When Does Knowing the State Help? Diagnosing Process vs. Outcome Reward Design

Wenpei Shao*
Center for Healthy Minds
University of Wisconsin–Madison
wshao33@wisc.edu

Ross Jacobucci
Center for Healthy Minds
University of Wisconsin–Madison
jacobucci@wisc.edu

Abstract

Process reward models (PRMs) dramatically outperform outcome reward models (ORMs) for math reasoning and agent tasks, yet provide no benefit for chat and QA. No theory explains why, and practitioners choose between them by expensive trial and error. We introduce *state-lift*, a diagnostic that screens whether a PRM is worth training for a new domain—in minutes, from $\sim$500 labeled steps, before committing to expensive reward model training. State-lift measures how much step quality depends on trajectory state versus action text alone, and an accompanying *effective gap* distinguishes two regimes: state-noise-dominant (PRM essential) and action-gap-dominant (ORM sufficient), with a decision threshold derived from a regret bound. State-lift is interpretable as an action-ranking quantity only when quality labels vary within a state; a one-pass, model-free audit checks this precondition and separates preference data (HH-RLHF: every state contrastive) from trajectory-labelled negotiation data (DealOrNoDeal: none). Across ten domains, a minutes-scale linear estimate is compared with full Llama-3.1-8B reward model training under matched multi-seed protocols: a positive reading is informative, whereas a near-zero reading is not, because probes cannot recover computation-dependent state-dependence that end-to-end training does. The resulting protocol—audit, screen, and a short truncated-training check when the screen abstains—turns state-lift into an actionable PRM-vs-ORM workflow.

1 Introduction

Every group building reinforcement-learning agents for sequential tasks eventually faces the same design decision: should the reward model evaluate each step of the trajectory (a *process reward model*, PRM) or only the final output (an *outcome reward model*, ORM)? The answer has so far been determined by expensive trial and error. Lightman et al. [17] showed that PRMs dramatically outperform ORMs for mathematical reasoning (+5.8–17.8%). AgentPRM [4] found +6–19% for web agents. Yet Lee et al. [13] found PRMs provide no advantage on MMLU-Pro, and standard RLHF for chat uses ORMs [21]. No theory explains why, and no diagnostic predicts the answer for a new domain.

The key insight is that step quality can be state-dependent in two fundamentally different ways. Consider "multiply both sides by 2." Whether this is correct depends on the current equation—that is *state-dependent quality*. But the nature of this dependence matters. If general features of the state ("a simplification step in an equation with fractions") and the action ("multiplies by an integer") suffice, the dependence is *representation-level*: detectable from states and actions encoded separately. If only a joint reading of the specific tokens suffices ("the denominator is 2, and the

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

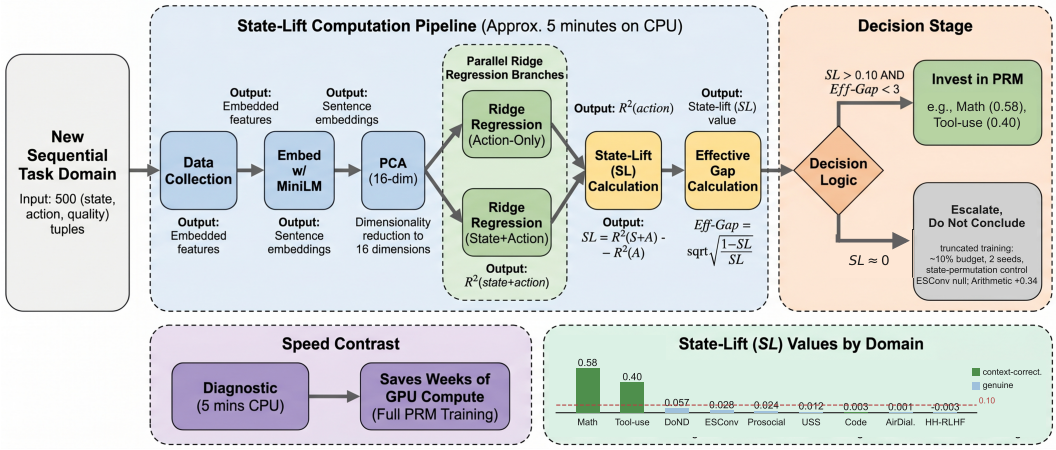

Figure 1: **State-Lift computation pipeline.** A practitioner with a new sequential task domain collects $\sim$500 labeled steps (after the one-pass admissibility check of Section 4.3), embeds them with a sentence encoder, and runs two Ridge regressions (action-only vs. state+action). The state-lift $\mathrm{SL} = R^2(s{+}a) - R^2(a)$ and effective gap $\sqrt{(1{-}\mathrm{SL})/\mathrm{SL}}$ are read one-directionally: $\mathrm{SL} > 0.10$ supports investing in a PRM, while a near-zero reading is escalated to a short truncated-training check rather than read as "ORM sufficient" (Section 6). The diagnostic runs in $\sim$5 minutes on CPU. Bars: corrected registry (Table 2).

action multiplies by 2"), it is *interaction-level*: it requires computation over the pair and is invisible to any encode-separately method.

We formalize this distinction through a framework built around **state-lift**: the fraction of reward variance attributable to trajectory state (Section 2). The diagnostic targets *novel domains* where the PRM–ORM design decision is genuinely unknown—enterprise workflows, specialized agents, new task formats. We present an empirical study across ten sequential decision-making domains with matched reward-model training. We show that: 1) state-lift, computable in minutes from sentence embeddings via Ridge regression, is a *one-directional* screen against matched multi-seed Llama-3.1-8B reward-model training on seven domains: readings above the derived threshold come only from context-correctness labels, where lifts are large ($+0.47$–0.50 AUC), while no genuine-label domain clears it and their lifts are small; 2) state-lift is interpretable as an action-ranking quantity only when labels vary within a state—a precondition that a one-pass, model-free audit checks, and that separates HH-RLHF (every state contrastive) from DealOrNoDeal (no state contrastive), which is why their training gains are not comparable; and 3) the diagnostic has a well-characterized boundary at *computation-dependent* state-dependence, including fine-grained math step grading, where a near-zero reading must be escalated rather than trusted. We characterize this boundary on four math step-grading datasets with eight encoding architectures: all probe-style measurements yield $\mathrm{SL} \in [-0.27, 0.04]$ while end-to-end training achieves $+0.238$ AUC lift on PRM800K (Appendix G).

We show under a Gaussian model (Theorem 1) that state-lift connects to the PRM–ORM performance gap, and an *effective gap* between action-mean spread and state noise predicts two regimes that match our empirical patterns. Together, these results yield a concrete practitioner protocol (Section 6): audit, screen, and escalate when the screen abstains. Throughout, the comparison is between a *prefix-blind* action scorer and a *prefix-conditioned* one; this is the analytical core of the PRM–ORM question but narrower than RLHF usage, where an "outcome" model typically sees the prompt and the full response, and our claims are scoped accordingly. More broadly, our findings suggest that the PRM advantage is not one phenomenon but two structurally distinct classes—representation-decomposable and interaction-irreducible—with different implications for reward model design (Section 6).

