REPLAYBIND: TRAINING LANGUAGE WORLD MODELS TO GENERATE ACTION-FAITHFUL FUTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

An agent deciding what to do next needs to distinguish the consequences of its alternatives. In language world models, however, the change that matters may occupy only a few words in an otherwise unchanged observation. Predicting that observation well can therefore leave the association between an action and its consequence weakly learned. We introduce REPLAYBIND, which learns this association from two observed branches sharing a visible state. Its crossed-likelihood objective favors the recorded action–outcome assignment over the assignment obtained by exchanging the outcomes, using forward prediction and inverse action reconstruction within one generator. Broad observation restoration and balanced branch replay carry this relational supervision into complete observation generation. Matched comparisons show improvements in action–outcome assignment, action reconstruction, and generated branch separation. These results show that REPLAYBIND better preserves action-induced changes across successive predictions, providing more reliable descriptions of future outcomes for distinguishing the consequences of different actions.

1 INTRODUCTION

An agent considering its next move needs to anticipate how each available action would change its surroundings. Language world models express these possible futures as text, where a consequential change may occupy only a few words in an otherwise familiar scene. Opening a refrigerator exposes its contents while leaving most of the kitchen unchanged. A prediction can preserve much of the scene yet miss the effect that makes the action useful. Capturing these differences is essential to making predicted futures informative for action choice.

Ordinary trajectory training learns from one action and its observed outcome at each step. The prediction target describes both the action's effects and the environment's persistent content. Likelihood training supervises both, but extensive shared text can make observation similarity insensitive to the small changes that distinguish alternative actions. Separately learning each transition also leaves the correspondence between competing actions and futures implicit. An explicit comparison can focus supervision on the differences an agent needs when considering its alternatives.

Two observed branches from the same visible state supply that comparison. Alongside two generation targets, they provide a recorded assignment between actions and outcomes. Exchanging the outcomes creates a competing assignment using the same actions and the same future texts. Their likelihood difference measures which pairing the model prefers; additive preferences for an action or an outcome alone cancel. This gives the generator a direct learning signal for which future belongs to which action, while retaining the observed outcomes as its generation targets.

We build REPLAYBIND around this assignment. Crossed forward likelihoods train the model to predict each action's outcome, while crossed inverse likelihoods train the same model to recover the action that explains it. Branch pairs emphasize local alternatives; ordinary trajectories supply the broader context needed to generate complete observations. We therefore follow binding with observation restoration in the model's native conversation format, then replay branches alongside broad transitions to retain action-dependent variation. Figure 1 shows how these stages bring relational supervision into the generator used for one-step and recursive prediction.

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.

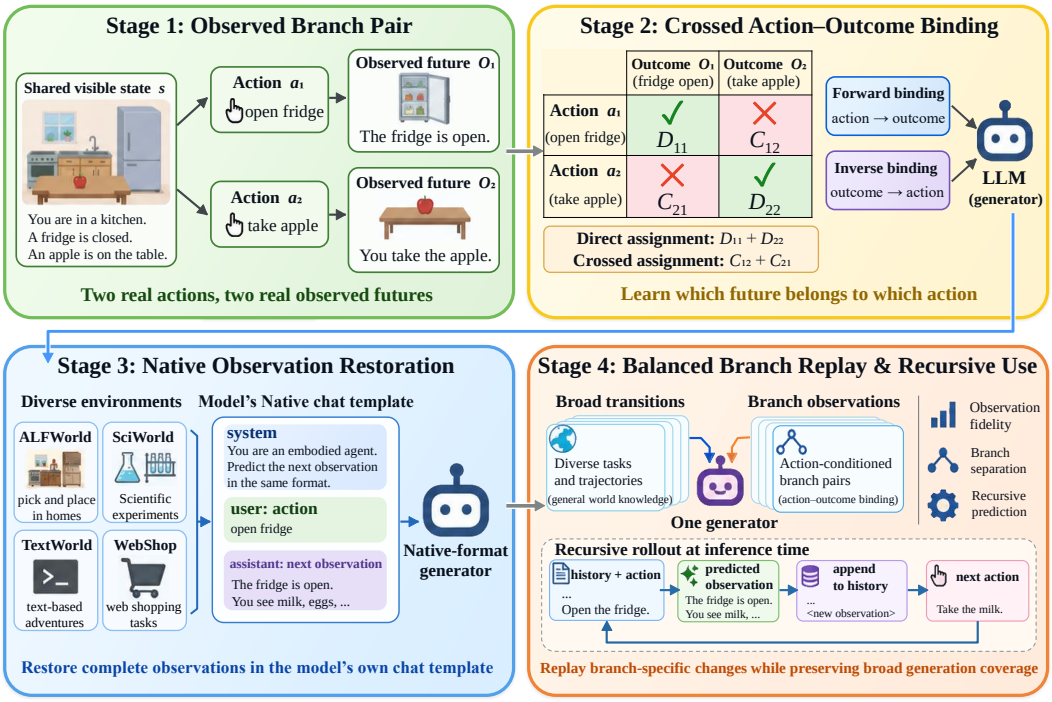


Figure 1: **Overview of REPLAYBIND.** Observed branches supply two actions and their futures from a shared state. Forward and inverse crossed assignments train one generator to bind each action to its outcome. Native observation restoration and balanced branch replay connect this relation to complete observation generation and recursive use. State illustrations and prompt text are schematic.

Our experiments examine whether stronger assignments translate into more informative predictions. Matched loss and directional comparisons connect crossed supervision to better action–outcome assignment and action reconstruction. Analysis of generated observations also shows improved separation of action outcomes; Figure 5 illustrates the recovery of an accepted command's effect from a shared failure response. Across environments, the complete procedure improves aggregate observation fidelity over the evaluated external models and retains this advantage as its own predictions accumulate in the history.

Our contributions are:

- We formulate action–outcome binding as an assignment of observed branches, using crossed likelihoods to supervise the relation between actions and futures within one generator.
- We develop REPLAYBIND, coupling forward and inverse binding with observation restoration and balanced replay to generate complete observations that preserve action-dependent changes.
- We isolate the effect of crossed supervision through matched comparisons and evaluate action reconstruction, generated branch separation, and the complete model's fidelity through recursive prediction.

2 RELATED WORK

World models and language dynamics. World models support planning or policy learning through predicted dynamics (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; Schrittwieser et al., 2020; Micheli et al., 2022; Hansen et al., 2024). Language-conditioned variants predict multimodal futures or transfer embodied experience (Lin et al., 2023; Xiang et al., 2023); recent web and general-agent models use imagined outcomes, broad trajectory collections, self-evolution, or test-

time memory (Chae et al., 2025; Gu et al., 2024; Xiao et al., 2026; Zuo et al., 2026; Li et al., 2026b; Fang et al., 2025; Zhang et al., 2026). These systems make textual future prediction a practical interface for agent planning. REPLAYBIND develops the corresponding training signal for associating each predicted future with its generating action.

Interactive language agents and evaluation. ALFWorld, WebArena, and AgentBench provide reproducible embodied, web, and cross-domain interaction settings (Shridhar et al., 2020; Zhou et al., 2024; Liu et al., 2024); ReAct couples reasoning with action (Yao et al., 2022), while API-Bank and BFCL evaluate executable tool use and function calling (Li et al., 2023; Patil et al., 2025). Together, these environments motivate transition models whose outputs remain grounded in executable actions and observable effects.

Learning signals beyond next-observation prediction. AAWM constructs decision-relevant targets from retrieved evidence (Cai et al., 2026); RWML optimizes an embedding-space reward (Yu et al., 2026); DWM learns from executed branches with a coupled ranker (Li et al., 2026a); and UI-Oceanus studies multiple dynamics directions at GUI scale (Wu et al., 2026). REPLAYBIND complements this line by applying the relational signal directly to the sequence likelihood of observed transitions. Forward and inverse views share the same causal model, so the learned relation is expressed through the generator used for rollout.

Behavioral evaluation of dynamics. WorldTest advocates behavioral evaluation of learned dynamics (Warrier et al., 2025). We follow this perspective across increasingly demanding uses of the transition model: direct next-observation fidelity, recursive free-running simulation, and the association between actions and their observed consequences.

3 METHOD

3.1 OBSERVATIONS, ACTIONS, AND TRANSITION VIEWS

At interaction step t, let $h_t = (c, o_{\le t}, a_{<t})$ contain the task context c, observations through t, and actions before t. The model $p_\theta(o_{t+1} \mid h_t, a_t)$, with parameters θ, predicts the next observation after action a_t. For a branch transition, write $s = o_t$, $a = a_t$, and $o = o_{t+1}$. Branch training uses the shared visible state s; broad transition training also supplies the recorded history.

Let $o = (w_1, \ldots, w_L)$ contain L observation tokens and $a = (v_1, \ldots, v_K)$ contain K action tokens. The token index j runs over the target, and $w_{<j}$ or $v_{<j}$ denotes its preceding tokens. The forward and inverse token-average negative log likelihoods are

$$\ell_f(s, a, o) = -\frac{1}{L} \sum_{j=1}^{L} \log p_\theta(w_j \mid s, a, w_{<j}), \tag{1}$$

$$\ell_i(s, o, a) = -\frac{1}{K} \sum_{j=1}^{K} \log p_\theta(v_j \mid s, o, v_{<j}). \tag{2}$$

Subscripts f and i denote forward prediction and inverse action reconstruction. Alternating these views trains the same model to generate a consequence and recover the action that caused it.

3.2 CROSSED ACTION–OUTCOME ASSIGNMENTS

An observed branch pair contains a common visible state s and two distinct action–outcome pairs (a_1, o_1) and (a_2, o_2). Branches may come from independently replayed environment states or recorded transitions with identical normalized observations. Both outcomes are observed. Here, subscripts label complete branches, and each sequence log likelihood sums its target-token log probabilities. The direct cost D_f uses recorded pairings; the crossed cost C_f exchanges their outcomes:

$$D_f = -\log p_\theta(o_1 \mid s, a_1) - \log p_\theta(o_2 \mid s, a_2), \tag{3}$$

$$C_f = -\log p_\theta(o_1 \mid s, a_2) - \log p_\theta(o_2 \mid s, a_1). \tag{4}$$

Let L_b be the scored token count of outcome o_b for branch $b \in \{1, 2\}$. With $T_f = L_1 + L_2$, the forward margin and loss are

$$\Delta_f = \frac{C_f - D_f}{T_f}, \qquad \mathcal{L}_f = \frac{\ell_f(s, a_1, o_1) + \ell_f(s, a_2, o_2)}{2} + \beta[m - \Delta_f]_+. \tag{5}$$

Here $[x]_+ = \max(x, 0)$ is the positive part, m is the desired margin, and β weights the hinge penalty. Positive Δ_f favors the recorded assignment. The inverse costs D_i, C_i analogously compare recorded and exchanged action targets given the outcomes. With K_b scored tokens in action a_b, $T_i = K_1 + K_2$ and $\Delta_i = (C_i - D_i)/T_i$. The inverse loss $\mathcal{L}_i$ combines the mean of the two ℓ_i terms with the same hinge on Δ_i, as expanded in Eq. 9.

Each outcome and each conditioning action occurs once in both assignments. Additive outcome-only or action-only sequence costs therefore cancel in their difference: improving the margin requires changing the relative compatibility of action–outcome combinations.

The two loss terms serve complementary roles. Positive likelihood rewards faithful target text even after the assignment is correct; the hinge adds gradients when the margin is too small. All target tokens participate in both terms. Action-argument redaction is used only as a later scoring diagnostic.

3.3 Native observation restoration

Observed branches emphasize local alternatives. Restoration broadens coverage using a distribution $\mathcal{D}_x$ of recorded transitions (h, a, o), where h is the history preceding action a and outcome o. Let $\bar{o}$ append the native assistant end-of-turn token to o, with length $|\bar{o}|$ and token $\bar{o}_j$. The restoration loss is

$$\mathcal{L}_{\text{restore}} = \mathbb{E}_{(h,a,o)\sim\mathcal{D}_x}\left[-\frac{1}{|\bar{o}|}\sum_{j=1}^{|\bar{o}|}\log p_\theta(\bar{o}_j \mid h, a, \bar{o}_{<j})\right]. \tag{6}$$

Here $\mathbb{E}$ averages over sampled transitions, and $\bar{o}_{<j}$ is the preceding target-token prefix. The native template supplies role and observation boundaries, aligning training with recursive use. Recorded histories also supply earlier actions and observations absent from state-only pairs.

