Table A10 use all 300 WebContrast pages. Hard two-goal accuracy checks both goals separately within each displayed candidate order, then averages the two orders; margin accuracy uses Appendix A.1. Version coverage is the proportion of resolved rationale comparisons.

Table A10: WebContrast readout sensitivity and coverage (%).

Backbone	Assignment	Seed	Margin acc.	Hard acc.	Coverage
Qwen	Aligned	20260901	74.33	62.83	99.00
Qwen	Aligned	20260917	82.67	77.83	99.33
Qwen	Reversed	20260901	85.67	68.00	98.83
Qwen	Reversed	20260917	88.33	82.17	99.50
Qwen	Shuffled	20260901	90.00	87.00	100.00
Qwen	Shuffled	20260917	89.33	86.33	99.67
Llama	Aligned	20260901	50.67	0.83	98.33
Llama	Aligned	20260917	48.00	29.17	98.00
Llama	Reversed	20260901	55.67	24.00	98.50
Llama	Reversed	20260917	43.67	44.33	99.33
Llama	Shuffled	20260901	58.67	44.00	98.50
Llama	Shuffled	20260917	64.33	59.00	98.50

The absolute scores depend on the readout, especially for aligned Llama. Its two seeds have hard-verdict candidate-order disagreement rates of 77.78% and 37.17% across the WebContrast input

views. Nevertheless, shuffled training improves two-goal accuracy over aligned training in both seeds and under both readouts. For the margin readout, mean goal coverage is 99.25%, 99.83%, and 99.92% for Qwen aligned, reversed, and shuffled models, and 97.83%, 96.17%, and 98.83% for Llama. Unknown goal responses remain in the accuracy denominator.

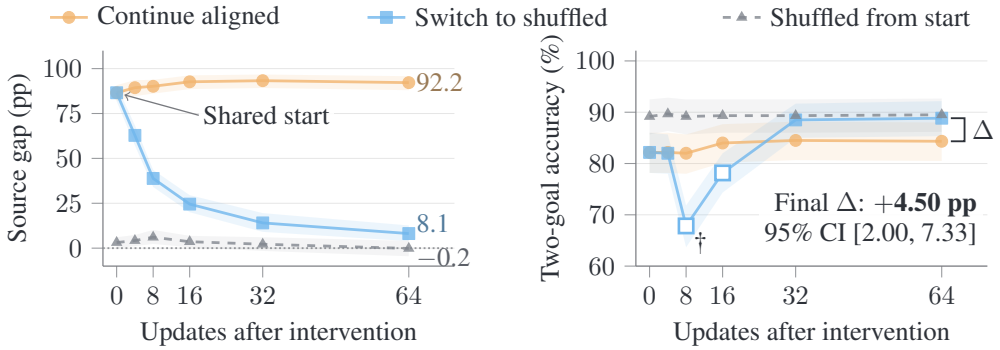
Table A11: Table 4 flip counts and full-denominator bounds (%).

		Flips / resolved, by seed		
Backbone	**Assignment**	**20260901**	**20260917**	**Flip bounds (%)**
Qwen	Aligned	243 / 594	114 / 596	29.75–30.58
Qwen	Reversed	124 / 593	30 / 597	12.83–13.67
Qwen	Shuffled	12 / 600	12 / 598	2.00–2.17
Llama	Aligned	309 / 590	373 / 588	56.83–58.67
Llama	Reversed	269 / 591	315 / 596	48.67–49.75
Llama	Shuffled	139 / 591	81 / 591	18.33–19.83

For each seed's 600 planned comparisons, we report confirmed flips divided by resolved comparisons in Table A11. We average the two rates before rounding in Table 4; the bounds average the per-seed extremes obtained by counting unresolved comparisons as non-flips or flips. The conditional-rate presentation was adopted after inspecting the results, using the same convention as Table A16. The original bounds and lower-bound contrasts remain in this appendix to make the effect of unresolved cases explicit.

A.6 Learning Curves and Mid-Training Intervention

In the intervention experiment, we fork an Aligned Qwen model after 32 updates. One copy retains the Aligned assignment and another switches to Shuffled for 64 further updates. A from-start Shuffled run is the fixed-budget reference. Figure A1 plots the source gap on 128 states from 80 tasks and WebContrast two-goal accuracy without current rationales. The intervention reduces the final gap to 8.12 rather than 92.19 points and improves goal accuracy to 88.83% rather than 84.33%; the paired gain is 4.50 [2.00, 7.33] points. The from-start reference reaches 89.50%.