2 Problem Formulation

2.1 Setup

Consider a sequential decision process with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$ with $|\mathcal{A}| = K$, and a true step-level reward function $R(s, a)$. A *prefix-blind* scorer (our formal stand-in for an outcome reward model, ORM) scores actions without access to state: $r_{\text{blind}}(a) = \mathbb{E}_s[R(s, a)]$. A *prefix-conditioned* scorer (process reward model, PRM) scores actions conditioned on state: $R(s, a)$. We use ORM/PRM as shorthand for this contrast. We decompose the reward as

$$R(s, a) = r_{\text{blind}}(a) + \delta(s, a), \tag{1}$$

where $\delta(s, a) = R(s, a) - \mathbb{E}_s[R(s, a)]$ is the state-dependent deviation. By construction, $\mathbb{E}_s[\delta(s, a) \mid a] = 0$ for all a: the deviation averages to zero over states for any fixed action.

2.2 State-Lift

We define **state-lift** as the fraction of reward variance attributable to state-dependence:

$$\text{SL}(R) \;=\; \frac{\mathbb{E}_a[\text{Var}_s(R \mid a)]}{\text{Var}(R)} \;=\; \frac{\mathbb{E}[\delta(s, a)^2]}{\text{Var}(R)}. \tag{2}$$

By the law of total variance, $\text{Var}(R) = \text{Var}_a(r_{\text{blind}}(a)) + \mathbb{E}_a[\text{Var}_s(R \mid a)]$, so $\text{SL} \in [0, 1]$. State-lift equals zero if and only if $R(s, a) = r_{\text{blind}}(a)$ for all s, a—that is, the reward is state-independent and an ORM is exact. State-lift equals one when $r_{\text{blind}}(a)$ is constant and all reward variance comes from state-dependence.

2.3 Connection to Existing Concepts

State-lift quantifies a specific type of reward misspecification [32]: the ORM ignores state. The decomposition parallels a natural distinction in sequential domains between *state-dependent quality*—where the same action text can be correct or incorrect depending on trajectory history (e.g., "multiply by 2" in math reasoning)—and *state-independent quality*—where action text alone determines value (e.g., factual correctness in QA). State-lift operationalizes this distinction as a single computable number for any sequential domain.

3 Why State-Lift? A Regret-Bound Analysis

Before turning to empirics, we present a stylized analysis that motivates state-lift and explains the two-regime structure observed in our experiments. Let $\pi^*(s) = \arg\max_a R(s, a)$ be the state-aware optimal policy and $\pi_{\text{blind}} = \arg\max_a r_{\text{blind}}(a)$ be the state-blind policy that always selects the action with highest average reward. We analyze the per-step expected regret $\mathbb{E}_s[R(s, \pi^*(s)) - R(s, \pi_{\text{blind}})]$ under a Gaussian model. The model is stylized, but it captures the essential structure and correlates tightly ($r = 0.993$) with simulated regret (see Section 7 for scope).

Theorem 1 (State-Blind Regret Bound)**.** *Let $R(s, a) = \mu_a + \sigma \cdot Z_a(s)$ where $\mu_a = r_{\text{blind}}(a)$, $Z_a(s)$ are independent standard normal random variables, and $\sigma^2 = \text{SL} \cdot \text{Var}(R)$. Let $\Delta = \mu_{(1)} - \mu_{(2)}$ be the gap between the best and second-best action means. Then:*

(i) $\text{SL} = 0 \implies \pi_{\text{blind}} = \pi^*$ *almost surely.*

(ii) ***Upper bound:*** $\mathbb{E}[\text{regret}] \le \sqrt{\text{SL} \cdot \text{Var}(R)} \cdot \sqrt{2 \ln K}$.

(iii) ***Gap-dependent lower bound (exact for $K = 2$):***

$$\mathbb{E}[\text{regret}] \;\ge\; \sigma\sqrt{2}\,\phi\left(\frac{\Delta}{\sigma\sqrt{2}}\right) - \Delta\,\Phi\left(\frac{-\Delta}{\sigma\sqrt{2}}\right), \tag{3}$$

where ϕ and Φ are the standard normal PDF and CDF.

(iv) ***Two regimes:***

- $\sigma \gg \Delta$ *(state noise dominates):* $\mathbb{E}[\text{regret}] \approx \sigma \cdot \sqrt{2 \ln K}$ *[PRM essential]*

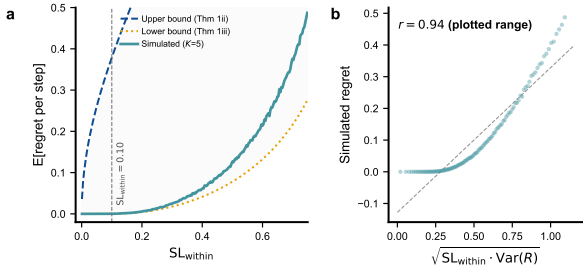

Figure 2: Regret of state-blind reward as a function of $\mathrm{SL}_{\mathrm{within}}$, with upper bound (Theorem 1(ii)), gap-dependent lower bound (Theorem 1(iii)), and simulated values on synthetic MDPs ($K = 5$ actions). The bound is tight for high state-lift and conservative for low state-lift where action gaps dominate.

- $\sigma \ll \Delta$ *(action gap dominates):* $\mathbb{E}[\text{regret}] = O\left(\sigma\, e^{-\Delta^2/4\sigma^2}\right)$ *[ORM sufficient]*

Proof sketch. Part (i) is immediate: if $\mathrm{SL} = 0$, then $\sigma = 0$ and $R(s,a) = \mu_a$, so the blind policy is optimal. Part (ii) follows from Jensen's inequality ($\max$ is convex) and the well-known bound $\mathbb{E}[\max_i Z_i] \le \sqrt{2\ln K}$ for K standard normals. Part (iii) reduces to the two-action subproblem where the exact regret formula is $\sigma\sqrt{2}\,\phi(z) - \Delta\,\Phi(-z)$ with $z = \Delta/(\sigma\sqrt{2})$ (the expected positive part of a $N(-\Delta, 2\sigma^2)$ variable), which provides a lower bound because it considers only one way the ranking can flip. Part (iv): for small z the formula approaches $\sigma\sqrt{2}\,\phi(0) \approx 0.56\sigma$ (noise-dominant regime); for large z the expansion $\Phi(-z) = \phi(z)(1/z - 1/z^3 + \cdots)$ makes the leading terms cancel, leaving $\mathbb{E}[\text{regret}] \sim \sigma\sqrt{2}\,\phi(z)/z^2 = O(\sigma e^{-\Delta^2/4\sigma^2})$ (gap-dominant regime). Full proofs are in Appendix C. □

Remark (what enters the regret). Write $\delta(s,a) = m(s) + g(s,a)$ with $m(s) = \mathbb{E}_a[\delta(s,a)]$ and $\mathbb{E}_a[g(s,a) \mid s] = 0$, so that $\mathrm{SL} = \mathrm{SL}_{\text{level}} + \mathrm{SL}_{\text{within}}$ with $\mathrm{SL}_{\text{level}} = \mathrm{Var}(m)/\mathrm{Var}(R)$ and $\mathrm{SL}_{\text{within}} = \mathbb{E}[g^2]/\mathrm{Var}(R)$. A level effect $m(s)$ shifts every action at a state by the same amount, so it cancels in $\mathbb{E}_s[R(s,\pi^*(s)) - R(s,\pi_{\text{blind}})]$; Theorem 1 holds verbatim with $\sigma^2 = \mathrm{SL}_{\text{within}} \cdot \mathrm{Var}(R)$ (Appendix C). The theorem thus concerns whether state changes *which action is best*; the two components can be told apart only if some states are observed with more than one action (Section 4.3).