3.4 Balanced supervised branch replay

After restoration, let $\mathcal{D}_b$ denote the distribution of positive branch transitions (s, a, o). Write ℓ_x for the history-conditioned token-average loss inside Eq. 6, and ℓ_b for its state-conditioned counterpart. With branch-mixture weight λ, the continuation loss is

$$\mathcal{L}_{\text{branch}} = (1 - \lambda)\,\mathbb{E}_{\mathcal{D}_x}[\ell_x] + \lambda\,\mathbb{E}_{\mathcal{D}_b}[\ell_b], \qquad 0 \le \lambda \le 1. \tag{7}$$

Each expectation averages over its indicated distribution. Both pools retain complete observation targets up to the training length limit. Branch replay supplies action-dependent consequences, while broad transitions preserve observation structure and longer histories. This stage uses positive likelihood, separating earlier relational supervision from later generation supervision.

3.5 One-step and recursive generation

At inference, the model generates the next observation using its native chat template. One-step fidelity uses the recorded conversation prefix. Branch prediction instead provides the common visible state and one of its recorded actions. In recursive prediction, the generated observation is appended as the next assistant turn, followed by the next fixed logged action. Every later prediction thus conditions on the model's own earlier observations. The model supplies environment text throughout this process; the action sequence is fixed by the recorded trajectory.

4 Experimental Setup

Models and native interfaces. We start from Word2World-7B and fine-tune it using low-rank adaptation (LoRA). The primary generation comparison evaluates models with and without crossed supervision, sharing initialization, examples, ordering, computation, optimization, restoration, and

Table 1: Native-template observation fidelity. Token F1 on 761 state-deduplicated transitions, with equal environment weighting. All models use their own chat template. Bold marks the best score and underline the next distinct score.

Model	ALFWorld	SciWorld	TextWorld	WebShop	Mean
Word2World-7B	**0.9969**	0.2711	0.0014	0.0678	0.3343
WMA-8B	0.5344	0.3465	0.3808	0.5687	0.4576
WebWorld-8B	0.1267	0.1102	0.2168	0.4059	0.2149
Qwen2.5-7B	0.5698	0.5353	0.4658	0.5926	0.5409
Qwen3-30B-A3B	0.6325	<u>0.6567</u>	<u>0.4911</u>	<u>0.6294</u>	<u>0.6024</u>
REPLAYBIND	<u>0.9666</u>	**0.8025**	**0.7839**	**0.6530**	**0.8015**

replay. An extended-context comparison adds forward-only crossed, inverse-only crossed, and row-wise ranking controls; each family has its own matched model without the crossed term. External models are Word2World-7B, WMA-8B, WebWorld-8B, Qwen2.5-7B, and Qwen3-30B-A3B (Yang et al., 2025). Every model uses its own chat template. Inputs are limited to 4,096 tokens. Appendices A and I give the LoRA configuration, loss weights, and details of both training settings.

Data. Broad fidelity uses 761 transitions: 320 ALFWorld, 187 SciWorld, 145 TextWorld, and 109 WebShop. They are obtained from the original 822-transition suite by excluding all 61 observations matching the new branch-training states, with the same exclusion applied to every model. Branch testing contains 134 ALFWorld, 156 SciWorld, 15 TextWorld, and 90 WebShop pairs. ALFWorld branches are replay-verified; the other test branches revisit the same visible text within a recorded conversation. The 15 TextWorld pairs come from two conversations. A separate TextWorld holdout supplies 200 unused source-ID groups from the released training pool, with known adapter-training groups and matching observations excluded. These groups can contain multiple logged records. Appendices B and J describe cohort provenance and candidate construction.

Metrics. Token F1 measures normalized token-multiset overlap. Broad fidelity averages domains equally. Likelihood evaluation reports Δ_f, Δ_i, and correct joint assignments. Action reconstruction reports normalized exact match, using either one recorded reference or all logged actions for the same visible transition. The multi-candidate cohort has 400 source groups and 1,523 transitions across four environments; a separate 374-group cohort supplies 1,085 new-source transitions. For branch $j \in \{1, 2\}$, let $\hat{o}_j$ be the generated observation, o_j its reference, and o_{3-j} the other branch's reference. With token-overlap score F_1, define

$$\delta_j = F_1(\hat{o}_j, o_j) - F_1(\hat{o}_j, o_{3-j}), \qquad M = \frac{\delta_1 + \delta_2}{2}, \qquad B_\tau = \mathbf{1}\{\min(\delta_1, \delta_2) > \tau\}. \tag{8}$$

Here δ_j measures preference for the correct outcome, and M averages both preferences. The indicator $\mathbf{1}\{\cdot\}$ is one when its condition holds and zero otherwise; $\tau \geq 0$ is the required separation. Thus B_0 requires both predictions to favor their own outcome. These scores assess generated text rather than teacher-forced likelihoods. Recursive evaluation uses 328 ten-step trajectories from SciWorld, TextWorld, and WebShop, averaging within each domain and then equally across domains. Within each matched comparison, objective arms use the same fixed cohort and stated training configuration.

5 RESULTS

5.1 OBSERVATION FIDELITY ACROSS ENVIRONMENTS

Token F1 measures how much of the reference observation the prediction recovers while penalizing extra content. Averaging environments equally prevents a larger cohort from dominating this measure of coverage. On 761 transitions, REPLAYBIND reaches 0.8015 versus 0.6024 for Qwen3-30B-A3B, the strongest external model overall (Table 1). Its gains in SciWorld, TextWorld, and WebShop show that the complete procedure preserves observation structure beyond the initialization's specialized environment.

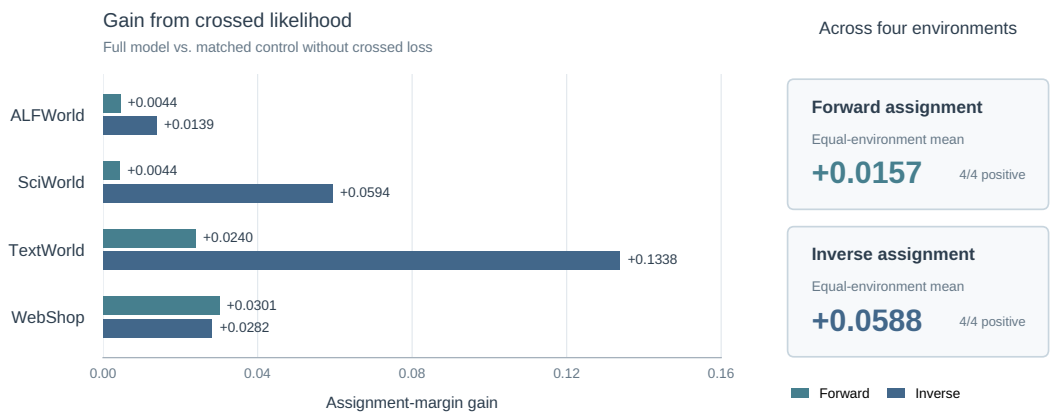


Figure 2: **Crossed-likelihood loss ablation.** Assignment-margin gains of the complete model over its matched control without crossed loss. Bars show forward and inverse gains by environment; cards give equal-environment means. Both models share restoration and replay.

Table 2: Crossed-loss ablation on the additional TextWorld holdout. One pair from each of 200 source groups excluded from known adapter-training sources. Margin and both-correct evaluate generated observations; forward and inverse are teacher-forced assignment margins. Dashes indicate unmeasured likelihoods for Qwen3.

Model	Token F1	Margin	Both correct	Forward	Inverse
Qwen3-30B-A3B	0.2605	0.0176	0.1000	—	—
Without crossed loss	0.7416	0.3011	0.5750	0.2197	0.9139
REPLAYBIND	**0.7482**	**0.3226**	**0.6100**	**0.2379**	**0.9982**

This coverage matters for action-conditioned prediction: a useful future must describe both the action's effect and the surrounding state that persists. Excluding local transitions found in any known adapter-training source retains an aggregate advantage, 0.7796 versus 0.5840 on 536 transitions. We next isolate the contribution of crossed supervision using matched comparisons.

5.2 CROSSED-LIKELIHOOD LOSS ABLATION

The loss ablation changes only the crossed-assignment hinge. Both arms compute the same four combinations per pair and share positive forward and inverse likelihoods, data, restoration, and replay. An assignment margin measures preference for the recorded action–outcome pairing over the exchanged pairing, normalized by scored target length. A larger margin therefore indicates stronger relational discrimination, even when both models already choose the correct pairing.

Figure 2 shows positive gains in both directions in all four environments after complete training. These gains survive restoration and replay, which use positive likelihood alone. The earlier relational signal thus remains expressed in the final generator rather than only at the binding checkpoint. In WebShop, the final inverse assignment accuracy also rises from 0.8444 to 0.8778, connecting a larger margin to more correct discrete assignments (Appendix C).

On the separate 200-group TextWorld holdout, crossed supervision improves observation F1, branch margin, and both-correct rate together (Table 2). The branch margin compares a prediction with its own and the alternative outcome; both-correct requires a strictly positive preference for each branch. Its increase from 0.5750 to 0.6100 therefore reflects more pairs with two correctly differentiated futures. Copying the same response for both actions cannot satisfy this criterion. This connects the likelihood objective to action-dependent generation.

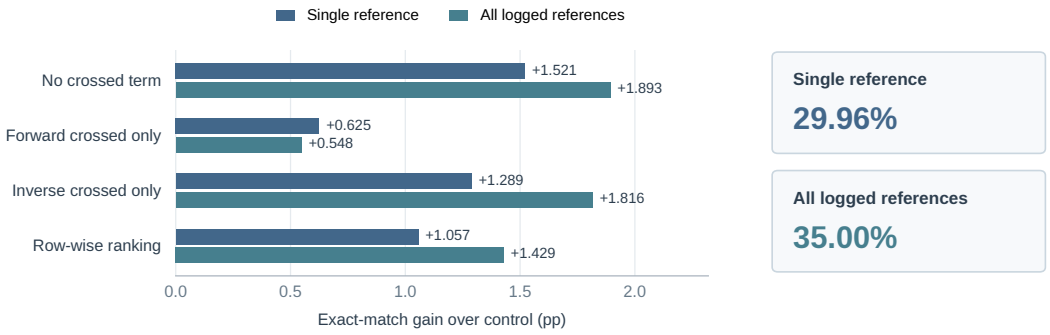


Figure 3: **Action reconstruction under directional and ranking controls.** Bars show the bidirectional model's exact-match gain over each matched control, in percentage points. Both reference rules score the same answers on 1,523 transitions; cards show its environment-average absolute scores.

5.3 Directional supervision and action reconstruction

Action exact match requires recovery of a complete recorded command from the before-and-after observations. It tests whether the predicted cause identifies the action, including its arguments. A single reference accepts one representative command; logged-reference scoring also accepts other commands observed for the same visible transition. Both rules score the same generated answers.

Figure 3 shows that the bidirectional objective improves reconstruction over each directional removal and row-wise ranking. Its gain over ranking persists when additional logged explanations are accepted, so the advantage is not tied to matching only one representative string. Forward supervision is also useful for this inverse task: learning which outcome follows an action complements learning which action explains the outcome. The directional controls support training both views within the shared generator.

The same comparison favors joint assignment over ranking each condition independently. A joint assignment must account for all supplied alternatives, using every target once. The observed advantage is consistent with learning this relation across candidates. The new-source comparison extends the assignment gains to additional records, as detailed in Appendix M.

5.4 Generated branch separation across environments

Generated branch margin subtracts similarity to the other outcome from similarity to the correct outcome. Shared scene text contributes to both comparisons, making this score more sensitive to action-dependent differences than observation F1 alone. ReplayBind's margins reach 0.1907 on ALFWorld and 0.2735 on WebShop, versus 0.1048 and 0.0740 for Qwen3. The separate TextWorld holdout likewise shows stronger separation (Table 2); Table 7 gives the full four-environment comparison.

Requiring both branches to exceed a margin of 0.10 tests whether their preferences extend beyond near ties. ReplayBind satisfies this condition on 0.4627 of ALFWorld pairs and 0.5333 of WebShop pairs, versus 0.3358 and 0.0111 for Qwen3. The particularly large WebShop difference indicates that its mean-margin gain accompanies stronger preferences on both sides of a pair, rather than improvement on only one action.

