# AUTOVALUEWORLD: LEARNING VALUE DYNAMICS FOR CONSEQUENCE-AWARE DECISION MAKING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In multi-turn interactions, language model assistants can shape users' deliberation and the conditions for subsequent decisions. Responses with similar immediate ratings may support or violate human values differently, changing the relative utility of later actions. Response-level judgments alone don't specify how these value-relevant states evolve under successive responses. We introduce AutoValueWorld, an action-conditioned value world model that separates stable value importance from a dynamic, ten-dimensional value-support state. Training combines procedural supervision of state transitions and simulated reactions with human-rating supervision from PRISM. Candidate-difference supervision learns relative value consequences and recursive state prediction supports a frozen gain-based selector between one-step and three-step policies. Experiments show candidate-difference supervision reduces held-out candidate-contrast RMSE by 13.1% and 13.8% at 3B and 7B. In a study of 5,000 procedural episodes, selective lookahead improves mean realized return over one-step selection for value alignment at 3B and weighted reaction at 7B. These results support explicit value-dynamics modeling and selective lookahead for consequence-aware decision making in procedural environments.

## 1 INTRODUCTION

In multi-turn interactions, assistant responses can shape a user's deliberation and conditions for later decisions (Zhou et al., 2024; Ryu et al., 2026). Consider schematic data-sharing example in Figure 1: recommendations for personalized sharing and minimal collection may receive similar immediate ratings, but support self-direction and security differently. These differences can change relative utility of the same follow-up actions, such as clarifying the sharing scope or confirming privacy settings. We refer to interaction's current support or violation of value concerns as its **value-support state**, distinct from user's underlying value priorities. Contextual value representations make such concerns explicit (Sorensen et al., 2024a; Yao et al., 2024); sequential challenge is to predict how this state evolves under successive responses and how its evolution affects later action choices.

Reward and preference learning use response judgments to guide model behavior (Ouyang et al., 2022; Rafailov et al., 2023), with related work incorporating multiple criteria and individual context (Wang et al., 2024b;a; Kirk et al., 2024). Such judgments don't by themselves specify how responses transform the value-support state. World models predict action-conditioned state transitions (Ha & Schmidhuber, 2018; Hafner et al., 2025; Lin et al., 2024), while reasoning and planning methods compare alternative futures (Hao et al., 2023; Yao et al., 2023; Gkountouras et al., 2025). We study **action-conditioned value dynamics**, modeling how responses transform the current value-support state while value importance remains fixed within an interaction. Recursive prediction then enables comparison of value consequences across candidate response sequences.

Learning and using value dynamics raise two challenges. First, low average state-prediction error need not preserve the **relative value consequences** that guide response selection (Farahmand et al., 2017; Schrittwieser et al., 2020; Hansen et al., 2024). A shared prediction offset can leave a ranking unchanged, whereas an error in a candidate difference can reverse it. Second, predicting further ahead does not by itself ensure better decisions, so lookahead should be guided by its **decision benefit** (Snell et al., 2025; Setlur et al., 2025). This requires identifying when a multi-step policy is expected to improve return over immediate selection.



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.

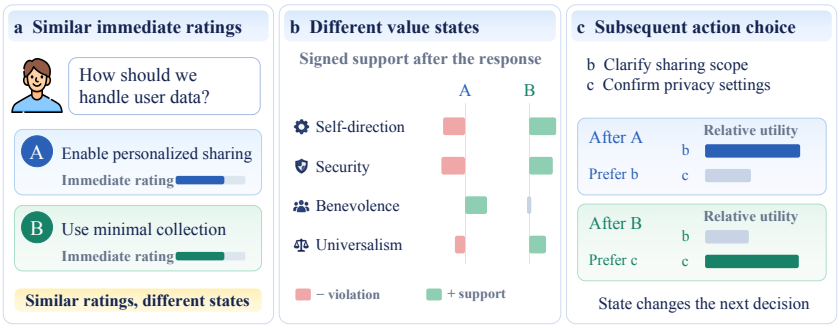


Figure 1: **Similar ratings, different value states, different subsequent choices.** Two responses to the same context leave distinct value-support states. Comparing the same follow-up actions from these states can change their relative utility. The illustration shows selected coordinates of the ten-dimensional value state. Bar lengths and relative utilities are schematic; the same follow-up candidates are compared in both states.

We propose AutoValueWorld, an action-conditioned value world model that predicts how candidate responses change the value-support state. Training combines procedural supervision of value dynamics and simulated reactions with human-rating supervision from PRISM (Kirk et al., 2024), using a shared representation and parallel heads. Candidate-difference supervision matches predicted and target differences between candidates' next value states within each context. Recursive prediction supports one-step comparison (H1) and three-step lookahead (H3). A frozen gain selector chooses H3 when its estimated return gain over H1 is positive and retains H1 otherwise. The system executes the selected first action in the simulator and replans from the resulting observation.

On 1,000 independently procedural worlds, AutoValueWorld recovers action-order effect directions with mean accuracies of 98.0% at 3B and 99.6% at 7B. In comparisons against training without difference loss, candidate-difference supervision reduces held-out state-difference RMSE by 13.1% and 13.8%. In 5,000 new procedural planning episodes, frozen selector improves mean realized return over H1 for 3B value alignment and 7B weighted reaction, with changes to only a small fraction of first actions. Complementary matched-bottleneck experiments show low-weight value-state supervision improve PRISM rating prediction.

Our contributions are as follows:

- We propose AutoValueWorld, a value world model separates stable value importance from dynamic support or violation, recursively predicts action-conditioned value-state transitions.
- We introduce candidate-difference supervision to learn relative value consequences, improving predictions of within-context state differences and strong two-step action-order effects in confirmation studies.
- We develop a frozen gain-based selector between immediate and multi-step policies, showing in a separate procedural study that selective lookahead can improve realized return over H1.

## 2 Related Work

**Human values and personalized evaluation.** Schwartz's theory organizes ten basic human values, and ESS measures their importance through portrait items (Schwartz, 2012; European Social Survey European Research Infrastructure (ESS ERIC), 2026b;a). Value FULCRA and Value Kaleidoscope make multidimensional and contextual value concerns explicit (Yao et al., 2024; Sorensen et al., 2024a), while OpinionQA and ValueBench evaluate opinion representation and value understanding (Santurkar et al., 2023; Ren et al., 2024). Pluralistic alignment, cultural context, and history-based personalization provide complementary perspectives (Sorensen et al., 2024b; Shi et al., 2024; Salemi et al., 2024). We model how responses change value support or violation, distinguishing this dynamic state from stable value importance.

**Reward and preference modeling.** Human-feedback learning and preference optimization use response judgments to guide model behavior (Ouyang et al., 2022; Rafailov et al., 2023). Related

resources provide multi-attribute ratings and preferences (Wang et al., 2024b; 2025b), AI-generated feedback (Cui et al., 2024), and in-situ conversational feedback (Shi et al., 2026). ArmoRM aggregates interpretable reward objectives (Wang et al., 2024a), while RewardBench evaluates reward-model judgments (Lambert et al., 2025). PRISM links ratings to participant context (Kirk et al., 2024), and SynthesizeMe derives personalized reward-model prompts from interaction histories (Ryan et al., 2025). Our model complements response-rating prediction with explicit state transitions: PRISM supervises ratings, while procedural targets supervise value dynamics and simulated reactions.

**World models and planning.** World models connect learned transitions, recursive prediction, and control (Ha & Schmidhuber, 2018; Hafner et al., 2025; Lin et al., 2024), while decision-focused representations and value-aware losses emphasize information relevant to decisions (Schrittwieser et al., 2020; Hansen et al., 2024; Farahmand et al., 2017). Language-based approaches connect world modeling to planning through preconditions and effects, causal variables, and joint preference learning for state prediction and action selection (Xie et al., 2025; Gkountouras et al., 2025; Wang et al., 2025a). Our focus is on relative value consequences: candidate-difference supervision complements absolute state prediction, while a frozen gain selector uses predicted futures to choose between immediate and multi-step policies.

# 3 METHOD

## 3.1 ACTION-CONDITIONED VALUE DYNAMICS

Given interaction context $x_t$, including available history, and a candidate response $a_t$, AutoValue-World predicts the value-support state $\hat{s}_{t+1}$ left by that response. When the predictor is available, it receives value importance $p \in \Delta^9$ and the current state $s_t \in [-1, 1]^{10}$. Their coordinates follow the ten Schwartz value names (Schwartz, 2012): $p$ specifies relative importance and remains fixed within an interaction, while positive and negative entries of $s_t$ represent support and violation under the procedural state convention. The model additionally predicts simulated reactions $\hat{o}_t \in [0, 1]^3$ for acceptability, trust, and continuation, and an observed human rating $\hat{r}_t \in [0, 1]$.

A procedural structural causal model (SCM) generates the value-transition and reaction targets:

$$s_{t+1} = F_\omega(s_t, p, e(a_t)), \qquad o_t = G_\omega(s_t, s_{t+1}, p), \tag{1}$$

where $e(a_t)$ is a designed action-effect vector and $\omega$ specifies the sampled linear, threshold, or hysteresis mechanism. These mechanisms produce state-dependent effects and can generate different endpoints for opposite action orders. The SCM provides complete candidate outcomes and trajectories of up to three steps; its hidden mechanism coefficients aren't available to the predictor. In procedural planning, $p$ and the observed current state are provided and imagined future states are predicted by the model. PRISM supplies complementary human-rating targets (Kirk et al., 2024), rather than value-state transition labels.