Effective gap. We define the *effective gap* as

$$\textsc{Eff-Gap} = \sqrt{\frac{1-\mathrm{SL}}{\mathrm{SL}}} = \frac{\mathrm{std}(r_{\text{blind}})}{\sigma}, \tag{4}$$

which is the ratio of action-mean spread to state-dependent noise. When $\textsc{Eff-Gap} < 1$, state noise dominates and PRM is essential; when $\textsc{Eff-Gap} > 3$, action gaps dominate and ORM is sufficient. The decision threshold used throughout, $\mathrm{SL} > 0.10$, is $\textsc{Eff-Gap} = 3$ solved for SL; it is fixed by the bound, not fitted to training results, and is conservative: with two actions $\mathrm{std}(r_{\text{blind}}) = \Delta/2$, so $\textsc{Eff-Gap} = 3$ is $\Delta = 6\sigma$, where the regret (3) is below 10^{-5} of its ceiling.[2]

Numerical validation. We validate the bound on synthetic MDPs by sweeping state-lift from 0 to 0.99 with $K = 5$ actions (Figure 2). The upper bound holds at all tested points. The correlation between $\sqrt{\mathrm{SL} \cdot \mathrm{Var}(R)}$ and simulated regret is $r = 0.993$ ($p < 10^{-9}$), confirming that $\sqrt{\text{state-lift}}$ is an excellent predictor of regret magnitude. The effective gap determines the regret regime (Figure 7): under context-correctness labels, math reasoning ($\textsc{Eff-Gap} = 1.11$) falls in the noise-dominant regime while code reasoning ($\textsc{Eff-Gap} = 11.91$) falls in the gap-dominant regime.

[2]Our author response described the regret at $\mathrm{SL} = 0.10$ as "$\approx 1\%$ of its noise-dominant ceiling"; that estimate read the gap as $\Delta = \textsc{Eff-Gap} \cdot \sigma$ rather than $2\,\textsc{Eff-Gap} \cdot \sigma$. The threshold is unchanged.

Table 1: Admissibility audit (model-free, one pass). "Both labels" is the fraction of states observed with at least one positive and one negative action. Trajectory-labelled dialogue corpora broadcast one outcome to every turn, so within a state the label never varies and $\mathrm{SL}_{\mathrm{within}}$ is undefined on them; preference data is admissible by construction.

Domain	Label type	States	Items/state	Both labels	ICC
Chat QA (HH-RLHF)	preference pair	6,500	2.00	100.0%	0.000
Verifiable constraints (constructed)	verifier	10	876	100.0%	0.000
Arithmetic (constructed)	verifier	2,796	2.15	100.0%	0.000
Emotional support (ESConv)	strategy progress	54	537	77.8%	0.025
PRM800K (paired subset)	human step rating	1,567	1.11	5.0%	0.894
Negotiation (CaSiNo)	deal outcome	6,634	1.10	2.8%	0.912
Persuasion (PersuasionForGood)	donation outcome	6,760	1.39	0.1%	0.997
Bargaining (CraigslistBargain)	deal outcome	12,426	1.01	0.1%	0.998
Negotiation (DealOrNoDeal)	deal outcome	10,657	1.00	**0.0%**	**1.000**

4 Computing State-Lift

4.1 Quality Signals from the Environment

State-lift requires step-level quality labels. In many domains these are available cheaply from the environment (verifier outcomes, task success, deal values, existing annotations; Appendix B, Table 4), and a few hundred labeled steps suffice for an estimate, orders of magnitude fewer than reward-model training requires.

4.2 Computation Pipeline

Given n tuples (s_i, a_i, y_i) where y_i is the quality label:

1. Embed states and actions with a sentence encoder (we use all-MiniLM-L6-v2).
2. Reduce to $D = 16$ dimensions via PCA.
3. Fit two Ridge regression models predicting y from (a) action features alone, and (b) state + action + interaction features.
4. Compute $\mathrm{SL} = R^2_{\text{conditioned}} - R^2_{\text{action}}$.

Two caveats hold: this is an R^2 difference under a fixed probe class, not the variance fraction of (2), so it can be negative and its magnitude moves with the encoder; and when labels are trajectory-level (one outcome broadcast to every turn), folds must be grouped by trajectory, otherwise the probe learns to recognise the dialogue rather than the action. All values here use trajectory-grouped folds where the label is trajectory-level (the reviewed version's code used pair-level splits for the negotiation domains; turn-level rows are unaffected, Section 5).

4.3 Admissibility: When State-Lift Can Be Read as an Action-Ranking Quantity

By the Remark in Section 3, $\mathrm{SL}_{\mathrm{within}}$ is identified only if some states are observed with more than one action carrying different labels. This is checkable in one pass with no model: per state, count the actions and distinct labels; report the fraction of states carrying both labels and the label ICC, $\mathrm{Var}_{\text{between-state}}(y)/\mathrm{Var}(y)$ (Table 1).

The domains separate with no middle ground. HH-RLHF sits at ICC 0.000 with every state contrastive; DealOrNoDeal at ICC 1.000 with none: its final score is median-split and assigned to every turn, so within-dialogue label variance is zero by construction, the within-state probe has no scorable fold, and a fixed-state best-of-N evaluation cannot be built (0 of 2,134 held-out states). On such data a measured state-lift is a between-trajectory quantity, and a reward model trained on it learns to score contexts rather than rank actions within one. The audit is Step 0 of the protocol (Section 6).

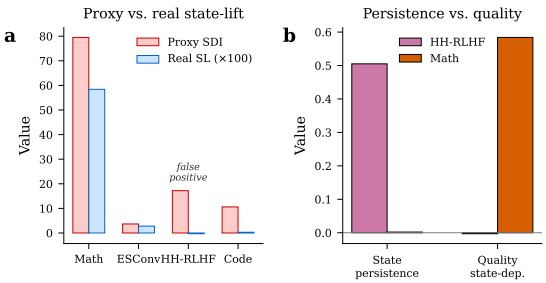

Figure 3: The improvement proxy pitfall. (a) Proxy-based SDI vs. real state-lift (scaled $\times 100$): HH-RLHF shows a massive false positive (proxy=17.2, real=-0.005). (b) Root cause: HH-RLHF has high state persistence (autocorrelation=0.272) but zero state-dependent quality. The proxy conflates the two.