5.5 Ten-step recursive prediction

Recursive prediction tests whether observation fidelity persists when earlier predictions become later inputs. Each model follows the same logged actions without resetting intermediate observations to their references. ReplayBind maintains an aggregate advantage at every horizon, reaching token F1 of 0.3962 at step ten versus 0.2657 for Qwen3 (Figure 8). Its all-horizon mean is 0.4943 versus 0.2776. This sustained separation indicates that the generated history continues to support subsequent predictions as errors accumulate.

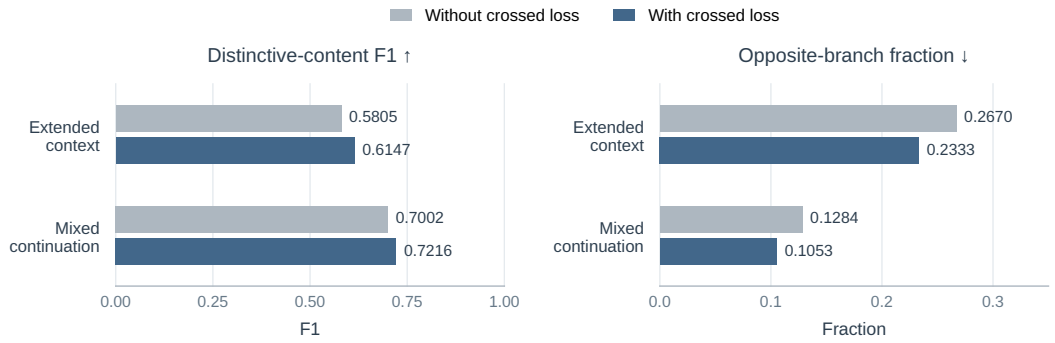


Figure 4: **Recovering branch-specific content.** Two matched training settings on 200 TextWorld source groups. Left: F1 after removing shared reference tokens, higher is better. Right: the predicted residual fraction belonging to the other branch, lower is better. Each setting compares crossed supervision with its own matched control.

6 Analysis

6.1 Complementary roles of restoration and replay

Broad observation fidelity and branch separation probe different demands on the generator. Restoration supplies full histories and diverse observation structure, raising macro token F1 from 0.6808 after binding to 0.7894. Balanced replay then reaches 0.8015 while strengthening generated separation on ALFWorld, TextWorld, and WebShop (Table 14).

For WebShop, replay raises both-correct rate from 0.2778 to 0.6556. Because this criterion requires each prediction to favor its own outcome, the improvement reflects retention of differences between actions alongside broad observation coverage. These stage-wise measurements support the role of branch examples during continued observation training; the matched loss ablation separately identifies the contribution of crossed supervision.

6.2 Recovering content that distinguishes the actions

Whole-observation overlap can reward text shared by both futures. Distinctive-content F1 removes that shared reference text before measuring recovery of the remaining branch-specific tokens. The opposite-branch fraction asks a complementary question: how much of the predicted residual belongs to the other outcome? Higher distinctive F1 and a lower opposite-branch fraction jointly indicate better recovery of the intended changes with less mixing of alternative consequences.

Figure 4 shows this pattern in two matched training settings on the 200-group TextWorld cohort. Extended-context training separates binding from observation training; mixed continuation combines their supervision. In extended-context training, distinctive F1 rises from 0.5805 to 0.6147 while the opposite-branch fraction falls from 0.2670 to 0.2333. Mixed continuation shows the same pattern. Precision and recall both improve, linking the gain to more correct distinctive content rather than merely producing less text. Shared-reference subtraction is applied during scoring; the model still generates complete observations.

Signed-edit F1 additionally compares additions and deletions relative to the starting observation. REPLAYBIND improves combined edit F1 over Qwen3 on ALFWorld and WebShop (Figure 9). Together, these diagnostics connect better observation fidelity to the content changed by an action and to keeping alternative outcomes distinct.

6.3 Case study: similar commands, distinct consequences

Figure 5 makes this distinction visible. From the same kitchen observation, “put bell pepper on stove” succeeds, whereas “place bell pepper on stove” is rejected. The matched zero-crossed model predicts the same failure for both. With crossed supervision, REPLAYBIND restores the placement

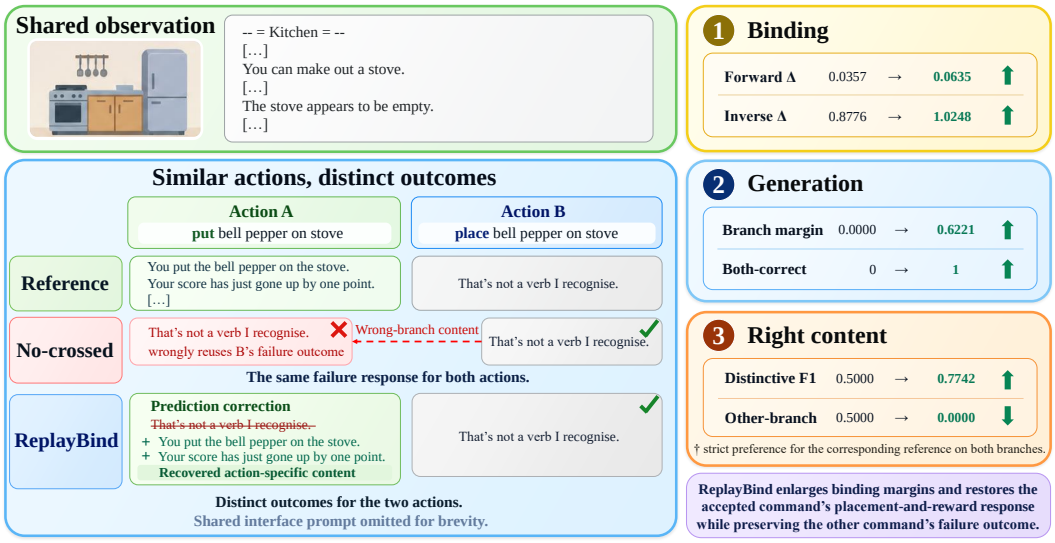


Figure 5: **TextWorld case: binding similar commands to different outcomes.** Matched extended-context generation models with and without crossed supervision receive the same visible state. Green marks recovered content; red strikethrough marks baseline error removed in the output comparison. Scores run from No-crossed to REPLAYBIND. Both-correct requires strict preference for each branch's own reference; Other-branch is the opposite-branch token fraction. Text is excerpted; the kitchen image is illustrative.

and reward response while retaining the other command's failure. Thus the correction respects the environment's recorded command behavior, even though the two commands are similar in ordinary language.

This behavior also motivates using outcome likelihoods for action choice. On the logged TextWorld goal cohort, the progress-selection advantage remains after explicit reward phrases are removed: 62.12% versus 54.55%. The score counts choices whose recorded outcome advances the task. It links the learned relation to selecting useful consequences from supplied alternatives (Appendix R).

7 CONCLUSION

REPLAYBIND combines crossed assignments, native restoration, and branch replay to generate action-faithful futures. Matched controls connect bidirectional supervision to action reconstruction, generated TextWorld separation, and stronger pairing on new sources. Offline choices link outcome likelihoods to recorded task progress. The complete procedure improves aggregate observation fidelity over external models through ten recursive transitions, connecting relational supervision to the identification and generation of action consequences.

AI USE STATEMENT

AI tools assisted with manuscript drafting, polishing, and translation, and with literature search, discovery, and preliminary organization. They supported research discussions and feedback, experimental coding and debugging, experiment execution, results organization, and synthetic data generation and formatting. Their suggestions and implementation references were adopted and refined according to research objectives and experimental evidence. The authors take responsibility for the paper's consistency with experimental results and for its final content and conclusions.

REFERENCES

Guangfeng Cai, Kaibing Yang, Shuo He, Yu Li, Shengtian Yang, Jiaqi Lv, and Lei Feng. Beyond next-observation prediction: Agent-authored world modeling for sequential decision making. *arXiv preprint arXiv:2606.25421*, 2026.

Hyungjoo Chae, Namyoung Kim, Kai Ong, Minju Gwak, Gwanwoo Song, Jihoon Kim, Sunghwan Kim, Dongha Lee, and Jinyoung Yeo. Web agents with world models: Learning and leveraging environment dynamics in web navigation. In *International Conference on Learning Representations*, volume 2025, pp. 63707–63738, 2025.

Tianqing Fang, Hongming Zhang, Zhisong Zhang, Kaixin Ma, Wenhao Yu, Haitao Mi, and Dong Yu. Webevolver: Enhancing web agent self-improvement with co-evolving world model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8970–8986, 2025.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, et al. Is your llm secretly a world model of the internet? model-based planning for web agents. *arXiv preprint arXiv:2411.06559*, 2024.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2(3):440, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Kelvin Li, Dhruv Pendharkar, Anish Pahilajani, Chuyi Shang, Leon Oks, Leonid Karlinsky, Rogerio Feris, Trevor Darrell, and Roei Herzig. Discriminative world models for web agents. *arXiv preprint arXiv:2609.02885*, 2026a.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 3102–3116, 2023.

Yixia Li, Hongru Wang, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Xiaoteng Ma, Guanhua Chen, and Heng Ji. From word to world: Can large language models be implicit text-based world models? In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8084–8111, 2026b.

Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to model the world with language. *arXiv preprint arXiv:2308.01399*, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Archana Warrier, Dat Nguyen, Michelangelo Naim, Moksh Jain, Yichao Liang, Karen Schroeder, Cambridge Yang, Joshua B Tenenbaum, Sebastian Vollmer, Kevin Ellis, et al. Benchmarking world-model learning with environment-level queries. *arXiv preprint arXiv:2510.19788*, 2025.

Mengzhou Wu, Yuzhe Guo, Yuan Cao, Haochuan Lu, Songhe Zhu, Pingzhe Qu, Xin Chen, Kang Qin, Zhongpu Wang, Xiaode Zhang, et al. Ui-oceanus: Scaling gui agents with synthetic environmental dynamics. *arXiv preprint arXiv:2604.02345*, 2026.

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36:75392–75412, 2023.

Zikai Xiao, Jianhong Tu, Chuhang Zou, Yuxin Zuo, Zhi Li, Peng Wang, Bowen Yu, Fei Huang, Junyang Lin, and Zuozhu Liu. Webworld: A large-scale world model for web agent training. *arXiv preprint arXiv:2602.14721*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Xiao Yu, Baolin Peng, Ruize Xu, Yelong Shen, Pengcheng He, Suman Nath, Nikhil Singh, Jiangfeng Gao, and Zhou Yu. Reinforcement world model learning for llm-based agents. *arXiv preprint arXiv:2602.05842*, 2026.

Xuan Zhang, Wenxuan Zhang, See-Kiong Ng, and Yang Deng. Self-evolving world models for llm agent planning. *arXiv preprint arXiv:2606.30639*, 2026.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

Yuxin Zuo, Zikai Xiao, Li Sheng, Fei Huang, Jianhong Tu, Yuxuan Liu, Tianyi Tang, Xiaomeng Hu, Yang Su, Qingfeng Lan, et al. Qwen-agentworld: Language world models for general agents. *arXiv preprint arXiv:2606.24597*, 2026.

A Implementation Details

Shared initialization. The base model is the released ALFWorld-specialized Word2World checkpoint built on Qwen2.5-7B. Initial adaptation used forward observation prediction, inverse action reconstruction, and action-counterfactual hinge supervision. Its corpus contains 10,000 positive transitions per domain from ALFWorld, SciWorld, TextWorld, WebShop, and StableToolBench, each with a counterfactual action. The initial adaptation uses 3,000 updates at learning rate 10^{-5}, a 1,000-update positive forward continuation at 3×10^{-6}, and 500 further bidirectional updates at 10^{-6}. Both arms inherit this entire initialization. The ablation therefore isolates the additional assignment of two observed outcomes; earlier action-conditioning supervision is shared.

Adapters and optimization. The base model is frozen. LoRA adapters target the query, key, value, and output projections of attention layers, with rank 32, adapter scaling coefficient $\alpha = 64$, and dropout 0.05. The paired coefficient settings share model initialization, input order, dropout seed, optimizer, update count, and forward computation. The restoration and replay stages also use matching schedules. Table 3 gives the configurations for each training stage.