(a) Source preference **(b) Goal judgment**

† Updates 8 / 16: reduced readout coverage; scores retained.

Figure A1: Mid-training correction of source preference and goal judgment. Shading: pointwise 95% cluster intervals (tasks, left; pages, right). Hollow markers indicate a lower fraction of valid verdicts. The endpoint annotation gives a paired difference.

The learning-curve runs are separate from the final models in Table 4. Figure A2 averages two Qwen3-8B training seeds for Aligned, Reversed, and from-start Shuffled, showing all measured checkpoints. The source probe and WebContrast use the different evaluation sets in Table A1. Step-

zero goal accuracy is unavailable because the configured verdict readout has no valid outputs; it is not scored as zero ability.

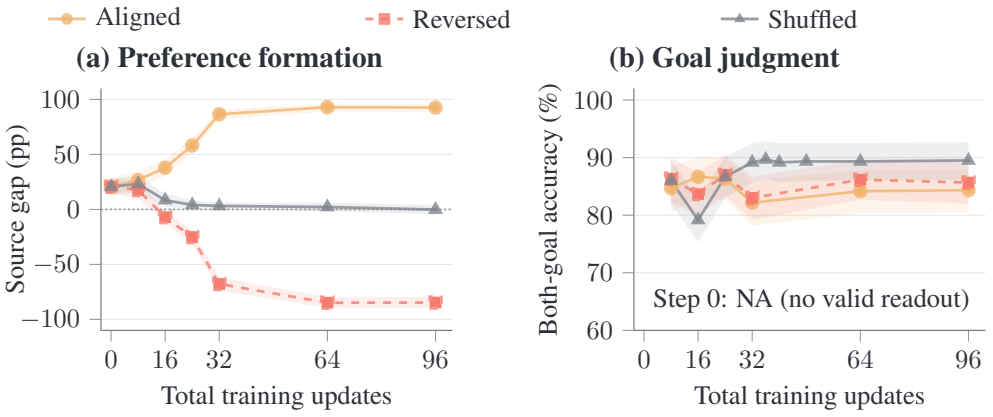


Figure A2: Training curves: source gap (left, pp) and no-rationale two-goal accuracy (right, %). Bands show pointwise 95% task-/page-cluster intervals.

At step 8, Aligned, Reversed, and Shuffled source gaps are 26.72, 17.66, and 23.12 points; by step 32, they reach 86.56, −67.81, and 3.12. In the intervention, continued-Aligned and switched-Shuffled branches restart from the same step-32 model using the same procedure. From-start Shuffled is a step-matched reference, not a branch of that checkpoint.

The direct switch-versus-continue comparison appears in Figure A3 and Table A12. Its simultaneous band covers the five post-intervention checkpoints; the shared-start difference is exactly zero. Machine-readable values for all curves are retained in the reproducibility materials.

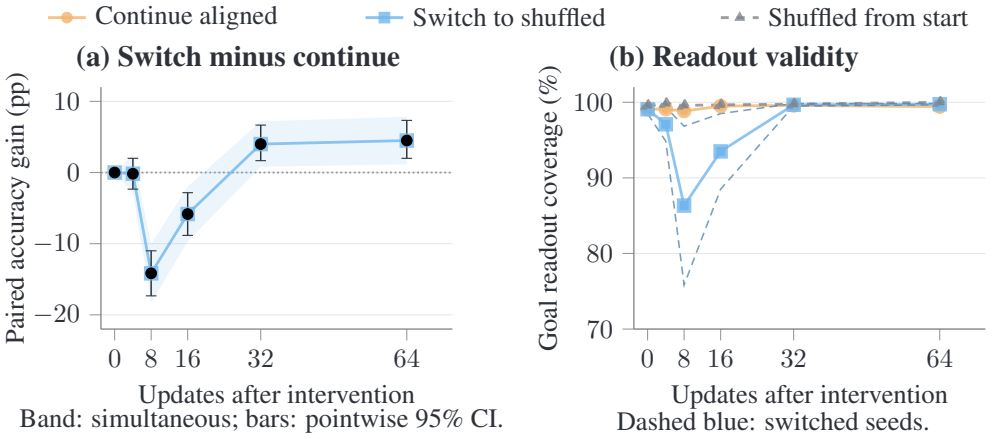


Figure A3: Switch-minus-continue goal accuracy (left, pp) and goal-margin coverage (right, %). Shading shows the simultaneous 95% band.