4.4 Obtaining Labels in a New Domain

The diagnostic assumes $\sim$500 step-level labels. Prefer environment-derived signals that need no annotation (verifier outcomes, task success, transaction values), and gather several labelled candidates at the same state, since one candidate per state cannot express the comparison however many labels there are (Table 1). Two free negative controls: permute the state–action pairing (state-lift should collapse; Appendix E), and check whether the label is predictable from the state *alone*, which flags the base-rate case that inflated our own early estimates.

4.5 The Improvement Proxy Pitfall

A natural approach is to bypass quality labels entirely by using an "improvement proxy": scoring each action by how much it moves the embedding state toward the population centroid. This proxy has a critical failure mode. On HH-RLHF it produces a state-lift estimate of 17.2—suggesting a massive PRM advantage—when the true state-lift from preference labels is -0.005. The proxy detects *state persistence* (autocorrelation in conversational states) rather than *state-dependent quality*: computing state-lift on HH-RLHF with *lagged* states (z_{t-1}) yields $\mathrm{SL} = 0.37$, because conversations with better early turns have better responses throughout (Appendix K). State-lift must be computed with respect to current-state quality dependence, from environment-derived or human labels; embedding heuristics in persistent-state domains would lead to wasted PRM investment. Figure 3 illustrates the failure mode.

5 Experiments

5.1 State-Lift Across Domains

We compute state-lift on ten domains spanning reasoning, agent tasks, dialogue, code, negotiation and question answering, plus DealOrNoDeal for validation (Appendix B), using step-level quality signals derived from the local context (annotation-based or structure-based). Table 2 reports the results.

Readings above threshold come only from labels that are state-dependent by construction. Among genuine labels the registry spans -0.003 to 0.057: state-lift as measured here separates the context-correctness mechanism cleanly, but on genuine labels it is a screen whose negative direction Section 5 tests directly. On DealOrNoDeal the value is stable under segmentation (full prefix 0.069 under the control's estimator, last two turns 0.058, last turn 0.042, empty state 0.000) and moves by under 0.002 with speaker-tag formatting; the magnitude is representation-dependent, the regime is not.

[3]The reviewed version reported CaSiNo at 0.220 and DealOrNoDeal at 0.180. Those two rows were produced under a context-correctness label construction with interaction features and a structure-based proxy respectively, not the stated genuine-label protocol; under deal-outcome labels with the registry estimator and dialogue-grouped folds they are -0.003 and 0.057. The other eight rows reproduce under independent re-derivation.

Table 2: State-lift across evaluation domains (MiniLM+Ridge, trajectory-grouped folds where labels are trajectory-level). Top: context-correctness or structural labels, defined by a match between action and context. Bottom: genuine labels; none clears the 0.10 threshold.[3] Admissibility per Table 1; "—": no reward model trained.

Domain	Label	n	State-lift	Admissible
Context-correctness / structural labels:				
Math reasoning (GSM8K)	step in correct chain	9,138	0.584	—
Tool-use agents (Glaive)	call in correct context	20,002	0.398	—
Code reasoning (CodeContests)	step in correct solution	9,362	0.003	—
Web navigation (Mind2Web)	action in correct trace	15,476	$-0.003^{\dagger}$	—
Genuine labels:				
Negotiation (DealOrNoDeal)	deal outcome	15,000	0.057	no
Emotional support (ESConv)	strategy progress	12,633	0.028	partial
ProsocialDialog	rule-of-thumb adherence	12,000	0.024	—
USS-Satisfaction	turn satisfaction	34,355	0.012	—
Bargaining (CraigslistBargain)	deal outcome	12,000	0.003	no
AirDialogue	booking outcome	12,000	0.001	—
Negotiation (CaSiNo)	deal outcome	10,889	-0.003	no
Persuasion (PersuasionForGood)	donation outcome	15,000	-0.001	no
Chat QA (HH-RLHF)	preference	7,094	-0.003	yes

[†]Embedding limitation: sentence embeddings cannot distinguish web actions. Bootstrap CIs are reported in Appendix K, Table 16.

5.2 Published PRM–ORM Comparisons

Published PRM–ORM gaps (math $+1.4$–17.8 points, web agents $+6$–19, code $+1.5$–2.6, general QA and chat 0; Table 14) order as the context-correctness state-lift values do, but they are not matched experiments, so we treat them as context and rest the empirical weight on the matched study below.

5.3 Matched Reward-Model Training: What State-Lift Does and Does Not Predict

To test whether state-lift (computed in minutes) predicts the outcome of full reward model training, we train Llama-3.1-8B-Instruct models with LoRA [9]: for each domain a *blind* model (action text only) and a *conditioned* model (state + action text), on the same labels, sampling, hyperparameters (3 epochs, batch size 16, learning rate 2×10^{-5}, rank-8 LoRA on query and value projections) and metric, differing only in whether the prefix is visible. Genuine-label comparisons are replicated over 8–13 seeds with dialogue-level splits where labels are trajectory-level; $\pm$ is SD across seeds. Table 3 reports the results.

Three things follow from Table 3. First, the positive direction of the screen is confirmed where it fires: the two domains above threshold show lifts of $+0.47$–0.50, though on labels that are state-dependent by construction. Second, on genuine labels the screen never fires, and on the two admissible genuine-label domains the training lifts are correspondingly small ($+0.030$, $+0.008$). Third, the reviewed version's ordering puzzle—HH-RLHF ($\mathrm{SL} \approx 0$) gaining more than DealOrNoDeal—does not replicate (over 8 paired seeds DealOrNoDeal $+0.039 \pm 0.013$ vs. HH-RLHF $+0.030 \pm 0.012$, paired $t = 2.19$, $p = .065$) and, by Table 1, was never a comparison of the same quantity: one lift is a choice between actions at a fixed state, the other a separation of better from worse trajectories. The inadmissible rows carry real lifts that are between-trajectory quantities whatever their size. Figure 4 plots state-lift against training lift.

Calibration, at its weakest. Taking multi-seed training lifts as ground truth and calling a lift material at $\delta = 0.02$, the screen at $\tau = 0.10$ fires on no genuine-label domain: zero true and zero false positives, recall zero. At $\tau = 0.05$ it fires once (DealOrNoDeal), correctly, but one trial gives a one-sided 95% false-alarm upper bound of 0.95, so this is not evidence of calibration. Since a false

[4]The reviewed version reported CaSiNo at $+0.055$ (single seed); our author response reported $+0.002 \pm 0.093$ over 14 dialogue-split runs, an aggregate that included a mis-configured smoke-test run (seed 99, a handful of examples, lift -0.31). The 13 valid seeds give $+0.027 \pm 0.022$. Under either value CaSiNo is inadmissible (Table 1) and not a supporting domain.