Table 3: Training-stage configurations. Binding alternates forward and inverse updates, using one pair per environment and four conditions per pair. The positive-branch row counts the two direct conditions; in inverse updates their targets are actions. Restoration uses two broad observations per environment. Replay combines one broad observation per environment with two branch pairs, rotating domains.

Setting	Binding	Restoration	Balanced replay
Updates	600	400	200
Learning rate	5×10^{-6}	3×10^{-6}	3×10^{-6}
Broad observations per update	0	8	4
Positive branch observations per update	8	0	4

Binding evaluates eight additional crossed conditions per update, with assignment weight $\beta = 0.5$ ($\beta = 0$ in the matched control) and margin $m = 0.25$; restoration and replay use only direct conditions with $\beta = 0$.

The restoration pool contains 4,000 selected conversation transitions per environment. Each source record contributes one transition selected by a fixed rule, and candidates follow a fixed ordering before selection. Exact local-transition matches against the development branches, test branches, and broad fidelity suite are excluded from this pool. During restoration, microbatches of two accumulate to eight observations per update. The final 200 updates retain the same optimizer and mix broad observations with positive branch observations. The 400-update restoration checkpoint exposes the behavior before the final mixture, while the 600-update checkpoint defines the complete model.

Inverse assignment and token counts. For the two recorded branches, K_b is the number of scored tokens in action a_b, and $T_i = K_1 + K_2$. The inverse direct cost D_i reconstructs each recorded action from its outcome; the crossed cost C_i swaps those action targets. With the same hinge weight β and desired margin m as the forward objective,

$$\begin{aligned} D_i &= -\log p_\theta(a_1 \mid s, o_1) - \log p_\theta(a_2 \mid s, o_2), \\ C_i &= -\log p_\theta(a_2 \mid s, o_1) - \log p_\theta(a_1 \mid s, o_2), \\ \Delta_i &= (C_i - D_i)/T_i, \\ \mathcal{L}_i &= \frac{\ell_i(s, o_1, a_1) + \ell_i(s, o_2, a_2)}{2} + \beta[m - \Delta_i]_+. \end{aligned} \tag{9}$$

Sequence costs sum token negative log likelihoods, whereas each positive loss averages its own target tokens. A positive inverse margin favors the recorded assignment. Token counts refer to scored targets after native encoding.

Native encoding and likelihood scoring. Each model's own tokenizer and chat template encode its conversation. For adapters, the accompanying template agrees with its base model's template. The native generation prefix is preserved when encoding a supervised continuation. If tokenization merges a template-final newline with a target-initial newline, the continuation is encoded at the fixed generation boundary. A target exceeding its training budget is truncated and closed with the native end token. Only target positions contribute to the likelihood.

Teacher-forced branch scoring evaluates all four assignments in each direction. The reported margin is the summed crossed negative log likelihood minus the summed direct negative log likelihood, divided by the number of tokens in the two direct targets. A positive margin selects the observed joint assignment. Forward and inverse margins use their own token counts and are reported separately. These are conditional likelihood measurements, distinct from the generated observations.

Forward instruction. Table 4 gives the system instruction, which is followed by the current environment observation.

Table 4: System instruction for forward observation prediction.

FORWARD PREDICTION PROMPT
You are a text-based environment simulator. The user message is a fixed action already chosen by an external agent. Return only the next environment observation after that action, matching the original observation format. Preserve state details and report action failures when appropriate. Do not propose an action, give a plan, explain your reasoning, or include dialogue labels.

The user turn contains the recorded action. Broad fidelity retains the original complete message prefix and prepends this instruction to its initial system turn. Recursive prediction begins with the recorded initial observation and appends generated assistant turns. The inverse instruction asks the model to infer the action causing the transition and return only that action; its user turn provides the previous and next observations.

B Data Construction and Cohort Boundaries

Observed branches. ALFWorld training, development, and test branches contain 985, 140, and 134 pairs, respectively. They come from the training, seen-validation, and unseen-validation environments. Each ALFWorld pair was constructed by replaying the same prefix in independently initialized environments and verifying the visible state and admissible-action set before the two actions. This study reads the resulting observations as fixed data.

For SciWorld, candidate branches are repeated visible observations within one conversation, followed by distinct recorded actions and distinct outcomes. For TextWorld and WebShop training, records are grouped by source identifier and normalized visible observation. A source identifier may cover multiple conversation records, so these training units are source groups. The primary released-test branches instead require a repeated observation within one conversation. Normalization collapses whitespace; overlap exclusion additionally lowercases text. The logged construction establishes equality of visible text, while ALFWorld replay additionally certifies the environment reconstruction.

One pair is selected per source group by a fixed rule for training and development. Complete state-text matches against development and primary test branches are excluded before branch training. The resulting training pool contains 2,417 pairs: 985 ALFWorld, 90 SciWorld, 342 TextWorld, and 1,000 WebShop. Domain-balanced training uses this pool without giving the largest domain proportionally more updates. The development pool contains 440 pairs. Table 5 separates pair counts from source coverage.

Table 5: Primary branch cohorts. Training counts refer to selected pairs, with source groups used for TextWorld and WebShop. The final column describes the primary test conversations; the additional 200-group TextWorld holdout is separate.

Environment	Training pairs	Development pairs	Test pairs	Test conversations
ALFWorld	985	140	134	134
SciWorld	90	100	156	156
TextWorld	342	100	15	2
WebShop	1,000	100	90	90

Additional TextWorld source-group holdout. The released training records contain many more observed TextWorld branches than the original test subset. We form a separate holdout by removing source identifiers used in any known adapter-training stage, including the shared initialization, new branch training, and native restoration. Development and primary test source identifiers are also excluded. We further exclude normalized observations found in those sources, including identifiable observation segments from the earlier concatenated-history training data and the native restoration histories. From the remaining groups, one pair is chosen by a fixed rule, and the first 200 groups in a fixed ordering form the cohort. The checkpoint is unchanged, and these selection rules precede this

cohort's predictions. Both outcomes come from logged records with the same visible observation; they need not come from a single conversation within the group.

This cohort contains 200 source groups and 400 generated observations per model. It broadens TextWorld coverage while preserving its separate provenance as held-out material from the released training pool. Its results are never pooled with the original 15-pair test score. The shared objective comparison and Qwen3 are evaluated on the same 200 pairs.

Broad fidelity. One transition is selected from each released test conversation using a fixed rule based on its identifier. The source suite contains 822 transitions across ALFWorld test and unseen-validation records, SciWorld, TextWorld, and WebShop. A complete-state comparison with the new branch-training pool removes 61 rows: 25 ALFWorld, 8 SciWorld, and 28 TextWorld. All methods share the remaining 761 rows. The same saved predictions support both this exclusion and the more stringent analysis below; no additional generation is required.

Local-transition sensitivity. History-conditioned training can contain the same local observation–action–outcome triple in a different recorded history. To examine dependence on such repeated transitions, we additionally exclude every local triple matching any known adapter-training corpus: the shared initialization, new branch training, or native restoration. This stricter cohort has 536 transitions: 213 ALFWorld, 108 SciWorld, 106 TextWorld, and 109 WebShop. Its mask depends only on data, and every model uses the same mask. After further excluding local transitions occurring in known adapter-training corpora, REPLAYBIND achieves an average token F1 of 0.7796, compared with 0.5840 for Qwen3 (Table 6). The observation-fidelity advantage therefore persists after removing examples with different histories but identical local transitions.

Table 6: Fidelity after additional local-transition exclusion. Token F1 on the 536-transition sensitivity cohort, averaged equally across environments. The mask additionally removes local triples occurring in any known adapter-training source. In this table, bold and underlining mark the best and next distinct scores within each comparison; ties receive the same emphasis.

Model	ALFWorld	SciWorld	TextWorld	WebShop	Mean
Word2World-7B	**1.0000**	0.2286	0.0020	0.0678	0.3246
WMA-8B	0.5476	0.3064	0.4296	0.5687	0.4631
WebWorld-8B	0.1299	0.1143	0.2435	0.4059	0.2234
Qwen2.5-7B	0.5788	0.5016	0.4558	0.5926	0.5322
Qwen3-30B-A3B	0.6461	0.5904	0.4702	0.6294	0.5840
Without crossed loss	<u>0.9722</u>	<u>0.7604</u>	**0.7499**	<u>0.6353</u>	<u>0.7795</u>
REPLAYBIND	0.9689	**0.7619**	<u>0.7346</u>	**0.6530**	**0.7796**

Recursive trajectories. The ten-step cohort contains 162 SciWorld, 118 TextWorld, and 48 WebShop trajectories. Every selected trajectory has at least ten recorded transitions. Each model begins from the same recorded initial observation and receives the same action at each horizon; only its generated observation history differs. No intermediate state is reset to the recorded target. Scores first average trajectories within an environment and then average the three environments equally.

C DETAILED OBJECTIVE AND GENERATION COMPARISONS

Figure 6 compares the effect of crossed supervision on teacher-forced assignment after binding and complete training. Point positions show gains over the matched no-crossed control, while labels retain both absolute scores. Assignment accuracy counts pairs whose recorded joint assignment has higher likelihood than its crossed alternative. The complete-model comparisons underlie Figure 2. ALFWorld and the small primary TextWorld cohort already have perfect joint assignment accuracy in both arms, making the margin informative even when the discrete decision is unchanged. WebShop provides improvements in both the margin and the number of correctly assigned pairs. The additional TextWorld source-group result in Table 2 shows that assignment-margin gains can accompany better generated branch separation.



Figure 6: **Crossed supervision across training stages.** Points show crossed-minus-control gains in forward and inverse assignment margins and accuracies. Diamonds denote the 600-update binding stage; circles denote complete training with restoration and replay. Labels give no-crossed → crossed absolute scores, with accuracy expressed as percentages; accuracy gains are percentage points. Zero means unchanged accuracy.

D Complete Branch Results

Table 7: Generated branch margins across environments. Mean own-outcome minus crossed-outcome token F1. The cohorts contain 134, 156, 15, and 90 pairs, respectively. TextWorld's original 15 pairs come from two conversations; Table 2 reports the larger, separately sourced holdout.

Model	ALFWorld	SciWorld	TextWorld	WebShop
Word2World-7B	0.0863	0.0014	0.0000	0.0000
WMA-8B	0.0463	0.0198	0.0366	<u>0.0747</u>
WebWorld-8B	0.0233	0.0064	0.0117	0.0205
Qwen2.5-7B	0.0945	**0.0645**	0.0235	0.0252
Qwen3-30B-A3B	<u>0.1048</u>	<u>0.0523</u>	<u>0.0500</u>	0.0740
ReplayBind	**0.1907**	0.0429	**0.1770**	**0.2735**

Figure 7 compares the complete model with the external models on the primary cohorts. Token and character-trigram F1 measure different aspects of text overlap; exact word content can remain similar despite differences in spacing or formatting. The branch margin compares the same prediction with both observed outcomes. Strict-pair success requires correct preferences on both sides and counts exact ties as failures.

E Recursive Prediction Results

Table 9 gives step-ten domain scores and the average over all horizons for every evaluated model. Table 10 gives the complete macro-average trajectory. The recurrent inputs consist of predicted assistant observations and recorded user actions. Consequently, the measurements describe observation dynamics along a fixed action sequence.

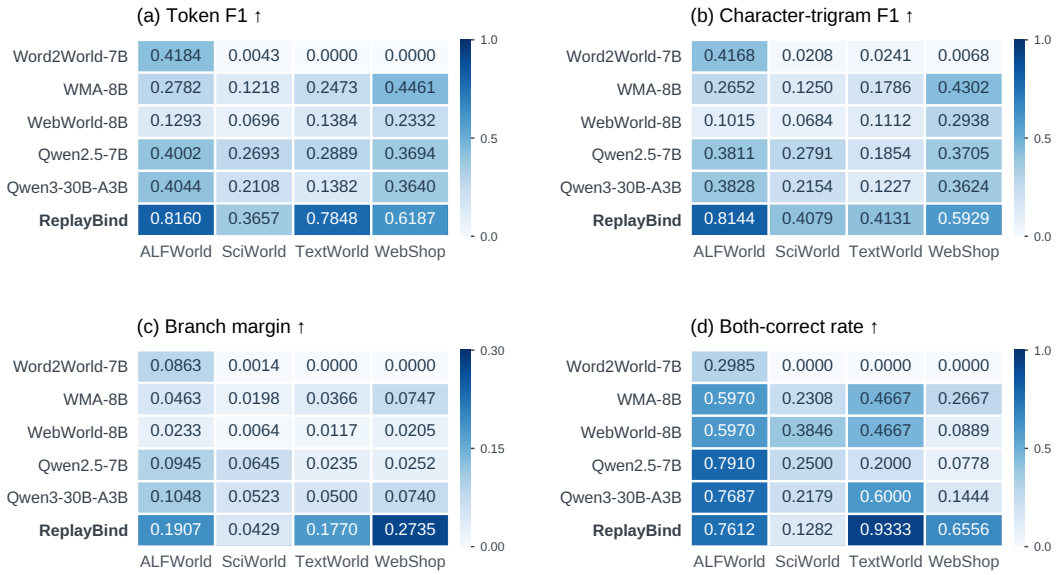


Figure 7: **Full primary branch measurements.** Rows compare models and columns compare environments on the same test pairs. Cells retain the absolute scores; darker shading means higher values within the labeled scale of each panel. Token and character-trigram F1 measure observation overlap. Branch margin is mean own-minus-crossed token F1; both-correct requires strictly positive preferences on both branches.