## 3.2 LEARNING ABSOLUTE AND RELATIVE VALUE CONSEQUENCES

Figure 2 summarizes the architecture. Model encodes the formatted context and candidate response. Its final non-padding representation $h_t$ is fused with numerical inputs and their availability masks:

$$n_t = [m_p p;\ m_s s_t;\ m_p;\ m_s], \qquad z_t = f_\theta\big([h_t;\ g_\theta(n_t)]\big), \tag{2}$$

where $m_p, m_s \in \{0, 1\}$ indicate whether the corresponding numerical inputs are available. Unavailable numerical inputs are zero-masked. Three parallel heads read the shared representation:

$$\hat{s}_{t+1} = \tanh(W_s z_t + b_s), \quad \hat{o}_t = \sigma(W_o z_t + b_o), \quad \hat{r}_t = \sigma(w_r^\top z_t + b_r). \tag{3}$$

The state head supplies the next state for recursive prediction; the reaction and rating heads read $z_t$ directly.

The base objective combines task-specific mean-squared errors:

$$\mathcal{L}_{\text{base}} = \lambda_s \mathcal{L}_{\text{state}} + \lambda_o \mathcal{L}_{\text{reaction}} + \lambda_r \mathcal{L}_{\text{rating}}. \tag{4}$$

Target-availability masks activate the state and reaction losses for SCM examples and the rating loss for PRISM examples.

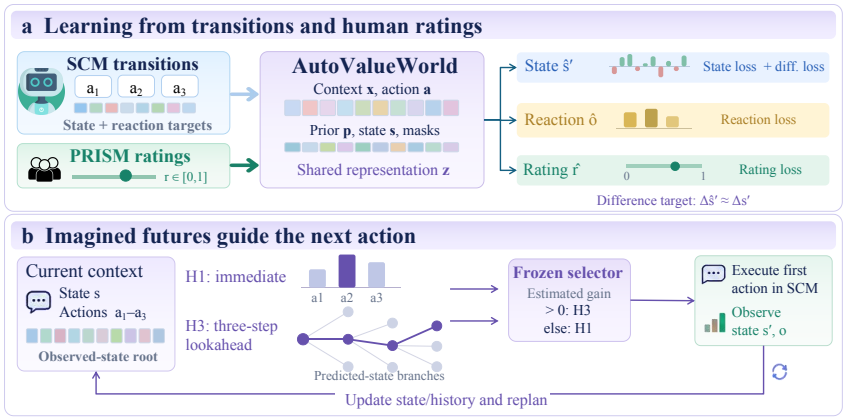


Figure 2: **Learning and planning with value dynamics.** A shared representation feeds parallel state, reaction, and rating heads. SCM targets supervise value dynamics and simulated reactions; PRISM supervises human ratings. Candidate-difference supervision learns relative value consequences. During planning, imagined branches propagate predicted states and reaction histories. A frozen gain selector chooses an immediate or multi-step policy; after execution in the SCM, planning resumes from the observed state and reaction.

To learn relative value consequences, we compare candidates in each synthetic group $g$, holding the context, value importance, and current state fixed. Let $s'_{g,a}$ denote the next-state target for candidate $a$, and let $a < b$ enumerate unordered candidate pairs. We optimize

$$\begin{aligned} \mathcal{L}_{\text{diff}} &= \mathbb{E}_{g,a<b}\left[\left\|(\hat{s}'_{g,a} - \hat{s}'_{g,b}) - (s'_{g,a} - s'_{g,b})\right\|_2^2\right], \\ \mathcal{L} &= \mathcal{L}_{\text{base}} + \lambda_d \mathcal{L}_{\text{diff}}. \end{aligned} \tag{5}$$

A shared additive prediction offset cancels in each difference. The relative objective therefore emphasizes how candidate responses differ in value support, while the absolute state objective anchors the states used in recursive prediction. Complete candidate groups form microbatches, with four groups accumulated per update.

### 3.3 Planning with predicted value dynamics

Procedural planning environment provides 3 candidate actions at each step. We evaluate 2 objectives:

$$u_{o,t} = 0.5o_{t,A} + 0.25o_{t,T} + 0.25o_{t,C}, \qquad u_{s,t} = 0.5 + 0.5p^\top s_{t+1}. \tag{6}$$

During search, the corresponding predicted utility $\hat{u}_t$ is computed from $\hat{o}_t$ or $\hat{s}_{t+1}$. For a candidate action sequence of length $H$, the basic lookahead score is

$$\hat{J}_H(a_{t:t+H-1}) = \sum_{k=0}^{H-1} \gamma^k \hat{u}_{t+k}, \qquad \gamma = 0.9. \tag{7}$$

H1 maximizes immediate predicted utility, whereas basic H3 compares length-three sequences. Each imagined child receives its parent branch's predicted state and a history entry containing the branch action and predicted reactions.

A shallow ExtraTrees gain selector (Geurts et al., 2006) uses model-side H1/H3 scores, predicted advantage, discount and planned strategy identities. Selector study considers six H3 policy variants. Let $\tilde{\pi}_3$ denote H3 policy proposed by selector and $\hat{\Delta}$ estimated return gain over H1. Selection rule is

$$\pi_{\text{sel}} = \begin{cases} \tilde{\pi}_3, & \hat{\Delta} > 0, \\ \pi_1, & \hat{\Delta} \le 0, \end{cases} \tag{8}$$

$\pi_1$ is H1 policy. Selector is fitted on development episodes; features, hyperparameters, and eligible objectives are frozen. Objectives that don't pass development screening retain H1.

Only the first action of a plan is executed. After each execution, subsequent planning starts from state and reaction realized by the SCM, rather than imagined branch outcome. Predictor and selector parameters remain fixed; execution updates current state and interaction history. Appendix G describes planning in the simulator, and Appendix G.1 reports how the selector changes actions.

# 4 Experimental Setup

**Models and training.** We use Qwen2.5-Instruct 3B and 7B (Qwen Team, 2024) with LoRA adapters (Hu et al., 2022). The transition and candidate-difference studies use the jointly tuned 1200 steps, with rank-8 LoRA on Q/K/V/O projections, 384-token inputs, learning rate $10^{-4}$, and state/reaction/rating loss weights 1.25/1.25/1.5. Appendix C gives the full optimization and checkpoint details.

**Data and evaluation splits.** Training combines procedural SCM examples for value-state transitions and simulated reactions with participant-split PRISM ratings (Kirk et al., 2024), normalized to $[0, 1]$. Synthetic examples are split by scenario and PRISM examples by participant. The tuned-model confirmation set contains 160 held-out synthetic trajectories and 42 PRISM participants absent from model selection. Human-rating studies use an unused PRISM reserve and external HelpSteer2 helpfulness annotations (Wang et al., 2024b); ESS portrait-item responses (European Social Survey European Research Infrastructure (ESS ERIC), 2026b;a) are used only to construct population priors. Table 1 summarizes the role of each evaluation source.

Table 1: **Evaluation sets and evidence roles.**

| Source | Evaluation units | Primary role |
|---|---|---|
| State/order SCM | 1,000 worlds | State and order contrasts |
| Planning SCM | 5,000 episodes | Realized policy return |
| PRISM reserve | 18 participants; 662 ratings | Personal-rating transfer |
| HelpSteer2 | 517 prompts; 1,036 ratings | Transfer and calibration |
| ESS R9/R11 | 91,151 complete questionnaires | Population-prior sensitivity |

**Evaluation.** Prediction metrics include value-state, human-rating, candidate-difference, and action-order RMSE. Decision-sensitive diagnostics include action-order direction accuracy, ranking-flip accuracy, and selection regret. Planning is evaluated by realized discounted return under the procedural SCM. Main learned-model results report mean $\pm$ sample SD across seeds; paired uncertainty estimates resample appropriate shared scenario, participant, prompt, or episode units.

**Baselines and controls.** We compare AutoValueWorld with a parameter-matched Direct predictor that removes explicit value-state supervision, structured ridge/MLP and Persistence references for procedural dynamics, and matched bottlenecks that isolate the effect of assigning value semantics to a compact representation (Koh et al., 2020). Component ablations remove counterfactual examples, numerical value inputs, or encoder adaptation. For selective lookahead, a frequency-matched random router preserves H3 usage and policy mixture while randomizing which episodes receive multi-step policies, providing a control for the selector's episode-specific routing. The candidate-difference confirmation compares $\lambda_d = 0.25$ with $\lambda_d = 0$, and planning uses horizon three with discount $\gamma = 0.9$ and a frozen ExtraTrees gain selector (Geurts et al., 2006).

# 5 Results and Analysis

## 5.1 Learning state-dependent value dynamics

We test whether AutoValueWorld captures state dependence and action-order effects in value dynamics. On 1,000 independently procedural worlds, we vary initial value state while holding scenarios and candidates fixed, and compare value-state endpoints after applying the same actions in opposite orders. SCM truth defines 250 strong-state worlds, 250 strong-order worlds, and ranking-reversal cases.