Table 3: Matched blind (ORM) vs. state-conditioned (PRM) training. Top: context-correctness labels (single seed, AUC). Bottom: genuine labels (accuracy, mean ± SD over seeds, dialogue-level splits), split by admissibility (Table 1); on inadmissible domains the lift is a between-trajectory separation.

Domain	Label	SL	Seeds	Blind	PRM	Lift
Context-correctness labels:						
Math reasoning	ctx-correct	0.584	1	0.526	1.000	$+0.474$
Tool-use agents	ctx-correct	0.398	1	0.486	0.989	$+0.503$
Code reasoning	ctx-correct	0.003	1	0.556	0.576	$+0.020$
Genuine labels (admissible):						
Chat QA (HH-RLHF)	preference	-0.003	8	0.624	0.654	$+0.030 \pm 0.012$
Emot. support (ESConv)	progress	0.028	8	0.570	0.579	$+0.008 \pm 0.009$
Genuine labels (inadmissible; between-trajectory):						
Negotiation (DealOrNoDeal)	deal outcome	0.057	8	0.658	0.697	$+0.039 \pm 0.013$
Negotiation (CaSiNo)	deal outcome	-0.003	13	0.513	0.540	$+0.027 \pm 0.022$[4]
PersuasionForGood	donation	-0.001	3	0.554	0.642	$+0.088 \pm 0.025$
CraigslistBargain	deal outcome	0.003	3	0.557	0.646	$+0.089 \pm 0.032$

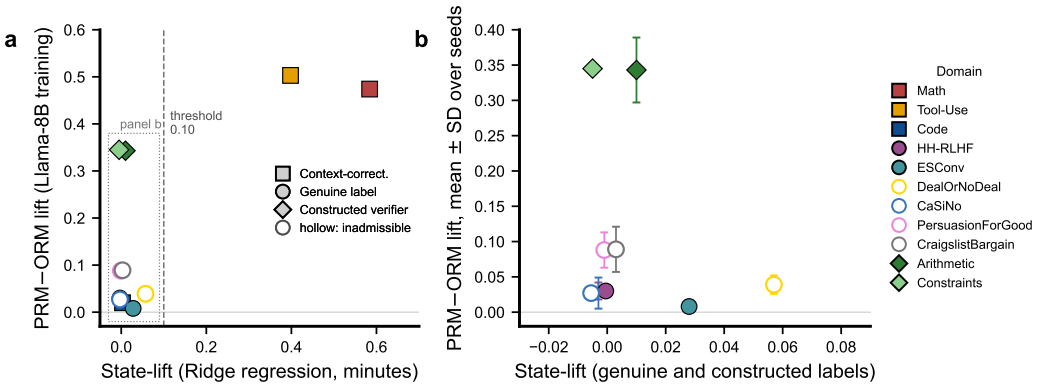

Figure 4: State-lift (x-axis, minutes) vs. PRM−ORM lift from matched Llama-8B training (y-axis, hours). Squares = context-correctness labels, circles = genuine labels, diamonds = constructed verifier domains; colours identify domains; hollow markers are inadmissible (labels never vary within a state, Table 1). (a) Full range. (b) Zoom on the dotted region with SDs over seeds. Large lifts above threshold occur only with context-correctness labels; no genuine-label domain clears the threshold; and on the admissible constructed domains the probe reads ≈ 0 against trained lifts of $+0.34$.

"invest" costs weeks of training and a miss one cheap re-check, the operating point should favour precision. Single-run probe values carry fold SEs up to 0.07; Step 1 therefore uses at least three seeds.

Fixed-state selection. A reward model is deployed to choose among candidates at a fixed state, so we also evaluate best-of-N top-1 selection with all N candidates drawn from the same held-out state—judge-free and unwinnable by a scorer that merely rates contexts. On ESConv (3 seeds, 600 trials each, 32 held-out states) the conditioned scorer gives no advantage at any $N \in \{2, 4, 8, 16\}$ (-0.019 ± 0.033 to -0.004 ± 0.017), matching its pairwise lift; on a verifiable-constraint positive control the same protocol returns $+0.336 \pm 0.023$ to $+0.705 \pm 0.038$, so the null is a property of ESConv. On DealOrNoDeal it cannot be run (no held-out state carries competing candidates).

Hyperparameter check. Identical hyperparameters across arms could disadvantage the blind arm on low-SL domains. Re-training both arms on ESConv, HH-RLHF and CaSiNo at $2.5\times$ the learning rate, at 5 epochs, and at both (two seeds each, 18 runs; Appendix J.4) moves no domain's mean lift by more than 0.005; all 18 lifts lie within one baseline SD.

Boundary: symbolic step grading. To probe the diagnostic's boundary we apply it to four math step-grading datasets with three label sources (PRM800K [17] human ratings, MathShepherd [27] MCTS labels in two formats, RLHFlow-PRM automated verification). All yield near-zero state-lift (-0.002 to 0.068; Appendix G, Table 8), far below the empirical PRM gap. Eight encoding architectures on MathShepherd, including cross-encoders with full cross-attention, all yield $\mathrm{SL} \in [-0.27, 0.038]$, while end-to-end training on PRM800K achieves $+0.238$ AUC lift (Table 9): the signal is outside the reach of any probe-style measurement, not just encode-separately ones. We do not claim the diagnostic covers fine-grained math step grading. The same limit appears on two constructed verifiable domains (arithmetic step validity and constraint satisfaction; every state contrastive, ICC 0.000): the probe reads $+0.010$ and -0.005 while the trained conditioned scorer gains $+0.343$ and $+0.345$ AUC, unchanged from bi-encoder to cross-encoder and up a four-tier capacity ladder. Judging these steps requires *computation* over the joint (state, action), which pooled embeddings with a linear head cannot perform; a near-zero reading is therefore not self-interpreting. When the screen reads near zero, the protocol trains both scorers for a small fraction of the budget and reads the gap against a control that permutes which state is paired with which action (Appendix J.3). On the two constructed domains a tenth of the budget already reads $+0.10$ and $+0.13$ (permuted ≈ 0); on HH-RLHF, the one public admissible human-labelled domain, all four seeds carry the wrong sign at a tenth and a read positive in every seed needs 46% of the budget. The escalation is cheap when the eventual lift is large and expensive when it is not, so it needs at least two seeds and a margin over their spread. This finding does not contradict state-lift's success under context-correctness labels (GSM8K: $\mathrm{SL} = 0.584$, $+0.474$ AUC; downstream beam search in Appendix F). The boundary is a property of what the label requires the scorer to compute, not of the domain.

Downstream and artifact checks. A step-level beam search on GSM8K confirms that state-lift tracks downstream performance, not just AUC: PRM-guided selection reaches 26% accuracy versus 15% ORM-guided and 13% random (Appendix F). Shuffled-state controls destroy 79–100% of measured state-lift across five domains (Appendix E).