Table 8: Required separation between branch outcomes. Fraction of pairs with both generated preferences exceeding τ. These thresholds rescore the saved predictions and do not invoke the model again.

Model	Environment	$\tau = 0$	$\tau = 0.01$	$\tau = 0.025$	$\tau = 0.05$	$\tau = 0.1$
Qwen3-30B-A3B	ALFWorld	**0.7687**	0.7388	**0.7015**	0.5597	0.3358
Qwen3-30B-A3B	SciWorld	**0.2179**	**0.1923**	**0.1731**	**0.1538**	**0.1218**
Qwen3-30B-A3B	TextWorld	0.6000	0.2667	0.2000	0.0000	0.0000
Qwen3-30B-A3B	WebShop	0.1444	0.1222	0.0889	0.0444	0.0111
REPLAYBIND	ALFWorld	0.7612	**0.7612**	0.6716	**0.6194**	**0.4627**
REPLAYBIND	SciWorld	0.1282	0.1282	0.0897	0.0385	0.0256
REPLAYBIND	TextWorld	**0.9333**	**0.9333**	**0.7333**	**0.7333**	**0.4667**
REPLAYBIND	WebShop	**0.6556**	**0.6222**	**0.6111**	**0.5889**	**0.5333**

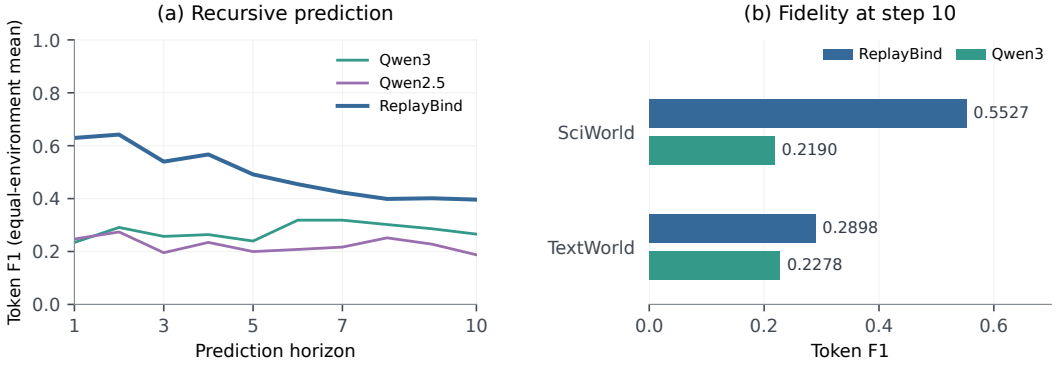


Figure 8: **Recursive prediction with native chat templates.** Left: equal-environment token F1 across ten prediction steps on 328 logged trajectories for ReplayBind and the two strongest external models by mean recursive fidelity. Right: step-ten token F1 on SciWorld and TextWorld, comparing ReplayBind and Qwen3. Table 9 gives results for all three environments. Predicted observations feed back into subsequent inputs.

Table 9: Ten-step native-template simulation. Environment columns and step-ten mean refer to the final horizon. All-steps averages the ten macro-average scores. There are 328 trajectories and 3,280 generated observations per model.

Model	SciWorld	TextWorld	WebShop	Step 10 mean	All steps
Word2World-7B	0.0130	0.0153	0.0000	0.0094	0.0127
WMA-8B	0.0503	0.2077	0.2054	0.1545	0.1476
WebWorld-8B	0.0511	0.1492	0.1828	0.1277	0.1334
Qwen2.5-7B	0.1578	0.2331	0.1715	0.1875	0.2242
Qwen3-30B-A3B	0.2190	0.2278	**0.3503**	0.2657	0.2776
REPLAYBIND	**0.5527**	**0.2898**	0.3460	**0.3962**	**0.4943**

Table 10: Macro token F1 at each recursive horizon. Each row reports one model's scores across prediction steps, averaged equally over the three environments.

Model	1	2	3	4	5	6	7	8	9	10
Word2World-7B	0.0023	0.0018	0.0129	0.0303	0.0285	0.0209	0.0083	0.0081	0.0046	0.0094
WMA-8B	0.1433	0.1317	0.1210	0.1322	0.1338	0.1475	0.1745	0.1748	0.1626	0.1545
WebWorld-8B	0.1318	0.1130	0.1074	0.1265	0.1316	0.1381	0.1599	0.1531	0.1445	0.1277
Qwen2.5-7B	0.2472	0.2739	0.1955	0.2344	0.1997	0.2078	0.2167	0.2516	0.2279	0.1875
Qwen3-30B-A3B	0.2342	0.2910	0.2571	0.2639	0.2395	0.3185	0.3185	0.3019	0.2861	0.2657
REPLAYBIND	**0.6294**	**0.6420**	**0.5398**	**0.5669**	**0.4915**	**0.4545**	**0.4229**	**0.3987**	**0.4013**	**0.3962**

F STATE EDITS AND TEXTUAL CONTROLS

Observation overlap includes persistent state content. For current state text s and next observation o, let $C(x)$ be the normalized token multiset of any text x, retaining repeated occurrences. Its signed edit representation is

$$E(s, o) = \{(+, w) : w \in C(o) \setminus C(s)\} \uplus \{(-, w) : w \in C(s) \setminus C(o)\}. \tag{10}$$

Here w is a token, $\setminus$ subtracts token counts with a floor of zero, and $\uplus$ adds multiset counts. The signs $+$ and $-$ distinguish additions from deletions. Edit F1 compares $E(s, \hat{o})$ for predicted observation $\hat{o}$ with the reference $E(s, o)$. Figure 9 uses the ALFWorld and WebShop branch predictions to show how observation gains relate to explicit changes.

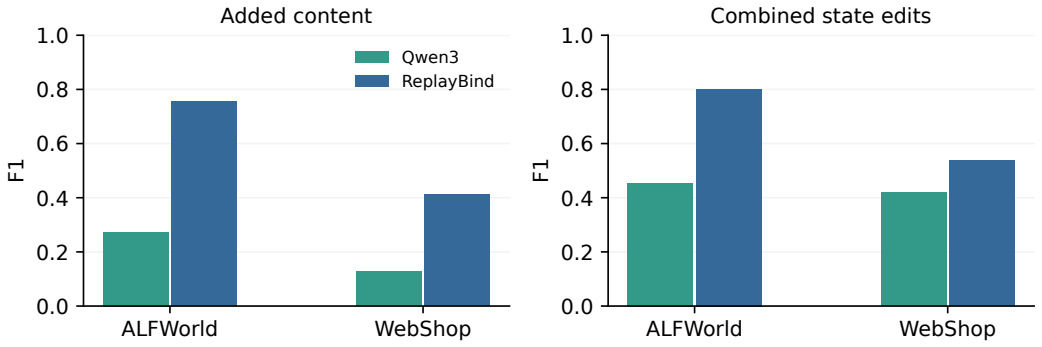


Figure 9: **State-change recovery on ALFWorld and WebShop.** The complete model is compared with Qwen3. Added-content F1 evaluates newly introduced tokens; combined edit F1 evaluates signed additions and deletions. Channel-specific scores use nonempty reference channels, and combined scores use observations with a reference change.

Added-content F1 is averaged over observations with at least one reference addition; removed-content F1 uses observations with at least one reference deletion. Combined edit F1 uses observations with any reference edit. A prediction that makes no changes has an empty edit multiset. An

empty observation deletes the entire current state, and repeating the action can share a few new tokens with the target; these controls help interpret the signed-edit score rather than serving as learned model baselines.

Table 11: State-edit recovery in all four environments. Changed counts reference observations with a nonempty signed edit. Add, remove, and edit F1 use the averaging rules described above. Predictions are reused from primary branch generation.

Model	Environment	Changed	Add F1	Remove F1	Edit F1
Qwen3-30B-A3B	ALFWorld	268	0.2735	0.7596	0.4544
Qwen3-30B-A3B	SciWorld	255	**0.2217**	**0.7548**	**0.3548**
Qwen3-30B-A3B	TextWorld	28	0.0920	0.4150	0.1794
Qwen3-30B-A3B	WebShop	126	0.1282	0.6442	0.4182
REPLAYBIND	ALFWorld	268	**0.7539**	**0.9161**	**0.8014**
REPLAYBIND	SciWorld	255	0.1975	0.3848	0.2728
REPLAYBIND	TextWorld	28	**0.6641**	**0.9341**	**0.8145**
REPLAYBIND	WebShop	126	**0.4133**	**0.6998**	**0.5359**

Table 12: Static state-edit controls. Combined edit F1 for copying the observation, repeating the recorded action, returning no text, and the complete learned model. These controls characterize the metric's sensitivity to additions and deletions.

Predictor	ALFWorld	SciWorld	TextWorld	WebShop
Copy observation	0.0000	0.0000	0.0000	0.0000
Repeat action	0.5111	**0.6081**	0.3943	0.4039
Empty output	0.4608	0.5282	0.3145	0.4051
REPLAYBIND	**0.8014**	0.2728	**0.8145**	**0.5359**

G ACTION-ARGUMENT DIAGNOSTICS

A symmetric redaction diagnostic removes both actions' argument words from generated and reference observations before scoring. ALFWorld's remaining margin is 0.1045 for REPLAYBIND and 0.0364 for Qwen3; WebShop gives 0.2703 and 0.0731. The preference therefore includes textual consequences beyond the direct repetition of action arguments. All pairs remain in the score, including those whose reference outcomes become indistinguishable after redaction.

For argument redaction, bracketed actions contribute the words inside brackets; other actions contribute tokens after the first word. The union of both actions' argument words is removed symmetrically from both predictions and both references. All pairs remain in the denominator. The redacted references remain distinguishable for 124 of 134 ALFWorld pairs, 155 of 156 SciWorld pairs, all 15 TextWorld pairs, and 87 of 90 WebShop pairs. Table 13 reports the resulting margin. This is a scoring diagnostic; training uses unredacted targets.

Table 13: Generated branch margin after symmetric argument redaction. All primary pairs remain included, including pairs made indistinguishable by removing the argument words.

Model	ALFWorld	SciWorld	TextWorld	WebShop
Qwen3-30B-A3B	0.0364	**0.0437**	0.0251	0.0731
REPLAYBIND	**0.1045**	0.0312	**0.1478**	**0.2703**

H RELATIONAL SUPERVISION AND GENERATIVE COVERAGE

Training stages. Table 14 separates broad observation fidelity from action-conditioned branch margins along the same training path.

Table 14: Training-stage comparison. Fidelity is the equal-domain token F1 on the 761-transition suite; environment columns are generated branch margins on the primary test cohorts. Each row is a successive stage of the same training path.

Training stage	Fidelity	ALFWorld	SciWorld	TextWorld	WebShop
Shared initialization	0.6251	0.0804	**0.0998**	0.0000	-0.0003
After binding	0.6808	<u>0.1816</u>	<u>0.0965</u>	<u>0.1742</u>	<u>0.2271</u>
After restoration	<u>0.7894</u>	0.1757	0.0434	0.0247	0.1357
REPLAYBIND	**0.8015**	**0.1907**	0.0429	**0.1770**	**0.2735**

Before restoration, crossed supervision raises broad fidelity from 0.6703 for the matched positive-likelihood control to 0.6808. After restoration, balanced replay raises WebShop branch margin from 0.1357 to 0.2735 and both-correct rate from 0.2778 to 0.6556. ALFWorld both-correct rate rises from 0.6791 to 0.7612; the original TextWorld cohort rises from 0.2000 to 0.9333.