AutoValueWorld captures strong action-order effects reliably. Order-effect RMSE is $0.0717$ at 3B and $0.0736$ at 7B, with $0.1041$ for Persistence, while strong-order direction accuracy reaches $98.0\%$

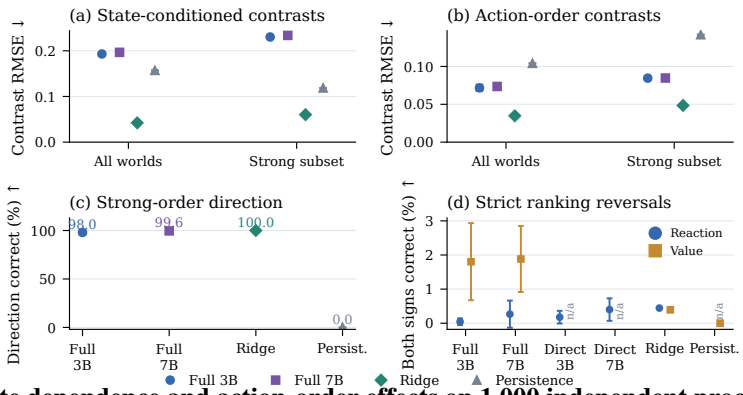


Figure 3: **State dependence and action-order effects on 1,000 independent procedural worlds.** (a,b) Value-state contrast RMSE across all worlds and strong subsets. (c) Direction accuracy for strong action-order effects. (d) Strict ranking-reversal accuracy. Ridge and Persistence are deterministic; Direct has no value-state predictions.

and 99.6%. **State-conditioned differences are more difficult.** State-contrast RMSE is 0.1931 at 3B and 0.1966 at 7B, with 0.1571 for Persistence, and strict value-ranking reversal accuracy remains low at 1.8% and 1.9%. So recovering broad value dynamics doesn't necessarily preserve relative consequences needed for action selection, motivating candidate-difference objective studied next. The procedural prompt used for this experiment is provided in Appendix I.

### 5.2 Learning Relative Value Consequences

The previous experiment shows that recovering broad value dynamics does not necessarily preserve the relative consequences that guide action selection. We therefore test whether candidate-difference supervision improves prediction of how alternative responses differ in the value states they leave behind. Development screening selects $\lambda_d = 0.25$, which is then compared with $\lambda_d = 0$ at both model scales with identical ordered training exposure.

Table 2: **Learning relative value consequences: matched confirmation.** 5 seed mean ± SD. Paired change is $\lambda_d = 0.25$ minus $\lambda_d = 0$; negative values favor lower RMSE. Candidate contrast and H3 state use the disjoint confirmation set, while strong order uses the independent state/order set.

| | | RMSE ↓ | | Paired change ↓ |
|---|---|---|---|---|
| Size | Target | $\lambda_d = 0$ | $\lambda_d = 0.25$ | Difference − baseline |
| 3B | Candidate contrast | 0.1121 ± 0.0082 | **0.0974 ± 0.0120** | -0.0147 ± 0.0049 |
| 3B | H3 state | 0.1477 ± 0.0026 | **0.1434 ± 0.0037** | -0.0042 ± 0.0022 |
| 3B | Strong order | 0.0852 ± 0.0078 | **0.0777 ± 0.0086** | -0.0075 ± 0.0039 |
| 7B | Candidate contrast | 0.1017 ± 0.0056 | **0.0876 ± 0.0027** | -0.0141 ± 0.0033 |
| 7B | H3 state | 0.1431 ± 0.0027 | **0.1400 ± 0.0032** | -0.0031 ± 0.0016 |
| 7B | Strong order | 0.0854 ± 0.0070 | **0.0718 ± 0.0046** | -0.0136 ± 0.0029 |

Candidate-difference supervision substantially improves relative value-state prediction. Candidate-contrast RMSE decreases from 0.1121 to 0.0974 at 3B and from 0.1017 to 0.0876 at 7B, reductions of 13.1% and 13.8%. H3 state RMSE also decreases by 0.0042 and 0.0031, indicating that the improvement persists under recursive prediction.

Benefit extends to composed action effects. Strong-order RMSE decreases 0.0852 to 0.0777 at 3B and 0.0854 to 0.0718 at 7B. In contrast, changes in selection regret are smaller and noisier. So candidate-difference supervision most improves prediction of relative and composed value consequences. The candidate-difference experiment reuses the procedural prompt shown in Appendix I.

### 5.3 Using Predicted Value Dynamics for Decisions

We test whether predicted value futures can improve realized decisions. In a separate planning study, a frozen gain selector is evaluated on 5,000 new procedural episodes, using each checkpoint's H1

policy as the reference. The selector chooses an H3 policy only when model-visible features predict a positive return gain. A frequency-matched random router preserves the selector's H3 usage and policy mixture while randomizing which episodes receive lookahead.

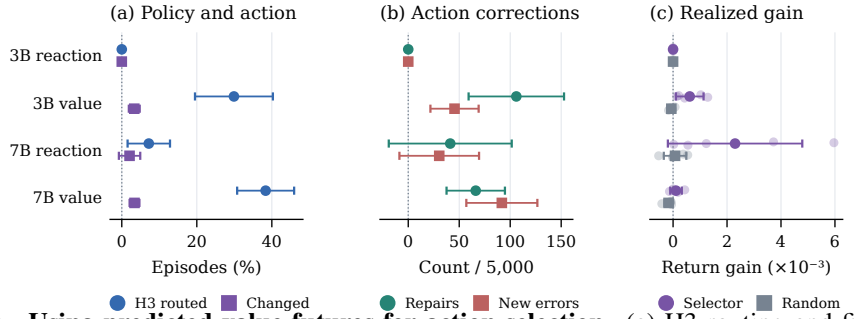


Figure 4: **Using predicted value futures for action selection.** (a) H3 routing and first-action changes. (b) Repairs and introduced errors relative to the SCM oracle-best first-action set. (c) Realized return gain over H1 for the selector and frequency-matched random routing. Markers and whiskers show 5 seeds means and SD.

Selective lookahead improves realized return for two objectives. The selector gains $0.000617$ over H1 for 3B value alignment and $0.002298$ for 7B weighted reaction. H3 is routed on $29.90\%$ and $7.20\%$ of episodes, while only $3.33\%$ and $2.07\%$ of first actions change. Frequency-matched random routing produces near-zero mean gains for same objectives. This suggests that benefit comes from selectively identifying episodes where multi-step prediction is useful, rather than using H3 more often. The recursive rollout prompt used by H1/H3 is provided in Appendix I.

### 5.4 What supports value-state prediction?

We test which components support learned value dynamics. Using the 600-step configuration, we ablate counterfactual training examples, the value prior, the current state, both numerical inputs, or encoder adaptation while holding model size and update budget. Input ablations remove the corresponding information from both numerical and formatted-text channels.

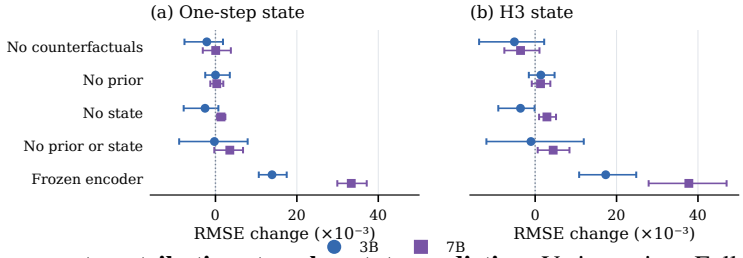


Figure 5: **Component contributions to value-state prediction.** Variant minus Full RMSE in $10^{-3}$ units. Points show 3 seeds paired differences, with marginal 95% intervals.

Encoder adaptation has the most consistent effect. Freezing the encoder increases one-step state RMSE by 0.0139 at 3B and 0.0334 at 7B, and H3 RMSE by 0.0173 and 0.0378. Effects of explicit value inputs are less uniform: removing the current state increases state and H3 error at 7B, but at 3B it lowers H3 error. These results indicate that adapting language representation is important for value-state prediction, while contribution of explicit prior and state inputs depends on model scale. The ablations reuse the procedural prompt, with the documented state and prior fields removed for the corresponding conditions; the prompt contracts are in Appendix I.

### 5.5 Connecting value states to human judgments

We test whether explicit value semantics can help a compact representation predict observed human judgments. We construct a bottleneck architecture in which both models predict ratings through the same ten-dimensional representation and numerical inputs. State-supervised variant aligns this bottleneck with procedural value-state targets but latent variant receives no state labels. Parameter counts, consumed examples, optimization settings, and all non-state targets are matched.

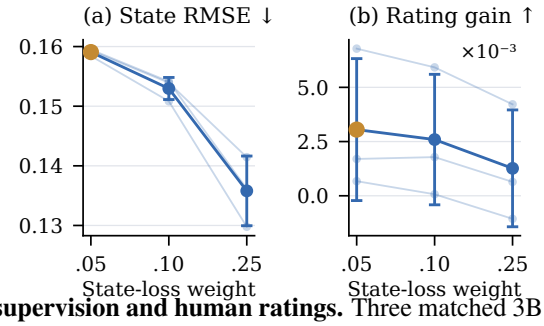


Figure 6: **Value-state supervision and human ratings.** Three matched 3B development seeds. Thin lines denote seeds; markers show mean ± SD. The selected state-loss weight is highlighted.

State reconstruction and rating prediction favor different supervision strengths as shown in Figure 6. Reducing the state-loss weight from 0.25 to 0.05 increases development state RMSE from 0.1358 to 0.1591, but also increases the mean PRISM rating gain from 0.00126 to 0.00305. We freeze weight 0.05 for the rating-oriented comparison. On the participant-disjoint audit, value-state supervision improves rating RMSE by 0.00371 at 3B and 0.00071 at 7B. The clearer 3B result suggests that light structural supervision can help a compact representation retain information useful for observed judgments, while the effect weakens at the larger scale. The PRISM rating prompt used in this experiment is provided in Appendix I.