After 8 and 16 intervention updates, switched-model goal coverage averages 86.33% and 93.50%. The first seed has coverage of 75.83% and 88.50%, whereas the second has 96.83% and 98.50%. These early accuracy dips mix changes in valid readout with changes in judgment and are retained in the curve. The final +4.50-point effect remains positive with website clustering: the pointwise interval is [2.00, 7.07] and the simultaneous band is [1.22, 7.78]. Thus, the observed recovery is delayed rather than immediate or monotonic.

Table A12: Paired two-goal accuracy effects (pp) and uncertainty.

Updates after intervention	Switch minus continue	Pointwise page-cluster 95% CI	Simultaneous 95% band
4	−0.17	[−2.33, 2.00]	[−2.86, 2.53]
8	−14.17	[−17.33, −11.00]	[−18.28, −10.05]
16	−5.83	[−8.83, −2.83]	[−9.69, −1.98]
32	+4.00	[1.67, 6.67]	[0.80, 7.20]
64	+4.50	[2.00, 7.33]	[1.17, 7.83]

A.7 PreAct Reconstruction and Transfer Results

The final PreAct training export replaces only the current candidate rationale spans. Its compatibility audit verifies all 1,024 states and 2,048 order-expanded rows against the original-data version: sample order, actions, labels, historical context, candidate-order permutations, prompt text outside those spans, and verdict targets are unchanged. Generation and review inputs omit historical thoughts, while the original history supplied to the trained PRM remains intact.

The generator receives the task, current page, URLs, prior actions, and the action to explain. It does not receive preference labels, sibling candidates, original rationales, later observations, branch rewards, or benchmark predictions. Both sides use a common 60–90-word budget. The logged API model identifiers are "qwen3.8-max-0902" for generation and "qwen3.7-max-2026-06-08" for factual review. Both calls use temperature 0 and a 1,024-token thinking budget, with output caps of 2,400 and 2,200 tokens. The generator also supplies supporting source quotations in a separate review record; those quotations are not inserted into the PRM training input.

Factual review checks action-target binding, task constraints, current-state claims, action semantics, unsupported assertions, and outcome claims. An additional assistant inspection covered 32 candidate rationales from 16 states. It initially marked six passages as requiring substantive repair and two as optional refinements; eight paragraphs were revised, and all 32 final text hashes were checked. This is a targeted AI-assisted audit, not a randomized error-rate estimate or independent human consensus. The reconstruction combines a common information boundary, a common generator, and factual calibration; the comparison tests this combined procedure.

Table A13: Effect of the same test rationales on WebContrast two-goal accuracy. Means over two seeds; effect = with minus without (pp). CI: 95% page-cluster interval. Qwen: Qwen3-8B; Llama: Llama-3.1-8B-Instruct.

Backbone	Training	Without rat. (%)	With rat. (%)	With−without (pp)	95% page-cluster CI
Qwen	Native	89.00	85.42	−3.58	[−6.33, −0.67]
	PreAct	88.67	94.25	**+5.58**	[2.83, 8.50]
Llama	Native	63.33	39.08	−24.25	[−28.92, −19.58]
	PreAct	74.50	79.83	**+5.33**	[1.92, 8.83]

The no-rationale accuracies in Table A13 are nearly identical for Qwen, but the same explanations hurt Native and help PreAct in the two-seed means. Llama also improves without rationales. Thus, improved rationale use need not coincide with an equally large improvement in no-rationale judgment.

The paired PreAct-minus-Native change in rationale effect is +9.17 [6.00, 12.42] points for Qwen and +29.58 [24.17, 35.00] for Llama. Website-cluster intervals are [6.00, 12.50] and [24.73, 34.58]. The calculation follows A.1, using unrounded page-level predictions; Table A14 reports each seed separately.

All four training interactions in Table A14 are positive, as illustrated in Figure A4, so the mean improvement is not driven by a single seed. Native Qwen's rationale effect is slightly positive in the

Table A14: Per-seed effect of adding WebContrast rationales (pp). Interaction: PreAct effect minus Native effect. Qwen: Qwen3-8B; Llama: Llama-3.1-8B-Instruct.