The crossed objective and positive observation likelihood shape different aspects of the same conditional distribution. Positive likelihood rewards the model for reproducing a recorded continuation. The assignment term additionally asks whether the same continuation becomes less compatible when the conditioning action is exchanged. This distinction explains why a small change in broad text overlap can coexist with a consistent change in assignment margins: common observation tokens contribute heavily to the first measurement, while the second compares the relation between two fixed alternatives. The two directions supply complementary evidence because observation prediction and action reconstruction have different target spaces.

Generation then turns this distribution into one concrete continuation. A sequence-level likelihood advantage can be distributed across many token decisions, whereas decoding produces a single token sequence. The additional TextWorld comparison is informative at this point in the evaluation: improved assignment margins accompany stronger separation of the generated futures themselves, on source groups excluded from the known adaptation data. The primary four-environment results describe the broader variation around that effect. Reporting both views makes the empirical role of relational training precise without treating likelihood and generated overlap as interchangeable scores.

The conditioning information also matters. Broad fidelity and recursive prediction expose a conversation history, while branch prediction holds only the current visible observation and action fixed. A short response such as a door-opening confirmation may omit the room, inventory, or task progress established earlier. The broader history can supply those details, while the branch interface directly tests sensitivity to the supplied action under a shared observation. Native restoration and balanced replay train these complementary uses of the generator. Their stage-wise behavior follows that division: restoration improves broad coverage, and replay strengthens generation under the branch interface while maintaining the broad gains. The resulting evidence links the training signals to distinct, observable behaviors of one model.

I DIRECTIONS AND PAIRING IN CROSSED SUPERVISION

Table 15: Directional and ranking controls. Exact-match percentages for the extended-context objective comparison. The first two columns reconstruct actions on the same four-environment cohort, with one decoding per unique input. The last column is four-candidate inverse assignment on new sources, averaging SciWorld (27 groups) and WebShop (100 groups) equally. Bold and underline denote the best and next distinct score in each column.

Objective	Single reference	Logged references	New-source assignment
No crossed term	28.44	33.11	40.83
Forward crossed only	<u>29.33</u>	<u>34.45</u>	42.69
Inverse crossed only	28.67	33.18	<u>43.83</u>
Row-wise ranking	28.90	33.57	42.33
Bidirectional crossed	**29.96**	**35.00**	**44.54**

The primary generation experiments and the extended-context experiments use the same full-token crossed objective, with separate matched training configurations. The primary setting uses $\beta = 0.5$. The extended-context setting retains longer observed outcomes during binding and longer recorded histories during restoration. Its directional and ranking comparisons use $\beta = 0.05$. The two settings share the adapted initialization described in Appendix A; results within each comparison use its corresponding control.

Training configuration. Extended-context binding runs for 600 updates with a 3,072-token total sequence budget and a 1,024-token target budget. Restoration and replay use a 4,608-token total budget and up to 512 target tokens. All five objective conditions then receive 400 broad restoration updates and 200 balanced replay updates. Learning rates remain 5×10^{-6} for binding and 3×10^{-6} for the subsequent stages. The training data, ordering, initialization, update counts, and positive forward and inverse likelihoods are shared. Each scheduled pair computes all four conditional combinations.

Directional controls. Forward-only crossed supervision applies the assignment hinge during forward updates and retains positive likelihood alone during inverse updates. Inverse-only crossed supervision makes the complementary deletion. Both retain positive targets in both directions. Deleting a direction leaves the other coefficient unchanged, making the comparison a component deletion at the stated coefficient.

Row-wise ranking. For branch indices $i, j \in \{1, 2\}$, let C_{ij} be the summed negative log likelihood of target j under condition i, and T_j its scored token count. Conditions are state–action pairs in the forward view and state–outcome pairs in the inverse view; targets are outcomes and actions, respectively. Diagonal entries are recorded pairings and off-diagonal entries are exchanged pairings. For either direction, the positive term is $\mathcal{L}_+ = (C_{11}/T_1 + C_{22}/T_2)/2$. The ranking control uses

$$\mathcal{L}_{\text{row}} = \mathcal{L}_+ + \frac{\beta}{2}\left([m - (C_{12}/T_2 - C_{11}/T_1)]_+ + [m - (C_{21}/T_1 - C_{22}/T_2)]_+\right). \tag{11}$$

The row-wise loss $\mathcal{L}_{\text{row}}$ compares each condition's two targets independently, using the same weight β, margin m, and positive-part operator $[\cdot]_+$ as Eq. 5. Crossed assignment instead uses a single hinge on $(C_{12} + C_{21} - C_{11} - C_{22})/(T_1 + T_2)$. Both see the same observed targets and exchanged combinations. This comparison tests joint assignment against an explicit per-condition ranking alternative at the fixed training budget.

J Action Cohorts and Candidate Construction

Observed alternatives. The action cohort groups logged transitions by environment, source identity, and normalized visible observation. Distinct actions and distinct recorded successor observations define candidate branches. Fixed ordering selects up to four alternatives, prioritizing sources with three or four candidates before model scoring. Each environment contributes 100 source groups. ALFWorld, SciWorld, and WebShop each supply 100 four-candidate groups; TextWorld supplies 23 four-candidate and 77 three-candidate groups. This gives 400 groups and 1,523 transitions, with 323 groups available for four-candidate assignment.

The candidate-rich construction focuses evaluation on competing explanations of a visible transition. Known adapter-training sources and identifiable training observations are excluded. Some sources were previously used for other evaluations: 92 ALFWorld groups and 78 TextWorld groups have that status, while the SciWorld and WebShop groups do not. Accordingly, this cohort introduces assignment and action-generation tasks on the specified sources. The new-source cohort below supplies a separate source-level comparison.

Native interfaces. Conditional likelihoods use each checkpoint's native chat template, a 4,096-token total budget, and up to 1,024 target tokens. Action generation uses up to 4,096 input tokens and 128 output tokens. Exact native inputs are deduplicated within each model/cohort evaluation, giving 1,451 unique action prompts for the initial 1,523 transitions. All transition references remain in scoring, including 19 prompt collisions with different representative actions. Each unique prompt is decoded once. Shared input text is handled by reference analysis rather than by excluding difficult targets.

Table 16: Source groups for assignment and action reconstruction. All initial groups have at least three candidates. Among new sources, 10 ALFWorld groups have three candidates, all SciWorld groups have at least three, and TextWorld groups have two. Candidate counts determine which environments contribute to each assignment comparison.

Environment	Initial groups	Initial four-way	New groups	New four-way
ALFWorld	100	100	74	0
SciWorld	100	100	100	27
TextWorld	100	23	100	0
WebShop	100	100	100	100
Total	400	323	374	127

K Multi-Candidate Assignment

For k observed action–outcome alternatives sharing state s, indices $i, j \in \{1, \ldots, k\}$ label complete candidates. Define forward costs $C^f_{ij} = -\log p_\theta(o_j \mid s, a_i)$ and inverse costs $C^i_{ij} = -\log p_\theta(a_j \mid s, o_i)$. In the expression below, C denotes the cost matrix for the direction being evaluated. Let Π_k be all one-to-one permutations of the k indices, with $\mathrm{id}(j) = j$ denoting the recorded pairing. The indicator A_k is one when that pairing has strictly lower cost than every alternative:

$$A_k = \mathbf{1}\left\{ \sum_{j=1}^{k} C_{jj} < \min_{\pi \in \Pi_k \setminus \{\mathrm{id}\}} \sum_{j=1}^{k} C_{j,\pi(j)} \right\}. \tag{12}$$

Here $\pi(j)$ is the target assigned to condition j; $\mathbf{1}$ is the indicator defined in Eq. 8. Ties count as failures. Every permutation uses each target once, so target-only additive preferences cancel. Three- and four-candidate scores use the corresponding fixed subsets of the same maximal conditional matrix. Their different cohort sizes are stated explicitly.

Table 17: Assignment with three and four observed alternatives. Environment-average exact assignment percentages for extended-context models with $\beta = 0$ and 0.05. Three-candidate evaluation has 100 groups per environment; four-candidate evaluation has the 323 groups in Table 16. Gains are percentage points.

Candidates	Direction	No crossed	Crossed	Gain (pp)
3	Forward	81.25	81.75	+0.50
3	Inverse	74.25	75.00	+0.75
4	Forward	77.66	78.16	+0.50
4	Inverse	55.52	58.70	+3.17

Table 17 connects the pairwise training signal to assignments with more alternatives. The inverse four-candidate gain is 3.17 percentage points, compared with 0.75 for three candidates. These tasks require all actions in a group to be assigned correctly. The larger inverse gain accompanies the additional alternatives, although the TextWorld cohort also changes from 100 to 23 groups. Directly pooling four-candidate groups gives 202 versus 208 correct inverse assignments and 237 versus 239 correct forward assignments for zero and nonzero crossed weight, respectively.

Table 18: Action pairing and reconstruction by environment. Percentages for the same extended-context zero/coefficient-0.05 comparison. Inverse assignment uses the listed four-candidate groups; generated action matching uses all 400, 400, 323, and 400 transitions, respectively.

Environment	Groups	Inverse: zero	Crossed	Action: zero	Crossed
ALFWorld	100	91.00	92.00	55.75	56.75
SciWorld	100	81.00	83.00	25.00	25.75
TextWorld	23	26.09	34.78	19.50	23.84
WebShop	100	24.00	25.00	13.50	13.50

Inverse assignment improves in all four environments (Table 18). TextWorld changes from 6 to 8 correct four-candidate groups, SciWorld from 81 to 83, and ALFWorld and WebShop each gain one group. In generated action matching, ALFWorld, SciWorld, and TextWorld improve while WebShop is unchanged. These two views locate the benefit in both the scored relation among alternatives and the generated action string.

Relative to row-wise ranking, inverse accuracy increases from 55.27% to 58.70%, or 201 to 208 correct groups. The complete crossed objective also improves forward assignment relative to inverse-only crossed supervision, from 77.41% to 78.16%. Each comparison refers to its named endpoint; Table 15 compares all five objectives on action reconstruction.

Table 19: External models on four-candidate assignments. Exact assignment percentages, averaged equally over the four environments. All models use their own native templates and identical candidate groups. The adapted model uses the extended-context coefficient-0.05 setting.

Model	Forward assignment	Inverse assignment
Word2World-7B	46.35	27.75
Qwen3-30B-A3B	34.60	27.34
REPLAYBIND (extended context)	**78.16**	**58.70**

Against the external models, the complete extended-context procedure improves forward and inverse assignment (Table 19). Its gains over Qwen3 are 43.57 and 31.36 percentage points, respectively. These model-level comparisons include the full adaptation procedure; the matched objective comparisons identify the incremental effect of the crossed term.

L ACTION RECONSTRUCTION AND LOGGED REFERENCES

Complete action strings. Given a previous and successor observation, the inverse instruction asks for the recorded action without an explanation. Exact match compares normalized complete strings. Table 15 reports the resulting environment means. The complete crossed objective improves over zero weight and over each directional or ranking control. The endpoint requires all output content to match an accepted action, so it reflects both action content and the avoidance of extraneous output.

Multiple recorded explanations. The candidate construction retains one representative action for each selected successor observation. Other actions in the same source can yield the same visible transition. For each evaluated (s, o), let $\mathcal{A}_{\log}(s, o)$ contain all action strings actually recorded with that source and normalized observation pair. The supplementary reference score is

$$\mathrm{EM}_{\log}(\hat{a}, s, o) = \mathbf{1}\{\mathrm{norm}(\hat{a}) \in \{\mathrm{norm}(a) : a \in \mathcal{A}_{\log}(s, o)\}\}. \tag{13}$$

Here $\hat{a}$ is the generated action, norm lowercases text and collapses whitespace, and $\mathrm{EM}_{\log}$ is one if the normalized prediction matches any logged reference, zero otherwise. The reference set is built from logs independently of predictions. It contains multiple actions for 454 of 1,523 transitions, with at most 36 actions per transition. This is a reference-based reanalysis of the same generated outputs. It accepts observed alternatives without assuming they are interchangeable under unobserved simulator state.