### 5.6 Human-rating grounding and transfer

We evaluate the rating head that shares the language representation used for value dynamics. The unused PRISM reserve contains 18 participants, 259 interactions, and 662 ratings that were not used for model selection. Full achieves RMSEs of 0.2041 at 3B and 0.2260 at 7B,compared with 0.2793 and 0.2755 for Direct. These correspond to relative error reductions of 26.9% and 18.0%, demonstrating lower mean prediction error on held-out PRISM ratings at both scales.

Table 3: **Human-rating grounding and transfer.** Five-seed mean ± SD with equal participant or prompt-group weighting. HelpSteer2 uses complete fields and 1,024-token inputs. Paired change is Full minus Direct; lower RMSE is better.

| | | Rating RMSE ↓ | | Paired change ↓ |
|---|---|---|---|---|
| Data | Size | Full | Direct | Full − Direct |
| PRISM reserve | 3B | **0.2041 ± 0.0031** | 0.2793 ± 0.0027 | 0.0009 ± 0.0010 |
| PRISM reserve | 7B | **0.2260 ± 0.0027** | 0.2755 ± 0.0018 | 0.0006 ± 0.0010 |
| HelpSteer2, full context | 3B | **0.2178 ± 0.0094** | 0.3243 ± 0.0062 | -0.0004 ± 0.0050 |
| HelpSteer2, full context | 7B | **0.2239 ± 0.0076** | 0.3154 ± 0.0065 | 0.0011 ± 0.0020 |

We further test frozen prediction on HelpSteer2 (Wang et al., 2024b), which shifts the target from PRISM personal satisfaction to annotator helpfulness. On 1,036 responses from 517 prompt groups, Full achieves RMSEs of 0.2178 at 3B and 0.2239 at 7B, compared with 0.3243 and 0.3154 for Direct under the same complete-field input interface. The relative reductions are 32.8% and 29.0%. The advantage extends beyond held-out PRISM ratings to a distinct annotation source and rating target.

Across both datasets, Full has lower mean error than Direct at each scale, supporting the usefulness of the joint value-dynamics model for the evaluated human-rating tasks. The 3B Full model also has lower mean error than its 7B counterpart on both sources, indicating that rating performance doesn't improve monotonically with model size in this comparison. The PRISM and HelpSteer2 prompt contracts are provided in Appendix I.

### 5.7 Input and output interface sensitivity

The Full-Direct differences above motivate examining whether the prediction interface has a larger effect on external rating accuracy. We study three complementary changes: preserving more language context, supplying population-level value priors, and calibrating the output score to the target domain.

**Preserving context.** The original interface truncates each text field at 420 characters and uses a 192-token encoder allocation. An input audit shows that this truncates 875 response fields and 424 prompt fields in HelpSteer2. Removing the character cap and increasing the encoder limit to 1,024 tokens reduces RMSE by 0.067–0.104 across Full and Direct conditions.

Both architectures benefit substantially from the additional context (Figure 7), and the improvement is much larger than their difference under the complete-context interface. Because the field cap and token allocation change together, this comparison evaluates the complete input interface rather than either factor in isolation. The short and complete HelpSteer2 prompt variants are given in Appendix I.

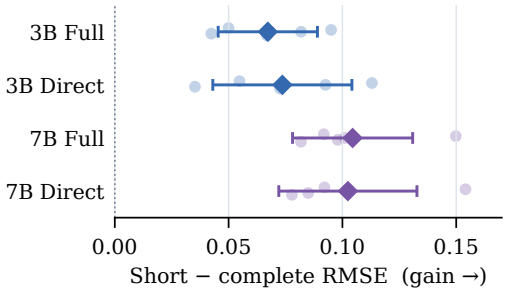


Figure 7: **Context sensitivity on HelpSteer2.** Seed-paired RMSE reduction from the original short interface to complete fields with a 1,024-token encoder limit. Points show individual seeds; markers and whiskers show mean ± SD.

**Population priors.** We construct country-level value-importance priors from ESS rounds 9 and 11 (European Social Survey European Research Infrastructure (ESS ERIC), 2026b;a) and insert them into frozen Full models. On 62 PRISM participants and 3,030 ratings, the country-R9 prior changes RMSE relative to an absent prior by only $(-3.56) \times 10^{-5}$ at 3B and $(-0.72) \times 10^{-5}$ at 7B. The current predictor is therefore largely insensitive to these inserted population priors. The ESS analysis reuses the PRISM prompt and changes only the numerical prior input; the prompt contract is in Appendix I.

**Target-domain calibration.** We test whether frozen HelpSteer2 predictions can be adapted through a simple affine output transformation. A deterministic prompt-group split assigns 258 groups to development and 259 to confirmation; development labels fit affine map.

Table 4: **Target-domain rating calibration.** Five-seed mean ± SD on 259 held-out HelpSteer2 prompt groups. The paired gain is uncalibrated minus calibrated RMSE, reported in $10^{-3}$ units.

| | | Rating RMSE ↓ | | Paired gain ↑ |
|---|---|---|---|---|
| Size | Model | Original | Calibrated | $(U - C) \times 10^3$ |
| 3B | Full | 0.3096 ± 0.0008 | **0.2473 ± 0.0090** | 16.99 ± 8.37 |
| 3B | Direct | 0.3114 ± 0.0007 | **0.2495 ± 0.0059** | 16.89 ± 5.95 |
| 7B | Full | 0.3102 ± 0.0018 | **0.2398 ± 0.0073** | 11.42 ± 5.99 |
| 7B | Direct | 0.3108 ± 0.0020 | **0.2381 ± 0.0063** | 10.40 ± 5.25 |

Calibration reduces mean RMSE in all four model conditions. Full improves from 0.2473 to 0.3096 at 3B and from 0.2398 to 0.3102 at 7B, with similar gains for Direct. Together with the context study, these results indicate that external human-rating accuracy in this setting depends strongly on the input and output interfaces, whereas the inserted population prior contributes little without corresponding numerical-prior supervision. Calibration changes only the output mapping; it reuses the complete-field HelpSteer2 prompt shown in Appendix I.

## 6 Conclusion

We study value-sensitive interaction as a dynamical modeling problem. AutoValueWorld represents the value state left behind by a response and recursively predicts its evolution under subsequent actions. In procedural worlds, candidate-difference supervision improves relative value consequences and composed action-order predictions; a separate planning study converts predicted value futures into realized-return gains through selective lookahead. Complementary human-rating studies identify useful value-state supervision at 3B and assess the shared representation's transfer. Together, these results support modeling both how a response is judged immediately and the value-relevant state from which interaction continues.

## AI USE STATEMENT

We use generative AI tools exclusively for language editing, grammatical checking, and manuscript proofreading. It isn't involved in data analysis, methodological implementation, or conceptual reasoning. It doesn't supply or replace the human eligibility labels. Frozen labels, predictions, thresholds, and confirmatory decision are strictly preserved. The authors are fully responsible for verifying the analyses, interpretations, citations, and final text; AI assistance doesn't substitute for their scientific judgment.

## REFERENCES

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, and Bingxiang He. ULTRAFEEDBACK: Boosting Language Models with Scaled AI Feedback. In *Proceedings of the 41st International Conference on Machine Learning(ICML)*, volume 235, pp. 9722–9744, 2024.

European Social Survey European Research Infrastructure (ESS ERIC). ESS11 - integrated file, edition 4.1, 2026a.

European Social Survey European Research Infrastructure (ESS ERIC). ESS9 - integrated file, edition 3.3, 2026b.

Amir-Massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-Aware Loss Function for Model-based Reinforcement Learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics(AISTATS)*, volume 54, pp. 1486–1494, 2017.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, 2006.

John Gkountouras, Matthias Lindemann, Phillip Lippe, Efstratios Gavves, and Ivan Titov. Language Agents Meet Causality – Bridging LLMs and Causal World Models. In *Proceedings of the 13th International Conference on Learning Representations(ICLR)*, volume 2025, pp. 37311–37336, 2025.

David Ha and Jürgen Schmidhuber. World Models, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025.

Nick Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, Robust World Models for Continuous Control. In *Proceedings of the 12th International Conference on Learning Representations(ICLR)*, volume 2024, pp. 47376–47405, 2024.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, and Zhen Wang. Reasoning with Language Model is Planning with World Model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pp. 8154–8173, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, and Yuanzhi Li. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the 10th International Conference on Learning Representations(ICLR)*, 2022.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, and Katerina Margatina. The PRISM Alignment Dataset: What Participatory, Representative and Individualised Human Feedback Reveals About the Subjective and Multicultural Alignment of Large Language Models. In *Advances in Neural Information Processing Systems(NIPS)*, volume 37, pp. 105236–105344, 2024.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, and Emma Pierson. Concept Bottleneck Models. In *Proceedings of the 37th International Conference on Machine Learning(ICML)*, volume 119, pp. 5338–5348, 2020.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, and Bill Yuchen Lin. RewardBench: Evaluating Reward Models for Language Modeling. In *Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025(NAACL)*, pp. 1755–1797, 2025.

Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, and Pieter Abbeel. Learning to Model the World With Language. In *Proceedings of the 41st International Conference on Machine Learning(ICML)*, volume 235, pp. 29992–30017, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, and Carroll L. Wainwright. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems(NIPS)*, volume 35, 2022.

Qwen Team. Qwen2.5: A party of foundation models, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, and Stefano Ermon. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems(NIPS)*, volume 36, 2023.