Backbone	Seed	Native effect (pp)	PreAct effect (pp)	Interaction (pp)
Qwen	20260901	+0.17	+6.50	+6.33
	20260917	−7.33	+4.67	+12.00
Llama	20260901	−24.00	+3.50	+27.50
	20260917	−24.50	+7.17	+31.67

first seed and negative in the second, so harm-to-help describes the two-seed mean, not every run. The test rationales are WebContrast's goal-blind descriptions, not naturally sampled actor thoughts. The PreAct comparison tests information access, generator, and factual review together.

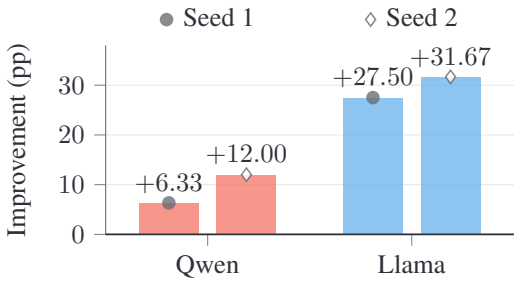


Figure A4: PreAct minus Native rationale effect on WebContrast (pp), shown separately for each seed.

Help and harm decomposition. For each page, rationale version, and seed, let correctness mean that both goals receive the correct choice. Help counts a transition from incorrect without rationales to correct with them; harm counts the reverse. We average these indicators across versions and seeds before averaging pages. Both rates use the full page denominator, not only initially correct or incorrect pages. Their difference recovers the rationale effect, apart from rounding. Unresolved choices keep their original incorrect status. Table A15 gives the decomposition.

Table A15: Help and harm on WebContrast (% of all 300 pages), averaged over two seeds. Net is help minus harm (pp).

Backbone	Training	Help	Harm	Net
Qwen	Native	3.83	7.42	−3.58
	PreAct	7.00	1.42	+5.58
Llama	Native	7.25	31.50	−24.25
	PreAct	10.25	4.92	+5.33

Both models gain some help, but reduced harm contributes more to their net improvement. The harm rate falls from 7.42% to 1.42% for Qwen and from 31.50% to 4.92% for Llama in Table A15. Thus, PreAct chiefly reduces how often rationales overturn an otherwise correct goal pair.

Scoring individual goals gives the same direction for reduced harm. Qwen help/harm rates are 2.67/4.50% after Native training and 3.75/0.75% after PreAct; Llama rates are 9.04/21.04% and 6.21/3.54%. The corresponding net effects are −1.83 versus +3.00 points for Qwen and −12.00 versus +2.67 for Llama. In this scoring view, Llama's help rate also falls, so the common conclusion is reduced explanation-induced error, not uniformly more errors corrected by explanations.

For WebContrast, the flip rates in Table A16 count choice changes only among comparisons with definite choices under both rationale versions. We compute the rate separately for each seed and then average the two rates; resolved coverage ranges from 97.83% to 99.83%. We report the original full-denominator bounds alongside these conditional flip rates and the goal test without current rationales in Table A16. The latter improves for Llama but not for Qwen: the paired differences are +11.17 [7.17, 15.33] and −0.33 [−2.50, 1.83] points. With rationales present, the corresponding gains are +40.75 [36.42, 45.25] and +8.83 [6.17, 11.50]. Thus, improved performance while reading rationales does not establish a uniform gain when they are absent.

Table A16: WebContrast flip statistics (%). No-rat. accuracy: two-goal accuracy without current rationales.

Backbone	Training	Flip rate	Coverage	Flip bounds	No-rat. accuracy
Qwen	Native	10.45	99.75	[10.42, 10.67]	89.00
Qwen	PreAct	1.25	99.83	[1.25, 1.42]	88.67
Llama	Native	44.43	97.83	[43.50, 45.67]	63.33
Llama	PreAct	3.95	99.17	[3.92, 4.75]	74.50

All percentages in Table A16 average the two seeds. The conditional flip rates come from the following confirmed-flip/resolved-comparison counts for seeds 20260901 and 20260917, respectively: Qwen Native, 46/600 and 79/597; Qwen PreAct, 5/598 and 10/600; Llama Native, 228/583 and 294/591; Llama PreAct, 26/593 and 21/597. Each seed has 600 planned comparisons. Paired reductions are 9.20 [6.86, 11.76] points for Qwen and 40.48 [36.27, 44.62] for Llama. Conditional denominators are recomputed within each bootstrap resample.