Table 20: Action matching with all logged references. Per-environment exact-match percentages for the extended-context zero/coefficient-0.05 comparison. The same predictions support single- and multiple-reference scoring.

Environment	No crossed term	Bidirectional crossed
ALFWorld	56.00	**57.25**
SciWorld	26.25	**27.00**
TextWorld	34.67	**40.25**
WebShop	**15.50**	**15.50**

The environment mean rises from 33.11% to 35.00%, with gains in ALFWorld, SciWorld, and TextWorld and an unchanged WebShop score under the logged-reference criterion. Table 15 further

shows the highest score among all five objectives. The gain over ranking grows from 1.06 percentage points under the representative reference to 1.43 under logged references. Thus accepting additional recorded actions preserves the advantage of bidirectional assignment supervision.

Before observation restoration. The binding checkpoint reveals the behavior immediately after the objective stage. Bidirectional supervision already gives the highest single- and multiple-reference action scores among the five conditions (Table 21). It reconstructs 452 of 1,523 representative actions, compared with 445 without the crossed term and 449 under row-wise ranking. The subsequent restoration and replay comparison evaluates the same endpoint after observation-generation training, showing that the favorable ordering is present at both stages.

Table 21: Action reconstruction after 600 binding updates. Environment-average exact-match percentages and pooled representative-reference counts. These are earlier checkpoints of the five training conditions in Table 15, evaluated on the same cohort.

Objective	Single reference	Logged references	Correct transitions
No crossed term	28.86	32.79	445/1523
Forward crossed only	28.62	32.68	441/1523
Inverse crossed only	28.59	32.74	440/1523
Row-wise ranking	<u>29.18</u>	<u>33.40</u>	<u>449/1523</u>
Bidirectional crossed	**29.35**	**33.51**	**452/1523**

Table 22: Generated actions compared with Qwen3. Environment-average exact-match percentages on the initial action cohort. Both checkpoints use their native chat templates, with 4,096 input and 128 output tokens.

Model	Single reference	Logged references
Qwen3-30B-A3B	<u>7.00</u>	<u>9.34</u>
REPLAYBIND (extended context)	**29.96**	**35.00**

M ASSIGNMENT AND ACTION RECONSTRUCTION ON AN ADDITIONAL HELD-OUT SET

The new-source cohort excludes identified training sources and sources used in earlier evaluations. Selection uses source identities and observations before scoring; model settings remain frozen. The initial comparison evaluates the zero and bidirectional conditions. The forward-only, inverse-only, and ranking conditions are subsequently evaluated with their already fixed training settings. Their extension to this cohort follows the initial comparison, while model training is unchanged.

Available alternatives. The 374 groups comprise 74 ALFWorld groups and 100 from each other environment, for 1,085 transitions. Four-candidate assignment is available for 27 SciWorld groups and 100 WebShop groups. The predefined macro-average includes environments with at least 20 eligible groups. Three-candidate averaging therefore also uses SciWorld and WebShop; the 10 eligible ALFWorld groups remain outside that average. At four candidates, the 23 eligible TextWorld sources were already used in the initial cohort, so the new-source TextWorld comparison uses the available two-candidate transitions for action generation.

The four-candidate inverse mean increases from 40.83% to 44.54%. SciWorld contributes 18 versus 20 correct groups out of 27, while WebShop contributes 15 in both conditions; the pooled total is 33 versus 35 out of 127. Across all five objectives, Table 15 reports the highest environment-average score for bidirectional supervision. This comparison uses equal environment weights despite the different domain sample counts. Forward assignment also improves relative to forward-only crossed supervision, from 54.80% to 57.15%, and relative to inverse-only crossed supervision, from 56.15% to 57.15%.

The action-generation mean increases from 22.65% to 23.02% against zero crossed weight. TextWorld accounts for the improvement, from 23.00% to 24.50%, with the other environments

Table 23: New-source assignment comparisons. Exact assignment percentages for the extended-context zero and bidirectional conditions. Each listed macro-average gives equal weight to SciWorld and WebShop; three-candidate evaluation has 100 groups per environment and four-candidate evaluation has 27 and 100.

Candidates	Direction	No crossed	Crossed	Gain (pp)
3	Inverse	58.50	**60.00**	+1.50
4	Forward	56.30	**57.15**	+0.85
4	Inverse	40.83	**44.54**	+3.70

Table 24: Action reconstruction on new sources. Exact-match percentages by environment, using 158 ALFWorld, 327 SciWorld, 200 TextWorld, and 400 WebShop transitions. Both conditions use the extended-context training configuration.

Environment	No crossed term	Bidirectional crossed
ALFWorld	**34.18**	**34.18**
SciWorld	**21.41**	**21.41**
TextWorld	23.00	**24.50**
WebShop	**12.00**	**12.00**

unchanged (Table 24). The bidirectional mean also exceeds the forward-only and inverse-only conditions, 22.79% and 22.72%, respectively. Multiple logged actions are available for 362 of the 1,085 new-source transitions, with at most 75 alternatives.

Input-level overlap. Source-level separation and unique model inputs are complementary properties. Native encoding yields 1,082 unique action prompts in the new-source cohort. Ten evaluated transitions share an encoded input with the initial cohort: eight SciWorld and two TextWorld transitions. Excluding these inputs by their identity retains 1,075 transitions and preserves the single-reference and logged-reference gains in Table 25. The exclusion depends only on input identity. The four-candidate comparison has no overlapping complete encoded problems in either direction, accounting for candidate reordering and the likelihood token budgets.

Table 25: New-source action matching after input exclusion. Environment-average exact-match percentages, with 1,085 transitions before exclusion and 1,075 afterward. Multiple references accept all actions supported by the corresponding raw logs. Both analyses reuse the existing predictions.

Inputs	References	No crossed	Crossed
All new-source inputs	Single	22.65	**23.02**
All new-source inputs	Multiple	27.15	**27.28**
Previously unseen inputs	Single	22.63	**23.01**
Previously unseen inputs	Multiple	27.19	**27.31**

N Generated Branch Discrimination under Different Training Schedules

The full-token crossed objective is also evaluated at its primary coefficient, $\beta = 0.5$, with longer training contexts. Two matched families examine generation. Extended-context training uses the 600 binding and 600 restoration/replay updates described above. Mixed continuation starts both conditions from the primary zero-coefficient complete model and adds 200 updates combining broad observation likelihood with bidirectional branch supervision. Its branch budgets are 3,072/1,024 tokens and its broad budgets are 4,608/512, at learning rate 3×10^{-6}. The mixed pool expands available SciWorld and TextWorld branch sources; the actual 200-update schedule uses 100 distinct pairs per environment. The extended-context binding schedule uses 300 distinct pairs each for ALFWorld, TextWorld, and WebShop and 90 for SciWorld.

Table 26: Generated separation under two matched training schedules. The four-environment column is macro strict-pair success (%) on 395 pairs. TextWorld columns use its separate 200-group cohort, reporting token F1 and strict-pair success (%). Each crossed condition uses coefficient 0.5 and its own family's zero control.

Training	Objective	Four-env. strict	TextWorld F1	TextWorld strict
Extended context	No crossed	58.48	0.6869	40.00
Extended context	Crossed	**58.68**	**0.7097**	**47.00**
Mixed continuation	No crossed	53.89	0.7565	65.50
Mixed continuation	Crossed	**56.16**	**0.7718**	**71.00**

For extended-context training, TextWorld token F1 increases from 0.6869 to 0.7097 and strict success from 80 to 94 of 200 pairs. The four-environment generation F1 also increases, from 0.6547 to 0.6581. Mixed continuation gives 131 versus 142 successful TextWorld pairs and increases its F1 from 0.7565 to 0.7718 (Table 26). On the 395-pair cohort, mixed continuation changes the pooled strict count from 187 to 191 and the environment-average success rate from 53.89% to 56.16%. These evaluations use established regression cohorts; each training family is interpreted through its own matched comparison.

Table 27: Generated action separation after mixed continuation. Strict-pair percentages. ALFWorld, SciWorld, and WebShop use 134, 156, and 90 primary pairs; TextWorld uses its separate 200-group cohort.

Environment	No crossed term	Crossed term
ALFWorld	**74.63**	**74.63**
SciWorld	15.38	**16.67**
WebShop	58.89	**60.00**
TextWorld	65.50	**71.00**

Table 28: Assignment margins after mixed continuation. Forward and inverse likelihood margins on the primary branch cohorts, using the same 4,096-token total and 1,024-token target scoring budgets in both conditions.

Environment	Forward: zero	Crossed	Inverse: zero	Crossed
ALFWorld	0.9248	**0.9355**	2.0684	**2.0744**
SciWorld	0.8924	**0.9023**	1.4558	**1.4693**
TextWorld	0.6596	**0.6731**	2.9288	**2.9814**
WebShop	0.0787	**0.0880**	0.9924	**1.0189**

The mixed comparison improves strict generated separation in SciWorld, WebShop, and the larger TextWorld cohort, while ALFWorld is unchanged (Table 27). Forward and inverse likelihood margins improve across all four primary environments (Table 28). Together, these measurements connect the relational training signal to preference among recorded outcomes and to the concrete observations produced by decoding.

O Recovering Branch-Specific Content

Observation-level overlap can be dominated by text shared between two candidate futures. To isolate their lexical differences, let R_1, R_2 be reference token multisets, $U = R_1 \cap R_2$ their multiset intersection, and $D_j = R_j \setminus U$ the content distinctive to branch j. For prediction multiset P_j, score $P'_j = P_j \setminus U$ against D_j. Precision, recall, and F1 use these residual multisets. The fraction of predicted residual tokens belonging to the other branch is

$$W_j = \frac{|P'_j \cap D_{3-j}|}{|P'_j|}, \tag{14}$$

Here $j \in \{1, 2\}$ indexes branches, so $3 - j$ identifies the other branch; $|\cdot|$ counts token occurrences, $\cap$ takes the smaller count for each token, and $\setminus$ subtracts counts with a floor of zero. Set $W_j = 0$

when the residual prediction P'_j is empty. Pairs require nonempty residual references on both sides; 391 of the 395 primary pairs and all 200 TextWorld pairs satisfy this rule. References determine the mask without being provided to the generating model. Scores average the two branches, then pairs within an environment, then environments equally.

Table 29: Branch-specific content on the 200-group TextWorld cohort. Precision, recall, and F1 after subtracting shared reference tokens. The final column measures the fraction belonging to the opposite branch; lower is better. Both training families use the full-token coefficient-0.5 objective, and this analysis reuses their generated observations.

Training	Objective	Precision	Recall	F1	Other-branch fraction ↓
Extended context	No crossed	0.6174	0.5760	0.5805	0.2670
Extended context	Crossed	**0.6525**	**0.6091**	**0.6147**	**0.2333**
Mixed continuation	No crossed	0.7493	0.6953	0.7002	0.1284
Mixed continuation	Crossed	**0.7794**	**0.7111**	**0.7216**	**0.1053**

Extended-context training raises distinctive-content F1 from 0.5805 to 0.6147 and lowers the opposite-branch token fraction from 0.2670 to 0.2333. Mixed continuation raises F1 from 0.7002 to 0.7216 while lowering that fraction from 0.1284 to 0.1053. Precision and recall improve in both comparisons (Table 29). Figure 4 compares target-specific recovery with the inclusion of opposite-branch content. They identify a concrete component of the generation benefit: more target-specific content and less content associated with the alternative outcome.

On the 391 eligible primary pairs, the extended-context coefficient-0.05 model raises distinctive F1 from 0.5352 to 0.5373 and lowers the opposite-branch fraction from 0.0814 to 0.0807. At coefficient 0.5, distinctive F1 is 0.5395 and the opposite fraction is 0.0812. These are lexical diagnostics of the fixed predictions; the branch-specific score and full-observation score answer complementary questions.

P Observation Fidelity on Held-Out Sets and Recursive Predictions

Primary-model one-step confirmation. A separate four-environment cohort supplies 800 transitions, 200 per environment, with identified training and prior-evaluation sources excluded. The fixed primary crossed model improves aggregate one-step F1 from 0.7566 to 0.7624, with increases in SciWorld, TextWorld, and WebShop and equal ALFWorld scores (Table 30). This is a source-held-out evaluation from the released log pool.

Table 30: Primary-model observation fidelity on 800 new-source transitions. Token F1 under native templates. The coefficient-0.5 and zero-coefficient primary models use identical observation restoration and replay.