Yuanyi Ren, Haoran Ye, Hanjun Fang, Xin Zhang, and Guojie Song. ValueBench: Towards Comprehensively Evaluating Value Orientations and Understanding of Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)(ACL)*, pp. 2015–2040, 2024.

Michael J. Ryan, Omar Shaikh, Aditri Bhagirath, Daniel Frees, and William Held. SynthesizeMe! Inducing Persona-Guided Prompts for Personalized Reward Models in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)(ACL)*, pp. 8045–8078, 2025.

Jeongwoo Ryu, Soomin Kim, Jinsu Eun, Kyusik Kim, and Changhoon Oh. Feeling Right vs. Being Right: How AI Sycophancy Affects Value-Laden Deliberation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)(ACL)*, pp. 44227–44245, 2026.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When Large Language Models Meet Personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)(ACL)*, pp. 7370–7392, 2024.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, and Percy Liang. Whose Opinions Do Language Models Reflect? In *Proceedings of the 40th International Conference on Machine Learning(ICML)*, volume 202, pp. 29971–30004, 2023.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, and Laurent Sifre. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020.

Shalom H. Schwartz. An overview of the schwartz theory of basic values. *Online Readings in Psychology and Culture*, 2(1), 2012.

Amrith Setlur, Nived Rajaraman, Sergey Levine, and Aviral Kumar. Scaling Test-Time Compute Without Verification or RL is Suboptimal. In *Proceedings of the 42nd International Conference on Machine Learning(ICML)*, volume 267, pp. 54058–54094, 2025.

Taiwei Shi, Zhuoer Wang, Longqi Yang, Ying-Chun Lin, and Zexue He. WildFeedback: Aligning LLMs With In-situ User Interactions And Feedback. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)(ACL)*, pp. 36701–36725, 2026.

Weiyan Shi, Ryan Li, Yutong Zhang, Caleb Ziems, and Sunny Yu. CultureBank: An Online Community-Driven Knowledge Base Towards Culturally Aware Language Technologies. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024(EMNLP)*, pp. 4996–5025, 2024.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *Proceedings of the 13th International Conference on Learning Representations(ICLR)*, volume 2025, pp. 10131–10165, 2025.

Taylor Sorensen, Liwei Jiang, Jena D. Hwang, Sydney Levine, and Valentina Pyatkin. Value Kaleidoscope: Engaging AI with Pluralistic Human Values, Rights, and Duties. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(18):19937–19947, 2024a.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L Gordon, and Niloofar Mireshghallah. Position: A Roadmap to Pluralistic Alignment. In *Proceedings of the 41st International Conference on Machine Learning(ICML)*, volume 235, pp. 46280–46302, 2024b.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024(EMNLP)*, pp. 10582–10592, 2024a.

Siyin Wang, Zhaoye Fei, Qinyuan Cheng, Shiduo Zhang, and Panpan Cai. World Modeling Makes a Better Planner: Dual Preference Optimization for Embodied Task Planning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)(ACL)*, pp. 21518–21537, 2025a.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, and Gerald Shen. HelpSteer2: Open-source dataset for training top-performing reward models. In *Advances in Neural Information Processing Systems(NIPS)*, volume 37, pp. 1474–1501, 2024b.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, and Gerald Shen. HelpSteer2-Preference: Complementing Ratings with Preferences. In *Proceedings of the 13th International Conference on Learning Representations(ICLR)*, volume 2025, pp. 56548–56575, 2025b.

Kaige Xie, Ian Yang, John Gunerli, and Mark Riedl. Making Large Language Models into World Models with Precondition and Effect Knowledge. In *Proceedings of the 31st International Conference on Computational Linguistics(COLING)*, pp. 7532–7545, 2025.

Jing Yao, Xiaoyuan Yi, Yifan Gong, Xiting Wang, and Xing Xie. Value FULCRA: Mapping Large Language Models to the Multidimensional Spectrum of Basic Human Value. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)(NAACL)*, pp. 8762–8785, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, and Thomas L. Griffiths. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems(NIPS)*, volume 36, 2023.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, and Haofei Yu. SOTOPIA: Interactive Evaluation for Social Intelligence in Language Agents. In *Proceedings of the 12th International Conference on Learning Representations(ICLR)*, 2024.

## APPENDIX CONTENTS


**A The Use of Large Language Models** **14**

**B Value states and simulation rules** **14**

**C Data and training settings** **15**
C.1 Data pools and evaluation samples 15
C.2 Training settings and model versions 15

**D Evaluation metrics and result summaries** **16**
D.1 Prediction and ranking metrics 16
D.2 Averaging results and comparing models 16

**E Results by model size and baseline** **16**

**F State, action order, and candidate differences** **16**
F.1 Generating the state and action-order tests 16
F.2 Training on candidate differences 17

**G Planning in the simulator** **18**
G.1 How the selector changes actions 19

**H ESS value priors and survey rounds** **19**
H.1 Questionnaire scoring and sample selection 19
H.2 Changing priors with fixed model weights 20
H.3 Country coverage and differences between rounds 20

**I Prompts** **21**
I.1 P1: Procedural-world prompt 21
I.2 P2: Closed-loop rollout prompt 21
I.3 P3: PRISM human-rating prompt 21
I.4 P4: HelpSteer2 original-interface prompt 22
I.5 P5: HelpSteer2 complete-context prompt 22

# A THE USE OF LARGE LANGUAGE MODELS

We used generative AI tools to improve wording and grammar and to help with LaTeX formatting. The authors are responsible for checking the final text, references, figures, tables, and files. The reported results come from the existing experiments; AI-generated text isn't used as human annotation or as independent experimental evidence.

# B VALUE STATES AND SIMULATION RULES

**The ten value dimensions.** We use this fixed order: self-direction, stimulation, hedonism, achievement, power, security, conformity, tradition, benevolence, and universalism. The stable importance vector gives nonnegative weights that sum to one. The changing value state describes how much the current situation supports or violates each value under the simulator's rules. It does not measure how strongly a person endorses that value. A scenario may set up a trade-off, such as self-direction versus conformity; this does not imply that those values always conflict.

**How actions change the state.** Let $e_j$ be the assigned effect of action $a$ on value $j$, and let $C$ be the fixed matrix describing how value dimensions interact. For each world, the simulator samples persistence $\rho \in [0.45, 0.78]$, action strength $\alpha \in [0.42, 0.72]$, interaction with the value profile $\beta \in [0.08, 0.24]$, interaction across values $\gamma \in [0.03, 0.11]$, threshold $\tau \in [0.20, 0.42]$, and outcome temperature $T \in [0.55, 0.85]$. The next state is

$$s'_j = \tanh\left(\rho_j s_j + \alpha \tilde{e}_j + \beta(10p_j - 1)\tilde{e}_j + \gamma \sum_k C_{jk} e_k\right). \tag{9}$$

In the linear family, $\tilde{e}_j = e_j$ and $\rho_j = \rho$. Here, "linear" describes the action transformation before $\tanh$. The threshold family uses $\tilde{e}_j = 1.35\,\mathrm{sign}(e_j)\max(0, |e_j| - \tau)$.

The hysteresis family makes negative states harder to repair. When $s_j < 0$, persistence becomes $\min(0.90, \rho{+}0.08)$, positive repair becomes $\max(0, e_j{-}0.30|s_j|)$, and negative effects are multiplied by 1.12. Other coordinates use the linear rule. Once a world and its inputs are fixed, the expected transition is deterministic. Randomness enters through world and profile sampling and the specified outcome distributions.

**Simulated reaction targets.** We define alignment as $A = \sum_j p_j s'_j$, violation as $V = \sum_j p_j \max(0, -s'_j)$, repair as $Q = \sum_j p_j \max(0, s'_j - s_j)$, and disruption as $D = \sum_j p_j |s'_j - s_j|$. Acceptability and shifted trust have centers

$$c_A = \mathrm{clip}_{[0,4]}\{2 + 1.72\tanh(2.55A - 0.65V)\}, \tag{10}$$

$$c_T = \mathrm{clip}_{[0,4]}\{2 + 1.55\tanh(2.15A + 0.55Q - 0.45D - 0.80V)\}. \tag{11}$$

Continuation is $q = \sigma[-0.20 + 1.25(c_A - 2) + 0.85(c_T - 2)]$. The model predicts the three scalar targets $(c_A/4, c_T/4, q)$. The simulator stores soft ordinal distributions around the two centers, but these aren't the model's outputs.

**Generating scenarios and actions.** The simulator generates 4,000 scenarios with four profiles each. A trajectory has three steps with probability 0.35 and one step otherwise. Actions are sampled using an acceptability softmax with temperature 0.2 and exploration probability 0.05. We store labels for every candidate action, including actions not taken. The selected action determines the next recorded step. Scenarios are assigned to training, validation, and test using proportions 0.75/0.15/0.10. The table below gives the resulting example counts. Simulated reactions are kept separate from observed human ratings.

The prompt for these simulated examples is in Appendix I.

Table 5: Processed example counts before sampling for individual runs. These are available data pools, not the number of examples used by each run.

| Split | Simulator action examples | PRISM rating examples |
|---|---|---|
| Train | 60,870 | 52,576 |
| Validation | 12,102 | 6,651 |
| Test | 8,190 | 6,900 |

# C Data and training settings

## C.1 Data pools and evaluation samples

We split synthetic examples by scenario and PRISM examples by participant. Each record identifies whether its target comes from the simulator or a human rating. For each task, reservoir sampling selects at most 8,000 training examples; a seed controls their shuffle order. This gives 16,000 eligible examples per run. Runs of 150 and 600 steps use 600 and 2,400 examples, respectively. The CF0 condition removes unobserved-action examples before sampling, so it can change which synthetic examples are selected.