All Collection views in Table A17 share one holdout and retain the source dataset's chosen/rejected labels. The WebLINX column combines 204 reference-anchored pairs from the test_geo and test_vis splits. Each pairs a demonstrated action with an alternative that targets a different element, and treats the demonstrated action as the reference. These pairs use no candidate rationales: the demonstrations do not provide them, and we do not add them to either candidate. These two evaluations use preference or demonstration labels rather than paired execution rewards.

Table A17: Per-seed Collection and WebLINX accuracy (%); "No rat." omits current candidate rationales.

			Collection			WebLINX
Backbone	Training	Seed	Native	PreAct	No rat.	No rat.
Qwen	Native	20260901	98.63	75.98	77.15	80.64
Qwen	Native	20260917	99.41	70.12	74.61	77.45
Qwen	PreAct	20260901	84.18	84.18	82.03	81.86
Qwen	PreAct	20260917	85.35	82.62	81.25	83.09
Llama	Native	20260901	99.61	67.77	56.25	48.04
Llama	Native	20260917	99.02	68.16	65.23	54.41
Llama	PreAct	20260901	80.08	80.66	77.34	58.82
Llama	PreAct	20260917	77.54	81.25	77.73	58.82

WebShop comparisons use public search trees released with AgentRM (Xia et al., 2025). Candidates share a parent state and have matching recorded continuation suffixes ending in purchase-score receipts. We use these existing receipts rather than execute new agent rollouts. In the original 78-task set, 167 of 283 pairs have equal logged rewards. Excluding those pairs leaves 116 pairs in 47 tasks for the post-hoc discrimination diagnostic reported in Table 6 and Table A18; every model and input view uses the same filter. Selection accuracy measures whether the higher-reward logged branch is chosen. Aggregation proceeds through candidate orders, receipt pairs, decisions, parent

Table A18: PreAct minus Native (pp; paired task-cluster 95% CIs).

Evaluation view	Qwen	Llama
Collection: Native	−14.26 [−18.63, −10.21]	−20.51 [−25.19, −16.01]
Collection: PreAct	+10.35 [5.22, 15.58]	+12.99 [7.67, 18.26]
Collection no rat.	+5.76 [2.23, 9.39]	+16.80 [12.64, 20.95]
WebLINX no rat.	+3.43 [0.61, 6.37]	+7.60 [2.33, 12.75]
WebShop non-tie, natural rat.	+1.07 [−6.12, 8.64]	+18.05 [7.52, 28.81]
WebShop non-tie, no rat.	+1.23 [−4.71, 7.38]	+11.94 [2.46, 21.50]

states, and tasks, with equal final task weights. Both backbones have positive mean differences in both input views, but the paired intervals in Table A18 include zero for Qwen3-8B and exclude zero for Llama. We retain the original set's reward-tied pairs and report conditional branch reward, scaled by 100, in Table A19. It is not full-task online success.

Table A19: Full-set WebShop branch rewards ($\times 100$). Δ: PreAct minus Native; rat.: current candidate rationale.

Backbone	Training	Natural rat.	No rat.
Qwen	Native	53.70	53.73
Qwen	PreAct	53.77	53.77
Qwen	Δ [95% CI]	+0.08 [−0.87, 1.11]	+0.05 [−0.65, 0.76]
Llama	Native	51.42	52.71
Llama	PreAct	53.71	53.80
Llama	Δ [95% CI]	+2.29 [0.73, 4.04]	+1.09 [−0.42, 2.58]

Rationale effects across datasets. We repeat matched with/without-current-rationale comparisons on Collection and WebShop, using their original state or task weights. Table A20 reports the within-model effects and their direct difference between training recipes. On WebShop, Llama's average loss from adding natural rationales decreases from 8.62 to 2.51 points for selection accuracy and from 1.29 to 0.09 for logged reward. The paired intervals quantify uncertainty in these changes: all six between-recipe differences in Table A20 include zero.

Table A20: Rationale effects beyond WebContrast. Collection and WebShop selection effects are accuracy differences (pp); WebShop reward effects use logged conditional reward $\times 100$. Interaction CIs resample tasks.