6 Discussion

What state-lift measures. State-lift captures whether action text alone suffices to predict step quality. When it does ($\mathrm{SL} \approx 0$), an ORM captures the quality-relevant signal. When it does not ($\mathrm{SL} \gg 0$), the ORM is blind to quality information that resides in the trajectory state. Crucially, state-lift measures state-dependent *quality*, not state *persistence*: a domain can have highly persistent states (multi-turn dialogues with strong autocorrelation) yet state-independent quality (helpful responses are helpful regardless of context). Our proxy pitfall analysis (Section 4.5) demonstrates this distinction concretely. And it measures *action-ranking* state-dependence only when labels vary within a state (Section 4.3); on trajectory-labelled data it measures between-trajectory differences. A per-quartile analysis of HH-RLHF confirms $\mathrm{SL} \approx 0$ is not an aggregation artifact (Appendix K).

The effective gap explains within-category variation. Why does code reasoning ($\mathrm{SL} = 0.003$) have a smaller published PRM advantage (1.5–2.6%) than math ($\mathrm{SL} = 0.584$, 1.4–17.8%)? Code has $\textsc{Eff-Gap} = 11.9$ (action gaps dominate) while math has $\textsc{Eff-Gap} = 1.11$ (state noise dominates); the predicted upper-bound regret at $K = 8$ is 0.056 versus 0.779, a $14\times$ ratio consistent with the observed gaps (Appendix H).

Practitioner decision protocol. We recommend the following workflow for a new domain.

Step 0. Admissibility audit (one pass, no model). Compute the fraction of states carrying more than one label and the label ICC (Section 4.3). If no state is contrastive, stop: the dataset cannot express the within-state comparison, and no state-lift value on it bears on the decision. If labels are trajectory-level, use trajectory-grouped folds throughout.

Step 1. Compute state-lift ($\sim$5 minutes per seed, CPU only; at least three seeds, report the SE). Collect $\sim$500 (state, action, quality) tuples following Section 4.4; embed, PCA to 16 dimensions, fit Ridge; compute SL and the effective gap.

Step 2. Read the result one-directionally:

- $\mathrm{SL} > 0.10$ **with the lower confidence bound above zero → Invest in PRM.** The state block carries predictive content the action block does not; in our study this regime is populated by context-correctness labels (math $+0.474$, tool-use $+0.503$).
- $\mathrm{SL} \approx 0 \rightarrow$ **Escalate, do not conclude.** Train both scorers for $\sim$10% of the budget with two seeds and the state-permutation control; accept the read when the gap exceeds the seed spread and the control is at zero. A near-zero screen is consistent both with ORM sufficiency (ESConv) and with computation-dependent state-dependence the probe cannot see (arithmetic, lift $+0.34$); only training distinguishes them.

Tasks whose steps require verifying a formal computation can go directly to Step 2 or default to PRM; the eight-architecture evidence below shows the probe reads near zero on them regardless.

Two structural classes of PRM advantage. Our results suggest that the PRM advantage is not one phenomenon but two. In *representation-decomposable* domains (negotiation, tool-use, agent tasks), step quality depends on general features of the state that sentence embeddings capture—which region of action-space is appropriate given the conversational or task context. State-lift detects this reliably. In *interaction-irreducible* domains (symbolic step grading), step quality depends on fine-grained joint computation between specific tokens in the state and action (e.g., checking that a particular algebraic manipulation is valid given the current equation). Eight encoding architectures fail to recover this signal while end-to-end training succeeds (Appendix G): the two classes are structurally different phenomena. Representation-decomposable domains can be triaged cheaply (Step 1), interaction-irreducible domains require training (Step 2); we believe this decomposition may inform how the field thinks about process supervision beyond the specific diagnostic presented here.

State-lift is a representation-dependent measurement with robust ordering. State-lift is a joint property of the *domain*, the *representation*, and the *label type*. Absolute values vary across encoders (MiniLM: 0.45 for Math vs. MPNet: 0.53; Appendix K), but the ordering GSM8K (context-correctness) $\gg$ ESConv $\approx$ HH-RLHF holds across all three sentence encoders and across linear, kernel, and MLP estimators. State-lift is therefore a reliable *ranking* by recoverable state-action signal; practitioners should compare values within an encoder rather than thresholds across encoders. The exception is computation-dependent state-dependence, which no probe architecture recovers (Appendix G).

7 Limitations

State-lift does not detect computation-dependent state-dependence: eight encoding architectures on four math datasets all yield $\mathrm{SL} \in [-0.27, 0.07]$ while LLM training achieves $+0.238$ AUC lift (Appendix G), and on two admissible constructed domains the probe reads ≈ 0 against trained lifts of $+0.34$; a near-zero reading must be escalated. The escalation costs 46% of full training on the one public admissible human-labelled domain we have. After correction no genuine-label domain in our study clears the threshold, so the positive direction is demonstrated only on context-correctness labels. DealOrNoDeal is the only natural domain that both fails admissibility and carries a non-trivial state-lift. Under a protocol that holds out entire constraint families, HH-RLHF reads $+0.238$ on one run, which we have not explained. Escalation results rest on 2–4 seeds per domain. Context-correctness labels inflate SL by construction (Section 5). Sentence embeddings may fail for structurally-differentiated actions (e.g., web UI elements). The Gaussian regret bound is approximate; domain ordering is preserved under heavier tails (Appendix K). Related work is discussed in Appendix A.

8 Conclusion

We introduced state-lift, a diagnostic that screens when process rewards outperform outcome rewards. Across ten domains and matched multi-seed reward-model training, it is a one-directional screen with a stated precondition (labels must vary within a state, checked by a one-pass audit) and an escalation for the near-zero case. Regret depends on how much quality varies with state within a fixed context (state-lift) and how distinguishable actions are (effective gap); the diagnostic is most useful for novel domains without a published comparison, and runs in minutes on a single CPU.

References

[1] Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return Decomposition for Delayed Rewards, September 2019. URL http://arxiv.org/abs/1806.07857. arXiv:1806.07857 [cs].

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback, April 2022. URL http://arxiv.org/abs/2204.05862. arXiv:2204.05862 [cs].

[3] Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. CaSiNo: A Corpus of Campsite Negotiation Dialogues for Automatic Negotiation Systems. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3167–3185, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.254. URL https://aclanthology.org/2021.naacl-main.254/.

[4] Sanjiban Choudhury. Process Reward Models for LLM Agents: Practical Framework and Directions, February 2025. URL http://arxiv.org/abs/2502.10325. arXiv:2502.10325 [cs].

[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, November 2021. URL http://arxiv.org/abs/2110.14168. arXiv:2110.14168 [cs].

[6] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a Generalist Agent for the Web, December 2023. URL http://arxiv.org/abs/2306.06070. arXiv:2306.06070 [cs].

[7] Huifang Du, Shuqin Li, Minghao Wu, Xuejing Feng, Yuan-Fang Li, and Haofen Wang. Rewarding What Matters: Step-by-Step Reinforcement Learning for Task-Oriented Dialogue. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8030–8046, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.472. URL https://aclanthology.org/2024.findings-emnlp.472/.