The original sampled test contains 141 synthetic scenarios and 95 PRISM participants, including 431 rating records. One-step and rollout evaluations use fixed subsets of the test pools in Table 5. For paired comparisons, we resample whole scenarios and participants so that related actions and interactions remain together. Keeping all responses from an interaction allows candidate ranking against observed ratings. These PRISM comparisons are observational.

**Model inputs.** Simulator inputs contain the scenario, domain, stakes, value trade-off, value weights, current state, available history, and candidate action. PRISM inputs contain available participant information (locale, residence, region, age, education, English proficiency, and stated preferences), up to two earlier turns, the current user message, and a candidate response. History uses the highest-rated earlier responses, so this offline setting assumes access to earlier feedback. The current response's rating is used only as a prediction target.

In the original input format, whitespace is normalized and each field is limited to 420 characters. The encoder then retains beginning and ending tokens within a 192-token budget.

## C.2 Training settings and model versions

Table 6: Selected 3B/7B training settings, checked against saved run and checkpoint records. The learning-rate schedule and optimizer settings support reproduction.

| Setting | Value |
|---|---|
| Backbones | Qwen2.5-3B-Instruct; Qwen2.5-7B-Instruct |
| LoRA targets / rank / alpha | q_proj and v_proj / 4 / 8 |
| LoRA and fusion dropout | 0.05 |
| Numerical projection / fusion width | 96 / 192 |
| Learning rate / weight decay | $1.5 \times 10^{-4}$ / 0.01 |
| Optimizer / schedule | AdamW / cosine with warmup |
| Warmup fraction | 0.06 |
| Batch size / accumulation | 1 / 4 |
| State / outcome / rating loss weights | 1 / 1 / 1 |
| Original encoder length | 192 tokens |

The prompts for simulated examples and human ratings are in Appendix I.

# D EVALUATION METRICS AND RESULT SUMMARIES

## D.1 PREDICTION AND RANKING METRICS

Value-state RMSE is the square root of the mean squared error across synthetic examples and all ten value coordinates. Reaction RMSE is calculated separately for each normalized target. H3 feeds predicted value states forward through three recorded actions and compares them with the simulator's original states. Other input context is kept as recorded.

Candidate top-1 accuracy uses groups with at least two responses. It checks whether the first candidate with the highest predicted score is also the first with the highest target score, in the stored order.

For simulator acceptability, we measure error in the score difference between every candidate pair:

$$E_{\mathrm{CF}} = \sqrt{\frac{1}{N_{\mathrm{pairs}}} \sum_{g} \sum_{a<b} [(\hat{o}_{g,a,A} - \hat{o}_{g,b,A}) - (o_{g,a,A} - o_{g,b,A})]^2}. \tag{12}$$

**The original combined score.** Let $R_s, R_A, R_T, R_C, R_P$ denote RMSE for state, acceptability, trust, continuation, and PRISM ratings. Let $U_A, U_P$ denote synthetic and PRISM top-1 accuracy, $E_A, E_P$ their calibration errors, and $\bar{R}_H$ the mean H1–H3 state RMSE. The original development score is

$$\begin{aligned} P = {} & R_s + 0.5R_A + 0.25R_T + 0.25R_C + 0.5(1 - U_A) \\ & + R_P + 0.5(1 - U_P) + 0.5E_{\mathrm{CF}} + 0.25E_A + 0.25E_P + 0.25\bar{R}_H. \end{aligned} \tag{13}$$

This score combines prediction errors and ranking penalties with chosen weights. We retain it to document the earlier model selection and also report its components. Direct predictors have no state or rollout outputs, so they are not compared using $P$.

Full and Direct are compared on their shared outputs using

$$M_{\mathrm{shared}} = \tfrac{1}{3}(R_A^2 + R_T^2 + R_C^2) + R_P^2. \tag{14}$$

## D.2 AVERAGING RESULTS AND COMPARING MODELS

For the additional human-rating studies, $c = 1, \ldots, C$ indexes participants or normalized prompts, and $n_c$ is their number of responses. We calculate each seed's RMSE and then average across seeds:

$$R_k = \sqrt{\frac{1}{C} \sum_{c=1}^{C} \frac{1}{n_c} \sum_{i=1}^{n_c} (\hat{r}_{kci} - r_{ci})^2}, \qquad \bar{R} = \frac{1}{5} \sum_{k=1}^{5} R_k. \tag{15}$$

# E RESULTS BY MODEL SIZE AND BASELINE

Table 7: Results by model size and training budget, shown as mean ± sample SD over five seeds. Lower Primary and RMSE are better; higher top-1 is better.

| Model / steps | Primary | State | H3 | PRISM | Top-1 |
|---|---|---|---|---|---|
| 0.5B / 150 | 1.226±0.040 | 0.164±0.003 | 0.210±0.005 | 0.292±0.003 | 0.470±0.021 |
| 3B / 150 | 1.220±0.026 | 0.162±0.001 | 0.204±0.004 | 0.288±0.007 | 0.444±0.013 |
| 7B / 150 | 1.224±0.035 | 0.159±0.003 | 0.198±0.003 | 0.288±0.004 | 0.450±0.028 |
| 3B / 600 | 1.128±0.014 | 0.139±0.005 | 0.181±0.008 | 0.271±0.004 | 0.493±0.019 |
| 7B / 600 | 1.099±0.055 | 0.113±0.007 | 0.148±0.007 | 0.270±0.005 | 0.478±0.068 |

# F STATE, ACTION ORDER, AND CANDIDATE DIFFERENCES

## F.1 GENERATING THE STATE AND ACTION-ORDER TESTS

We generated the state and action-order test. We create 1,000 new worlds: 335 with hysteresis, 321 linear, and 344 threshold mechanisms. Each world starts from three conditions, with the focal

Table 8: Numerical and structured baselines, with the first three seeds used for language models. Parentheses give the number of training examples matched exactly across methods. "Pool" baselines use different amounts of data and their own optimization. Numerical baselines do not use language representations.

| Predictor | Primary | State | H3 | PRISM | Top-1 |
|---|---|---|---|---|---|
| 3B / 150 | 1.2071 | 0.1618 | 0.2032 | 0.2851 | 0.4465 |
| 7B / 150 | 1.2113 | 0.1575 | 0.1983 | 0.2847 | 0.4633 |
| 3B / 600 | 1.1204 | 0.1358 | 0.1767 | 0.2694 | 0.4822 |
| 7B / 600 | 1.0701 | 0.1096 | 0.1445 | 0.2670 | 0.5136 |
| Structured ridge (2,400) | 0.8325 | 0.0418 | 0.0522 | 0.3013 | 0.4591 |
| Structured ridge (600) | 0.8849 | 0.0487 | 0.0696 | 0.3019 | 0.4591 |
| Numeric MLP (pool) | 1.2283 | 0.1416 | 0.1954 | 0.3013 | 0.4591 |
| Numeric ridge (pool) | 1.2328 | 0.1452 | 0.1950 | 0.3012 | 0.4591 |
| Persistence | 1.2682 | 0.1608 | 0.2575 | 0.3012 | 0.4591 |
| Structured MLP (pool) | 0.8327 | 0.0424 | 0.0533 | 0.3013 | 0.4591 |
| Structured ridge (pool) | 0.8270 | 0.0403 | 0.0487 | 0.3012 | 0.4591 |

value coordinate set to $+0.55$, $0$, or $-0.55$ to represent support, neutrality, or violation. Each checkpoint makes 9,000 one-step candidate predictions. Full checkpoints also make 18,000 two-action predictions, covering both orders of each distinct strategy pair.

Table 9: Effects of initial state and action order on 1,000 new worlds. Strong-effect subsets are defined from simulator truth before inference.

| Model | State diff. RMSE ↓ | | Action order | | | Decision |
|---|---|---|---|---|---|---|
| | All | Strong | RMSE all ↓ | Strong ↓ | Direction (%) ↑ | Flip (%) ↑ |
| Full 3B | $0.193 \pm 0.001$ | $0.230 \pm 0.001$ | $0.072 \pm 0.004$ | $0.085 \pm 0.003$ | $98.0 \pm 2.4$ | $1.8 \pm 1.1$ |
| Full 7B | $0.197 \pm 0.001$ | $0.234 \pm 0.001$ | $0.074 \pm 0.005$ | $0.085 \pm 0.004$ | $99.6 \pm 0.7$ | $1.9 \pm 1.0$ |
| Structured ridge | 0.042 | 0.060 | 0.035 | 0.048 | 100.0 | 0.4 |
| Persistence | 0.157 | 0.119 | 0.104 | 0.142 | 0.0 | 0.0 |

Table 10: Candidate selection as initial states change. Top-1 and ranking-flip accuracy are percentages; lower regret is better. Direct models predict reactions but do not output value states.

| Model | Objective | Top-1 all (%) | Regret all | Top-1 strong (%) | Flip acc. (%) |
|---|---|---|---|---|---|
| Full 3B | Reaction | $33.9 \pm 0.0$ | $0.0259 \pm 0.0000$ | $35.2 \pm 0.0$ | $0.0 \pm 0.1$ |
| Full 3B | Value | $63.8 \pm 2.2$ | $0.0029 \pm 0.0003$ | $67.1 \pm 1.5$ | $1.8 \pm 1.1$ |
| Full 7B | Reaction | $34.0 \pm 0.2$ | $0.0268 \pm 0.0020$ | $35.7 \pm 1.0$ | $0.3 \pm 0.4$ |
| Full 7B | Value | $65.1 \pm 1.0$ | $0.0027 \pm 0.0002$ | $68.5 \pm 2.1$ | $1.9 \pm 1.0$ |
| Direct 3B | Reaction | $33.9 \pm 0.1$ | $0.0260 \pm 0.0002$ | $35.2 \pm 0.1$ | $0.2 \pm 0.2$ |
| Direct 7B | Reaction | $33.9 \pm 0.1$ | $0.0261 \pm 0.0002$ | $35.2 \pm 0.2$ | $0.4 \pm 0.3$ |
| Structured ridge | Reaction | 76.3 | 0.0040 | 76.9 | 0.4 |
| Structured ridge | Value | 80.2 | 0.0011 | 79.5 | 0.4 |

### F.2 Training on candidate differences

The candidate-difference loss averages squared error over all unordered action pairs in each synthetic group. It varies $\lambda_d \in \{0, 0.05, 0.10, 0.25\}$.