Environment	No crossed term	Crossed term
ALFWorld	**0.9844**	**0.9844**
SciWorld	0.7493	**0.7584**
TextWorld	0.6898	**0.6929**
WebShop	0.6030	**0.6138**
Environment mean	0.7566	**0.7624**

TextWorld branches with shared histories. A further TextWorld analysis follows two branches for three steps from an identical complete logged prefix, on 200 previously unused source groups. Each model generates 1,200 observations, retaining its own predictions after the fork and receiving the fixed logged actions. The primary crossed model obtains mean token F1 0.6913 versus 0.6854 for its matched control. This TextWorld result measures recursive prediction with a shared visible history, complementing state-only one-step branch evaluation. Source grouping certifies the recorded prefix rather than hidden simulator state.

Q Action Content and Branch Reconstruction

Separating command content from presentation. An action can contain the correct operation and arguments while differing in its surrounding presentation. We analyze the first command in the existing action predictions, ending at the first newline or code-fence boundary; a WebShop command ends at its closing bracket. A reference-grammar parser represents the command as its operation, object or primary argument, and remaining ordered arguments. Object identifiers and qualifiers are preserved, and verbs are not merged as synonyms. A leading "the" in an entity and presentation markers are normalized. When a command cannot be parsed, the comparison falls back to normalized exact matching. This analysis was introduced after inspection of the existing outputs and applies the same rules to every objective.

Table 31: Leading action content on the original action cohort. Environment-average percentages on 1,523 transitions from 400 groups. Action content requires the complete parsed structure to match, with exact-match fallback. Object or main-argument accuracy uses references containing that field. All models use the extended-context coefficient-0.05 objective comparison.

Objective	Action content	Object / main argument
No crossed term	37.473	<u>52.231</u>
Bidirectional crossed	**38.253**	**52.689**
Forward crossed only	37.410	51.805
Inverse crossed only	37.521	52.206
Row-wise ranking	<u>37.613</u>	52.028

The bidirectional objective gives the highest whole-command content score and object accuracy among the five objectives (Table 31). Removing trailing prose therefore retains a content-level advantage: complete structure improves from 37.473% to 38.253% against zero crossed weight, and object or main-argument accuracy improves from 52.231% to 52.689%. These measurements describe the recovered command rather than the surrounding answer format.

Reference-defined branch distinctions. We additionally reconstruct the actions for all 595 branch pairs: the 395 primary pairs and the separate 200-group TextWorld cohort. The references define two nested subsets before inference. In 490 pairs the two actions have different parsed content; in 318 of these pairs, the operation is the same but the object or arguments differ. The 490-pair subset contains 134 ALFWorld, 142 SciWorld, 124 TextWorld, and 90 WebShop pairs. The same-operation subset contains 134, 82, 12, and 90 pairs, respectively. The 1,190 transition references require 957 unique native prompts per model, each decoded once with the existing 128-token action budget. The cohorts supply a new inverse reconstruction measurement on previously evaluated forward branches.

Table 32: Reconstructing branch-specific actions. Environment-average percentages. "Action" averages structural correctness over individual actions; "both" requires both reconstructed actions in a pair to be correct. Subsets depend on reference content, and the primary-pair rows retain all 395 original pairs. All five objectives use coefficient 0.05 where the auxiliary term is present.

Reference-defined set / metric	Zero	Crossed	Forward	Inverse	Row rank
Distinct content: action (490)	69.362	**70.005**	<u>69.482</u>	69.400	69.261
Distinct content: both (490)	<u>53.817</u>	**54.827**	53.781	53.616	53.338
Same type: action (318)	74.150	**74.705**	74.195	<u>74.288</u>	74.150
Same type: both (318)	<u>66.168</u>	**67.002**	65.982	<u>66.168</u>	65.891
Primary pairs: action (395)	72.519	**73.074**	<u>72.725</u>	72.658	72.519
Primary pairs: both (395)	59.354	**60.348**	<u>59.488</u>	59.354	59.076

On the 490 content-distinct pairs, the complete crossed objective reaches 70.005% action correctness and 54.827% both-correct, exceeding each directional and ranking control. Relative to zero weight, the gains are 0.643 and 1.009 percentage points. Requiring both actions to be recovered makes the pair sensitive to a model that returns one plausible action for two different changes. The 318-pair subset further requires object or argument discrimination while holding the operation type

fixed; its both-correct score rises from 66.168% to 67.002%. These are modest aggregate gains, concentrated primarily in WebShop. Within the 490-pair subset, SciWorld action correctness also rises from 56.690% to 57.042%. The action-content analysis complements full-string reconstruction by identifying which part of the command distinguishes the observed branches.

R GOAL-CONDITIONED SELECTION OVER LOGGED OUTCOMES

Candidate outcomes and task progress. We form a goal-conditioned cohort of 409 logged state groups: 99 ALFWorld, 181 SciWorld, and 129 TextWorld. Each group contains two to four recorded actions with their observed outcomes, including at least one progress outcome and one non-progress outcome. Ambiguous action–outcome mappings are excluded before scoring. A binary label $u(o)$ records the environment's success marker or, in TextWorld, its explicit one-point score increase. The task instruction is appended to the current observation. The model scores each candidate outcome under each action using its native template, with a 4,096-token total limit and up to 1,024 target tokens.

Each logged group supplies k actions a_i and outcomes o_j, with $i, j \in \{1, \ldots, k\}$. Let s be the current observation, g the task instruction, and $q_{ij} = p_\theta(o_j \mid s, g, a_i)$ the candidate outcome's sequence likelihood. The utility $u(o_j)$ is one for recorded progress and zero otherwise. Normalizing over the supplied outcomes gives expected progress $V(a_i)$ and selected action a^*:

$$V(a_i) = \frac{\sum_{j=1}^{k} q_{ij} u(o_j)}{\sum_{j=1}^{k} q_{ij}}, \qquad a^* = \arg \max_{a_i : 1 \le i \le k} V(a_i). \tag{15}$$

We report whether the selected action's recorded outcome has positive progress. Stable log-sum-exp differences rank the equivalent success-versus-failure log odds, avoiding probability saturation. A secondary policy replaces summed target log likelihood by its mean per scored token. These are offline choices over known outcome supports; the measured quantity is recorded progress selection rather than online task completion.

Source composition. The 409 groups exclude identified adapter-training sources and observations, as well as the initial sources of the earlier 400- and 374-group action cohorts. Of these groups, 251 sources appeared in earlier evaluations of other tasks. The remaining 158 consist of 148 SciWorld and 10 TextWorld sources. Eighteen groups share the complete logged prefix; the others share visible observations. These properties describe the available conditioning evidence, with no additional hidden-state equivalence assumed. The cohort is used for analysis and coefficient development.

Table 33: Goal-conditioned choices for the five training objectives. Percentages at coefficient 0.05. The first three columns use summed sequence likelihood on SciWorld: all 181 groups, the 148 sources absent from the indexed earlier evaluations, and all 181 groups after removing explicit reward markers from scored outcomes. The last column is the prespecified mean-token policy on all 409 groups, averaged equally over three environments.

Objective	SciWorld (181)	Unused (148)	No markers (181)	Mean-token (409)
No crossed term	86.740	85.811	86.740	73.111
Bidirectional crossed	**88.398**	**87.838**	**87.845**	**73.738**
Forward crossed only	<u>87.293</u>	<u>86.486</u>	<u>87.293</u>	72.884
Inverse crossed only	86.188	85.135	<u>87.293</u>	72.378
Row-wise ranking	86.740	85.811	86.740	<u>73.632</u>

SciWorld provides a direct five-objective comparison: the full crossed model selects a progress outcome in 160 of 181 groups, versus 157 for zero crossed weight and row-wise ranking. Within the 148 previously unused SciWorld sources, it selects 130 progress actions versus 127 for both controls. Removing the explicit success marker from scored outcomes preserves the ordering, with 159 versus 157 correct choices. Utility labels remain fixed during this text removal. The mean-token policy also gives the highest three-environment average to bidirectional supervision, 73.738%. That result concerns the secondary length-normalized policy; the SciWorld columns use complete sequence likelihood.

Selecting the crossed coefficient. We compare the existing extended-context coefficients $\beta \in \{0.05, 0.25, 0.5\}$ using summed-likelihood environment-average progress selection on the 409 development groups. The highest development score selects $\beta = 0.5$; ties would favor the smaller coefficient. The loss formula, initial adapter, data order, update count, restoration, and replay are unchanged. Table 34 compares the selected model with its matched zero-crossed control. Its development mean is 87.280%, versus 81.830% for the control. The coefficient is then fixed for the additional TextWorld comparison.

Additional TextWorld sources. Broader extraction of task instructions from the text preceding the first room heading yields 22 further eligible task sources after excluding identified training and indexed prior-evaluation sources. Reference-only ordering first selects one state per source. The selected coefficient obtains 20 correct choices out of 22, versus 15 for its zero-crossed control. Baseline predictions on these sources were inspected before the selected coefficient was evaluated; alternative coefficients were selected only on the 409-group development cohort.

After this comparison, retaining every eligible state from the same sources yields 35 states. Each task first averages over its states, and the final score weights the 22 tasks equally. The selected model reaches 86.364%, compared with 67.424%; directly pooling states gives 29 versus 23 correct choices. The 13 added states contribute 9 versus 8 correct choices. The 22-state and 35-state results therefore describe overlapping views of the same task sources. State eligibility depends on recorded outcomes and source exclusions rather than model scores.

Table 34: Goal selection with the development-selected coefficient. Percentages for matched zero and coefficient-0.5 extended-context models. Development averages three environments equally. The additional TextWorld comparison uses 22 task sources; the 35-state extension and marker-removal analysis weight these same sources equally. Gains are percentage points.

Evaluation	$\beta = 0$	$\beta = 0.5$	Gain (pp)
Development, three environments (409)	81.830	**87.280**	+5.450
TextWorld, one state per source (22)	68.182	**90.909**	+22.727
TextWorld, all eligible states (35)	67.424	**86.364**	+18.939
TextWorld, reward markers removed (35)	54.545	**62.121**	+7.576

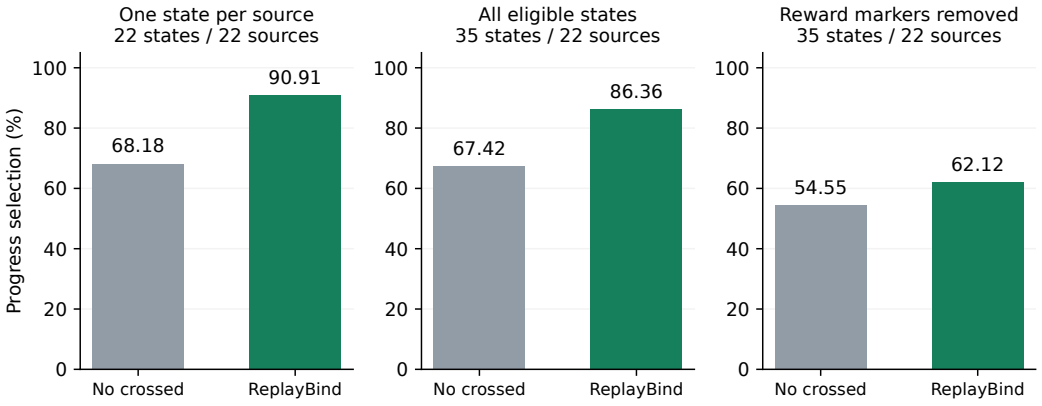


Figure 10: **Goal selection on the additional TextWorld sources.** The fixed coefficient-0.5 model is compared with its matched zero-crossed control. The left panel uses one state per source; the other panels use all 35 eligible states from the same 22 sources and average within each source first. Marker removal changes scored outcome text while retaining original progress labels.

Outcome text without explicit reward markers. As a subsequent analysis of the 35-state cohort, we remove "[SUCCESS]" and the explicit one-point score-increase sentence from every candidate outcome before computing likelihoods. All actions, labels, states, and the selected coefficient remain fixed. Source-balanced progress selection is 62.121% for crossed supervision and 54.545% for zero

weight; the corresponding counts are 22 and 18 correct states. The retained difference shows that explicit reward phrases are not the sole source of the observed choice advantage. This analysis extends the same task sources rather than introducing another independent cohort.