Backbone / evaluation	Native effect	PreAct effect	Interaction [95% CI]
Qwen / Collection	−2.83	+1.76	+4.59 [−0.68, 9.94]
Qwen / WebShop selection	−0.70	−0.86	−0.16 [−9.47, 9.01]
Qwen / WebShop reward	−0.03	−0.00	+0.03 [−1.20, 1.27]
Llama / Collection	+7.23	+3.42	−3.81 [−8.87, 1.45]
Llama / WebShop selection	−8.62	−2.51	+6.10 [−5.19, 17.66]
Llama / WebShop reward	−1.29	−0.09	+1.20 [−0.25, 2.77]

WebShop selection uses Table 6's non-tie cohort; the reward analysis keeps the full set. Both retain hierarchical aggregation and equal task weights. These are post-hoc analyses of frozen outputs.

The evaluation sets were frozen for regression testing but had been examined during development. The non-tie WebShop view, the conditional flip-rate presentation, and the rationale-effect analysis in Table A13 were examined after earlier results were available; full-set, unresolved-case, and per-seed statistics are therefore retained above. These analyses are not presented as a newly opened

independent confirmation set. All comparisons in this section use verdict SFT; they do not reproduce the complete judgment-rationale distillation and reinforcement-learning pipeline of WebArbiter.

A.8 Neutral-Text Control

Construction and execution. We test the same eight final Native and PreAct models on all 300 WebContrast pages. Neutral paragraphs are assembled from 120 inspected language-and-writing templates, without page facts, candidate actions, outcome evidence, or evaluator instructions. Assignment is deterministic and independent of goals, labels, order, and model identity. Only current rationale spans change; input checks verify all action labels and the remaining prompt text. Two goals, two versions, and two candidate orders give 2,400 inputs per model, or 2,384 after deduplication. The eight evaluation runs comprise 19,072 new requests and pass scoring-integrity, batch-invariance, and replay checks. Rationale and no-rationale conditions reuse frozen outputs, rechecked against Table A13.

Length and scope. We target each existing rationale's whitespace word count and its token counts under both backbones. Across 1,200 text slots, mean absolute differences are 1.30 words, 2.29 Qwen tokens, and 1.71 Llama tokens. A total of 1,073 slots (89.42%) differ by at most five on all three measures. Before neutral-model outputs were available, we fixed a 247-page sensitivity set requiring all four text slots on a page to satisfy that rule; all 300 pages remain the primary analysis. The bank yields 751 distinct neutral paragraphs and preserves identical original versions. Its topics are not balanced: 59.4% of slots concern punctuation. This is a control for these task-unrelated texts, not a representative sample of every possible irrelevant input or a pure isolation of semantic content. No neutral-text evaluation is added to the controlled Aligned/Reversed/Shuffled models.

Per-seed results and completeness. Table A21 retains all eight runs. Neutral-text format coverage is 99.58–100%; all ten invalid or truncated raw outputs occur in the first Native Llama seed and remain in the denominator. Definite goal-choice coverage is 96.50–100% under neutral text, 96.83–100% without rationales, and 98.58–100% with rationales. Unresolved margins and invalid outputs count as incorrect, using the same frozen rule in all conditions. The analysis verifies all raw-output hashes and rederives page-level decisions. These are new inference conditions on an existing development benchmark, not new independent pages.

Table A21: Per-seed two-goal accuracy (%) for the neutral-text control. Rationale uses the existing WebContrast explanations.

Backbone	Training	Seed	No rationale	Neutral text	Rationale
Qwen	Native	20260901	88.67	79.50	88.83
	Native	20260917	89.33	69.17	82.00
	PreAct	20260901	88.00	86.50	94.50
	PreAct	20260917	89.33	87.33	94.00
Llama	Native	20260901	66.00	40.33	42.00
	Native	20260917	60.67	25.50	36.17
	PreAct	20260901	74.33	72.50	77.83
	PreAct	20260917	74.67	71.67	81.83

A.9 Reproducibility and Resource Scope

The reproducibility materials include conversion and evaluation code, model and generator identifiers, training configurations, split and sample indices, text-version hashes, raw prediction receipts, and the data behind every table and curve. Figure source files retain editable vector versions. Distinct generations of PreAct text and distinct inference backends are indexed separately, so an earlier semantic review or score is not silently transferred to a later version.

The planned release contains our code, task indices, hashes, result tables, and derived annotations whose redistribution is permitted. Upstream raw trajectories remain local where redistribution rights are unresolved. WebContrast contains 300 pages. Its factual checks, development history, and

rationale versions accompany the release so that users can distinguish behavioral diagnosis from independent deployment evaluation.