For confirmation, we retrain the selected weight and $\lambda_d = 0$ at 3B and 7B with 5 seeds. Evaluation uses separate confirmation participants and trajectories, plus the independent state and order set.

Weight 0.25 lowers state-difference RMSE from $0.10793$ to $0.09566$ and selection regret from $0.01252$ to $0.01195$. Mean state and H3 RMSE also improve, while PRISM RMSE changes from $0.25110$ to $0.25227$.

Table 11 adds decision and rating measures to Table 2. One-step state RMSE falls by $0.00836$ at 3B and $0.00799$ at 7B. Selection regret changes by $-0.000224$ and $-0.000150$, while PRISM RMSE changes by $+0.00100$ and $-0.00021$. Strong-state RMSE remains close to baseline. Strict value-ranking flip accuracy is still low: $2.51\%$ and $3.69\%$. Thus, better prediction of relative state changes does not yet show a comparably large improvement in decisions.

The state, action-order, and candidate-difference tests use the simulated-input and rollout prompts in Appendix I.

Table 11: Additional confirmation results. Values are mean ± sample SD over five seeds. Paired changes are selected $\lambda_d$ minus $\lambda_d = 0$. Lower errors and regret are better; higher accuracy and realized return are better.

| Size | Test | Metric | $\lambda_d = 0$ | Selected $\lambda_d$ | Paired change |
|---|---|---|---|---|---|
| 3B | Disjoint | Mean regret ($\times 10^{-3}$) | 14.323 ± 2.375 | 14.099 ± 1.539 | -0.224 ± 1.000 |
| 3B | Disjoint | State RMSE | 0.0843 ± 0.0052 | 0.0759 ± 0.0085 | -0.0084 ± 0.0038 |
| 3B | Disjoint | PRISM RMSE | 0.2745 ± 0.0011 | 0.2755 ± 0.0012 | 0.0010 ± 0.0012 |
| 3B | Disjoint | Reaction return | 0.54588 ± 0.00487 | 0.54613 ± 0.00328 | 0.00024 ± 0.00185 |
| 3B | Disjoint | Value return | 0.53527 ± 0.00022 | 0.53548 ± 0.00035 | 0.00020 ± 0.00026 |
| 3B | State/order | State dependence RMSE | 0.2308 ± 0.0008 | 0.2310 ± 0.0011 | 0.0002 ± 0.0007 |
| 3B | State/order | Value flip accuracy | 1.92 ± 0.97 | 2.51 ± 0.64 | 0.59 ± 0.84 |
| 7B | Disjoint | Mean regret ($\times 10^{-3}$) | 14.482 ± 1.857 | 14.331 ± 2.128 | -0.150 ± 0.450 |
| 7B | Disjoint | State RMSE | 0.0769 ± 0.0049 | 0.0690 ± 0.0019 | -0.0080 ± 0.0036 |
| 7B | Disjoint | PRISM RMSE | 0.2739 ± 0.0017 | 0.2737 ± 0.0034 | -0.0002 ± 0.0024 |
| 7B | Disjoint | Reaction return | 0.54549 ± 0.00356 | 0.54558 ± 0.00415 | 0.00009 ± 0.00100 |
| 7B | Disjoint | Value return | 0.53536 ± 0.00025 | 0.53556 ± 0.00034 | 0.00021 ± 0.00016 |
| 7B | State/order | State dependence RMSE | 0.2325 ± 0.0009 | 0.2324 ± 0.0012 | -0.0000 ± 0.0012 |
| 7B | State/order | Value flip accuracy | 3.41 ± 0.33 | 3.69 ± 0.49 | 0.27 ± 0.70 |

## G PLANNING IN THE SIMULATOR

The planning set contains 5,000 simulated episodes from the same generator distribution and one profile per scenario. Each episode has three actions available at each of three steps, discount $\gamma = 0.9$, and no recorded action history. Model inputs are stored separately from simulator states and oracle labels. Before loading any checkpoint, we evaluate all $3^3 = 27$ action sequences with the simulator to obtain the oracle results.

For each objective, a "trap" episode is one where the best H3 plan beats every H3 plan starting with an H1-optimal action by more than $10^{-12}$. Such episodes test whether a good immediate action can limit later return. "Natural" results use all episodes; "trap" results use this harder subset.

Full H3 evaluates both objectives and replans after each executed action. During search, a child branch receives the parent's predicted value state and a history entry with that branch's action and predicted outcomes. After execution, the next search starts from the state and outcome produced by the simulator.

Full H1 uses the same execution loop and chooses the best immediate action. Direct H1 predicts only weighted reaction. Structured ridge is tested at H1 and H3 for both objectives. Random selection, the simulator oracle, and the three fixed strategies do not require model inference.

The primary comparison is Full H3 minus Full H1 realized return at each size and objective. The table below also reports Direct H1 and structured ridge on the full set and the trap subsets.

Table 12 gives each policy's return, regret relative to the oracle, and first-action accuracy. It includes comparisons that favor a baseline or H1. Natural rows use all 5,000 episodes. Trap rows use the objective-specific subsets defined from simulator truth before evaluation.

Table 12: Planning results averaged across available seeds. Return is realized discounted utility. Regret is the gap to the best sequence found by the simulator oracle. Hit is the fraction of first actions that are oracle-best.

| System | Policy | Obj. | Split | $N$ | Return | Regret | Hit |
|---|---|---|---|---|---|---|---|
| 3B Direct | H1 | WR | Natural | 5000 | 1.49716 | 0.07061 | 0.3654 |
| 3B Direct | H1 | WR | Trap | 164 | 1.50673 | 0.00170 | 0.7476 |
| 7B Direct | H1 | WR | Natural | 5000 | 1.49382 | 0.07395 | 0.3644 |
| 7B Direct | H1 | WR | Trap | 164 | 1.50030 | 0.00813 | 0.7476 |
| 3B Full | H1 | VA | Natural | 5000 | 1.43597 | 0.02809 | 0.4111 |
| 3B Full | H1 | VA | Trap | 252 | 1.43916 | 0.00687 | 0.8619 |
| 3B Full | H1 | WR | Natural | 5000 | 1.49172 | 0.07605 | 0.3635 |
| 3B Full | H1 | WR | Trap | 164 | 1.49997 | 0.00846 | 0.7207 |
| 3B Full | H3 | VA | Natural | 5000 | 1.43550 | 0.02856 | 0.4013 |
| 3B Full | H3 | VA | Trap | 252 | 1.43925 | 0.00678 | 0.8738 |
| 3B Full | H3 | WR | Natural | 5000 | 1.48710 | 0.08067 | 0.3612 |
| 3B Full | H3 | WR | Trap | 164 | 1.49685 | 0.01159 | 0.7073 |
| 7B Full | H1 | VA | Natural | 5000 | 1.44725 | 0.01681 | 0.5563 |

| System | Policy | Obj. | Split | $N$ | Return | Regret | Hit |
|---|---|---|---|---|---|---|---|
| 7B Full | H1 | VA | Trap | 252 | 1.43666 | 0.00937 | 0.6222 |
| 7B Full | H1 | WR | Natural | 5000 | 1.48245 | 0.08532 | 0.3630 |
| 7B Full | H1 | WR | Trap | 164 | 1.48597 | 0.02247 | 0.6927 |
| 7B Full | H3 | VA | Natural | 5000 | 1.44612 | 0.01794 | 0.5332 |
| 7B Full | H3 | VA | Trap | 252 | 1.43770 | 0.00833 | 0.6857 |
| 7B Full | H3 | WR | Natural | 5000 | 1.47937 | 0.08840 | 0.3610 |
| 7B Full | H3 | WR | Trap | 164 | 1.48383 | 0.02460 | 0.6841 |
| Ridge | H1 | VA | Natural | 5000 | 1.45774 | 0.00632 | 0.7484 |
| Ridge | H1 | VA | Trap | 252 | 1.43464 | 0.01139 | 0.3690 |
| Ridge | H1 | WR | Natural | 5000 | 1.54707 | 0.02070 | 0.6807 |
| Ridge | H1 | WR | Trap | 164 | 1.48962 | 0.01881 | 0.4512 |
| Ridge | H3 | VA | Natural | 5000 | 1.45778 | 0.00628 | 0.7453 |
| Ridge | H3 | VA | Trap | 252 | 1.43469 | 0.01134 | 0.3743 |
| Ridge | H3 | WR | Natural | 5000 | 1.54217 | 0.02560 | 0.6003 |
| Ridge | H3 | WR | Trap | 164 | 1.49549 | 0.01294 | 0.6280 |

### G.1 How the selector changes actions

For each confirmation episode, we record whether the selector uses H3, whether H3 changes the H1 first action, and whether that action is among the simulator oracle's best first actions. A repair replaces an incorrect H1 action with an oracle-best action. An introduced error replaces an oracle-best H1 action with an inferior one.