[8] Miroslav Dudik, John Langford, and Lihong Li. Doubly Robust Policy Evaluation and Learning, May 2011. URL http://arxiv.org/abs/1103.4601. arXiv:1103.4601 [cs].

[9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. URL http://arxiv.org/abs/2106.09685. arXiv:2106.09685 [cs].

[10] Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process Reward Models That Think, December 2025. URL http://arxiv.org/abs/2504.16828. arXiv:2504.16828 [cs].

[11] Hyunwoo Kim, Youngjae Yu, Liwei Jiang, Ximing Lu, Daniel Khashabi, Gunhee Kim, Yejin Choi, and Maarten Sap. ProsocialDialog: A Prosocial Backbone for Conversational Agents. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4005–4029, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.267. URL https://aclanthology.org/2022.emnlp-main.267/.

[12] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating Reward Models for Language Modeling. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational*

Linguistics: NAACL 2025, pages 1755–1797, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.96. URL https://aclanthology.org/2025.findings-naacl.96/.

[13] Dong Bok Lee, Seanie Lee, Sangwoo Park, Minki Kang, Jinheon Baek, Dongki Kim, Dominik Wagner, Jiongdao Jin, Heejun Lee, Tobias Bocklet, Jinyu Wang, Jingjing Fu, Sung Ju Hwang, Jiang Bian, and Lei Song. Rethinking Reward Models for Multi-Domain Test-Time Scaling, October 2025. URL http://arxiv.org/abs/2510.00492. arXiv:2510.00492 [cs].

[14] Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. Deal or No Deal? End-to-End Learning of Negotiation Dialogues. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1259. URL https://aclanthology.org/D17-1259/.

[15] Qingyao Li, Xinyi Dai, Xiangyang Li, Weinan Zhang, Yasheng Wang, Ruiming Tang, and Yong Yu. CodePRM: Execution Feedback-enhanced Process Reward Model for Code Generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8169–8182, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.428. URL https://aclanthology.org/2025.findings-acl.428/.

[16] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson dAutume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, December 2022. doi: 10.1126/science.abq1158. URL https://www.science.org/doi/10.1126/science.abq1158.

[17] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. October 2023. URL https://openreview.net/forum?id=v8L0pN6EOi.

[18] Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. Towards Emotional Support Dialog Systems. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3469–3483, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.269. URL https://aclanthology.org/2021.acl-long.269/.

[19] Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An Analytical Evaluation Board of Multi-turn LLM Agents. November 2024. URL https://openreview.net/forum?id=4S8agvKjle#discussion.

[20] Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 278–287, San Francisco, CA, USA, June 1999. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-612-8.

[21] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 27730–27744, Red Hook, NY, USA, November 2022. Curran Associates Inc. ISBN 978-1-7138-7108-8.

[22] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models. October 2021. URL https://openreview.net/forum?id=JYtwGwIL7ye.

[23] Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding Progress: Scaling Automated Process Verifiers for LLM Reasoning. October 2024. URL https://openreview.net/forum?id=A6Y7AqlzLW.

[24] Joar Max Viktor Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and Characterizing Reward Gaming. October 2022. URL https://openreview.net/forum?id=yb3HOXO3lX2.

[25] Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. October 2024. URL https://openreview.net/forum?id=4FWAwZtd2n.

[26] Weiwei Sun, Shuo Zhang, Krisztian Balog, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten de Rijke. Simulating User Satisfaction for the Evaluation of Task-oriented Dialogue Systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 2499–2506, New York, NY, USA, July 2021. Association for Computing Machinery. ISBN 978-1-4503-8037-9. doi: 10.1145/3404835.3463241. URL https://dl.acm.org/doi/10.1145/3404835.3463241.

[27] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

[28] Wei Wei, Quoc Le, Andrew Dai, and Jia Li. AirDialogue: An Environment for Goal-Oriented Dialogue Research. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1419. URL https://aclanthology.org/D18-1419/.

[29] Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, Ying Fan, Jungtaek Kim, Hyung Il Koo, Kannan Ramchandran, Dimitris Papailiopoulos, and Kangwook Lee. VersaPRM: Multi-Domain Process Reward Model via Synthetic Reasoning Data. June 2025. URL https://openreview.net/forum?id=l19DmXbwPK.

[30] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The Lessons of Developing Process Reward Models in Mathematical Reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.547. URL https://aclanthology.org/2025.findings-acl.547/.

[31] Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. SWEET-RL: Training Multi-Turn LLM Agents on Collaborative Reasoning Tasks, March 2025. URL http://arxiv.org/abs/2503.15478. arXiv:2503.15478 [cs].

[32] Simon Zhuang and Dylan Hadfield-Menell. Consequences of misaligned AI. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 15763–15773, Red Hook, NY, USA, December 2020. Curran Associates Inc. ISBN 978-1-7138-2954-6. URL https://dl.acm.org/doi/10.5555/3495724.3497046.

A Related Work

Process reward models. PRMs outperform ORMs for math [17, 27, 23], agents [4], and dialogue [7], with modest gains for code [15]. ThinkPRM [10] and VersaPRM [29] generalize across domains. Song et al. [30] documented practical lessons on when PRMs help versus hurt. Critically, Lee et al. [13] found PRMs offer no advantage on MMLU-Pro, establishing that the PRM advantage is domain-dependent. Our diagnostic explains why.

Reward decomposition and diagnostics. State-lift connects to potential-based reward shaping [20] and return decomposition for credit assignment [1]. It identifies state-blindness as a specific form of reward misspecification [32, 24, 22]. The variance decomposition relates to contextual bandits [8]. To our knowledge, no prior work provides a diagnostic predicting PRM vs. ORM suitability; RewardBench [12] evaluates trained models rather than predicting which type to train.

Test-time compute. PRMs enable efficient test-time compute scaling [25]. Agent frameworks such as SWEET-RL [31] and AgentBoard [19] rely on turn-level signals; state-lift could predict when such supervision is worthwhile.

B Dataset Descriptions

Table 4: Quality signals used across the ten evaluation domains, by label type. "Automated" = derivable from environment without human input; "Existing" = pre-existing human-annotated labels in the dataset; "Proxy" = a heuristic stand-in (e.g., movement toward population centroid).

Domain	Quality Signal	Source	Cost
Math reasoning (GSM8K)	Step in correct chain	GSM8K solutions	Automated
Tool-use agents (Glaive)	Action in correct context	Conversation logs	Automated
Negotiation (CaSiNo)	Deal-outcome class	Final point score	Automated
Emotional support (ESConv)	Therapeutic progress	Conversation endpoints	Proxy
ProsocialDialog	Rule-of-thumb adherence	Human annotation	Existing
USS-Satisfaction	Turn-level satisfaction	User satisfaction rating	Existing
Code reasoning (CodeContests)	From correct solution	Execution results	Automated
AirDialogue	Booking outcome	Deal-success label	Existing
Chat QA (HH-RLHF)	Chosen vs. rejected	Human preference	Existing
Web navigation (Mind2Web)	Action in correct trace	Trajectory logs	Automated

Table 5 summarizes the evaluation domains, including task format, data source, size, and the nature of the sequential structure.