The random-routing control uses 2,000 randomizations with fixed seeds within each model size, training seed, and objective. It preserves the selector's number of H3 episodes and mix of H3 policy variants. This tests whether choosing the right episodes helps beyond using H3 equally often.

Figure 4 in the main text reports routing rates, action changes, repairs, introduced errors, and return gains.

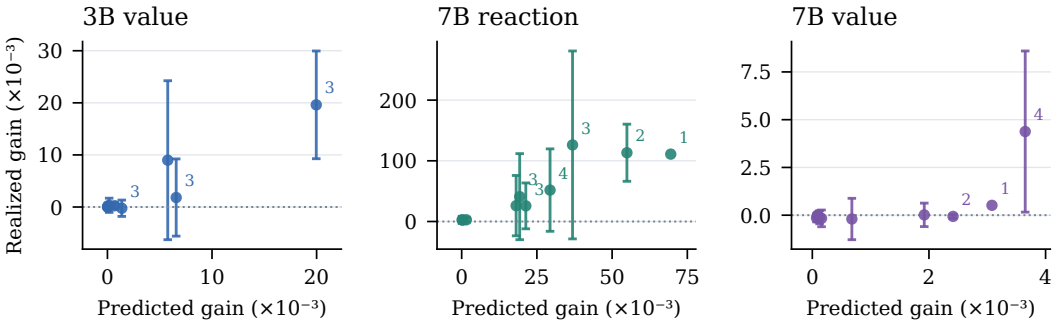


Figure 8: Predicted and realized gains on episodes assigned to H3. Within each seed, episodes are grouped by predicted-gain quantiles; points average the available group results across seeds. Error bars show sample SD.

Planning uses the simulated-input and recursive rollout prompts in Appendix I. The gain selector takes numerical features.

## H ESS value priors and survey rounds

### H.1 Questionnaire scoring and sample selection

ESS9 edition 3.3 and ESS11 edition 4.1 contain human responses to portrait-value questionnaires. We require valid answers to all 21 items, remove responses that give every item the same score, and keep country/round groups with at least 200 complete respondents.

For respondent $i$, each item $x_{ij}$ is reversed as $7 - x_{ij}$ and centered by subtracting that respondent's 21-item mean. Averaging centered items for value $v$ gives $u_{iv}$. The model input vector and country

prior are

$$p_{iv} = \frac{\exp(u_{iv})}{\sum_{w=1}^{10} \exp(u_{iw})}, \qquad p_{cv} = \frac{\sum_{i \in c} w_i p_{iv}}{\sum_{i \in c} w_i}, \quad w_i = \texttt{pspwght}_i. \tag{16}$$

We apply softmax with temperature one to each respondent before averaging within countries. This normalizes model inputs; it is not a validated alternative to standard PVQ scoring. Within-country weights account for survey design and poststratification. The pooled prior gives each country equal weight, so it does not estimate a population-weighted global distribution.

Table 13: ESS questionnaire items for each value, in the model's fixed value order. Renamed fields in later rounds are checked before this mapping is applied.

| Value | ESS items |
|---|---|
| Self-direction | ipcrtiv, impfree |
| Stimulation | impdiff, ipadvnt |
| Hedonism | ipgdtim, impfun |
| Achievement | ipshabt, ipsuces |
| Power | imprich, iprspot |
| Security | impsafe, ipstrgv |
| Conformity | ipfrule, ipbhprp |
| Tradition | ipmodst, imptrad |
| Benevolence | iphlppl, iplylfr |
| Universalism | ipeqopt, ipudrst, impenv |

### H.2 Changing priors with fixed model weights

The prior test uses 62 PRISM test participants and 3,030 ratings, including examples used in earlier evaluations. At each size, we compare the Full model with an R9 country prior against three alternatives: no prior, an R9 prior averaged equally across countries, and randomly reassigned R9 country priors. Model weights are fixed and were not trained on measured numerical PRISM value priors. A country match supplies group information only; it does not measure the participant's own values or current state.

Table 14: Rating RMSE with no prior and with an ESS country-R9 prior: mean $\pm$ sample SD over five fixed Full checkpoints. The final column gives country minus absent in $10^{-5}$ units.

| Size | Absent prior RMSE | ESS country prior RMSE | Country $-$ absent ($\times 10^5$) |
|---|---|---|---|
| 3B | $0.2715 \pm 0.0030$ | $0.2715 \pm 0.0029$ | $-3.56 \pm 6.95$ |
| 7B | $0.2681 \pm 0.0013$ | $0.2681 \pm 0.0013$ | $-0.72 \pm 9.77$ |

Table 15: ESS country-R9 prior minus each control. Differences are multiplied by $10^5$. Each participant has equal weight. Tests use 4,000 paired draws of participants and seeds, with Holm correction across six comparisons.

| Size | Country R9 minus | $\Delta$ RMSE $\times 10^5$ | $p_{\text{Holm}}$ |
|---|---|---|---|
| 3B | absent | $-3.564$ | 1.0 |
| 3B | pooled9 | $-0.027$ | 1.0 |
| 3B | shuffled9 | $+0.051$ | 1.0 |
| 7B | absent | $-0.717$ | 1.0 |
| 7B | pooled9 | $-0.009$ | 1.0 |
| 7B | shuffled9 | $-0.062$ | 1.0 |

All six Holm-adjusted approximate $p$-values are 1.0. This result concerns how the current model uses inserted group priors.

### H.3 Country coverage and differences between rounds

The retained complete responses total 45,187 in R9 and 45,964 in R11, covering 29 and 30 countries, with 27 countries in both.

The two survey rounds sample different people, so differences between their country priors do not measure changes within individuals. Keeping only complete responses, changes in sampled

populations, survey mode and question routing, and incomplete accounting for survey-design variance all limit comparisons over time. These results don't attribute any change to AI interaction.

The ESS test uses the PRISM prompt and changes only the numerical prior. The prompt is in Appendix I.

# I PROMPTS

This section lists the prompts used for simulated state prediction, recursive rollout, PRISM ratings, and HelpSteer2 transfer. Fields in braces are filled at inference time. The examples contain no original participant records or benchmark conversations.

## I.1 P1: PROCEDURAL-WORLD PROMPT

**P1 procedural-world prompt**

```
Task: predict the next normative value state and reaction distribution.

Scenario: {scenario_text}
Domain: {domain}; stakes: {stakes}; focal trade-off: {value_a} versus
    {value_b}.
Individual stable values: {stable_values_in_canonical_order}
Current signed value state: {pre_state_in_canonical_order}
History: {synthetic_history_or_none}
Candidate action: {candidate_action_text}
```

## I.2 P2: CLOSED-LOOP ROLLOUT PROMPT

P2 keeps the scenario, domain, trade-off, stable values, and candidate action from P1. During rollout, it inserts the predicted state and uses the history field shown below:

**P2 closed-loop rollout prompt**

```
Task: predict the next normative value state and reaction distribution.

Scenario: {scenario_text}
Domain: {domain}; stakes: {stakes}; focal trade-off: {value_a} versus
    {value_b}.
Individual stable values: {stable_values_in_canonical_order}
Current signed value state: v0={s0:.3f}, v1={s1:.3f}, v2={s2:.3f}, v3
    ={s3:.3f}, v4={s4:.3f}, v5={s5:.3f}, v6={s6:.3f}, v7={s7:.3f}, v8={
    s8:.3f}, v9={s9:.3f}
History: prior rollout outcomes hidden; current state supplied
    explicitly
Candidate action: {candidate_action_text}
```

## I.3 P3: PRISM HUMAN-RATING PROMPT

**P3 PRISM rating prompt**

```
Task: predict this participant's subjective rating of a candidate
    assistant response.
Participant context: locale={locale}; residence={residence_country_iso
    }; region={region}; age={age}; education={education}; English
    proficiency={english_proficiency}; stated preferences=[{
    stated_preferences}].
History: {prism_history_or_none}
Current user message: {user_prompt}
Candidate assistant response: {candidate_response}
```

### I.4 P4: HELPSTEER2 ORIGINAL-INTERFACE PROMPT

**P4 HelpSteer2 prompt: original interface**

```
Task: predict this participant's subjective rating of a candidate
    assistant response.
Participant context: locale=unknown; residence=unknown; region=unknown
    ; age=unknown; education=unknown; English proficiency=unknown;
    stated preferences=[].
History: none
Current user message: {normalized_helpsteer2_prompt}
Candidate assistant response: {normalized_helpsteer2_response}
```

The original input format normalizes whitespace and limits each field to 420 characters before tokenization.

### I.5 P5: HELPSTEER2 COMPLETE-CONTEXT PROMPT

**P5 HelpSteer2 prompt: complete context**

```
Task: predict this participant's subjective rating of a candidate
    assistant response.
Participant context: locale=unknown; residence=unknown; region=unknown
    ; age=unknown; education=unknown; English proficiency=unknown;
    stated preferences=[].
History: none
Current user message: {all_normalized_prompt_text}
Candidate assistant response: {all_normalized_response_text}
```

The longer-input format removes the 420-character limit and uses an encoder limit of 1,024 tokens. ESS tests reuse P3 and change only the numerical prior.