Table 5: Datasets used for state-lift evaluation. All datasets are publicly available. n denotes the number of step-level transitions extracted for state-lift computation.

Domain	**Source**	n	**Description**
Math reasoning	GSM8K [5]	9,138	Grade-school math word problems solved in multi-step chains. Each step is an arithmetic or algebraic operation. Quality depends on whether the step is logically valid given prior steps.
Tool-use agents	Glaive Function Calling v2[5]	20,002	Multi-turn dialogues where an assistant invokes APIs/tools. Each turn involves selecting and parameterizing a function call. Quality depends on whether the call is appropriate for the user's request and prior tool outputs.
Negotiation	CaSiNo [3]	24,474	Two-party camping supply negotiations. Participants trade firewood, water, and food packages to maximize individual point scores. Quality is labeled by final deal outcome (points above/below median).
Negotiation	DealOrNoDeal [14]	28,200	Two-party item-division negotiations over books, hats, and balls with private valuations. Quality is labeled by deal value (agent value / max value, above/below median).
Emotional support	ESConv [18]	12,633	Emotional support conversations between a help-seeker and supporter. Multi-turn dialogues following a structured support strategy. Quality proxied by therapeutic progress (movement toward population centroid).
Code reasoning	CodeContests [16]	9,362	Competitive programming problems with multi-step solutions. Each step is a code block or reasoning step. Quality derived from whether the step belongs to a correct solution.
Chat QA	Anthropic HH-RLHF [2]	7,094	Multi-turn human–assistant conversations with human preference labels (chosen vs. rejected). Covers helpful and harmless response selection.
Web navigation	Mind2Web [6]	15,476	Web browsing tasks where an agent selects UI elements and actions to complete goals. Each step is a click/type/select action on a webpage.
Prosocial response	ProsocialDialog [11]	15,000	Multi-turn dialogues annotated for whether responses follow prosocial rules-of-thumb. Safety labels on a 5-level scale from casual to needs-intervention.
User satisfaction	USS-Satisfaction [26]	15,000	Task-oriented dialogues with turn-level user satisfaction ratings (1–5 scale) from multiple annotators across four source datasets.
Booking dialogue	AirDialogue [28]	15,000	Customer-agent flight-booking dialogues with deal-outcome labels (booking succeeded vs. failed).

C Full Proofs

Reduction to the interaction component. With $\delta(s,a) = m(s) + g(s,a)$ as in the Remark of Section 3, $\max_a R(s,a) = m(s) + \max_a\{\mu_a + g(s,a)\}$ and $R(s,\pi_{\text{blind}}) = m(s) + \mu_{(1)} + g(s,\pi_{\text{blind}})$; taking $\mathbb{E}_s$, m cancels and $\mathbb{E}_s[g(s,\pi_{\text{blind}})] = 0$, so the regret equals $\mathbb{E}_s[\max_a\{\mu_a + g(s,a)\}] - \mu_{(1)}$. The proofs below therefore apply to $R(s,a) = \mu_a + \sigma Z_a(s)$ with $\sigma^2 = \text{SL}_{\text{within}} \cdot \text{Var}(R)$; when $m \equiv 0$ this is the statement of the reviewed version with $\sigma^2 = \text{SL} \cdot \text{Var}(R)$.

C.1 Proof of Theorem 1(i)

If $\text{SL} = 0$, then $\text{Var}_s(R \mid a) = 0$ for all a, which implies $R(s,a) = \mathbb{E}_s[R(s,a)] = r_{\text{blind}}(a)$ almost surely for all a. Therefore $\arg\max_a R(s,a) = \arg\max_a r_{\text{blind}}(a) = \pi_{\text{blind}}$ for all s. □

C.2 Proof of Theorem 1(ii)

The expected regret is:

$$\mathbb{E}[\text{regret}] = \mathbb{E}_s\Big[\max_a R(s,a)\Big] - \max_a \mathbb{E}_s[R(s,a)] \tag{5}$$

$$= \mathbb{E}_s\Big[\max_a(\mu_a + \sigma Z_a)\Big] - \mu_{(1)}. \tag{6}$$

Since $\max$ is convex and the Z_a are i.i.d. standard normal, the worst case (maximizing regret over $\{\mu_a\}$) occurs when all μ_a are equal, giving:

$$\mathbb{E}[\text{regret}] \le \sigma \cdot \mathbb{E}\Big[\max_{a=1}^{K} Z_a\Big] \le \sigma \cdot \sqrt{2\ln K}, \tag{7}$$

where the last inequality is the standard bound on the expected maximum of K standard normals. Substituting $\sigma = \sqrt{\text{SL} \cdot \text{Var}(R)}$ gives the result. □

C.3 Proof of Theorem 1(iii)

For $K = 2$ with $\mu_1 > \mu_2$, the blind policy selects action 1. The regret is:

$$\mathbb{E}[\text{regret}] = \mathbb{E}\big[(\mu_2 + \sigma Z_2 - \mu_1 - \sigma Z_1)^+\big]. \tag{8}$$

Let $W = Z_2 - Z_1 \sim N(0,2)$ and $\Delta = \mu_1 - \mu_2$. Then:

$$\mathbb{E}[\text{regret}] = \mathbb{E}\big[(\sigma W - \Delta)^+\big] \tag{9}$$

$$= \sigma\sqrt{2}\,\phi\left(\frac{\Delta}{\sigma\sqrt{2}}\right) - \Delta\,\Phi\left(\frac{-\Delta}{\sigma\sqrt{2}}\right), \tag{10}$$

using the known formula for the expected positive part of a shifted normal: for $X \sim N(m, s^2)$, $\mathbb{E}[X^+] = m\,\Phi(m/s) + s\,\phi(m/s)$; here $m = -\Delta$ and $s = \sigma\sqrt{2}$, and ϕ is even. (The prose statement of part (iii) in the reviewed version carried a plus sign on the second term; the expression above, used by the released code and every figure, is the correct one.) We verified this expression against Monte Carlo simulation (2×10^6 draws) at $(\Delta,\sigma) \in \{(1,1),(0.5,1),(2,1),(1,0.3),(0.2,2)\}$; agreement holds to four decimal places. For general K, the actual regret is at least as large as this two-action lower bound, since additional actions can only increase $\max_a R(s,a)$. □

C.4 Monotonicity of Regret in State-Lift

Proposition 1. *For fixed action means $\{\mu_a\}$, the expected regret is non-decreasing in σ (and hence in state-lift).*

Proof. Let $g(\mathbf{x}) = \max_a(\mu_a + x_a) - \mu_{(1)}$. Then g is convex, $g(\mathbf{0}) = 0$, and $g(\mathbf{x}) \ge 0$. Define $h(\sigma) = \mathbb{E}[g(\sigma\mathbf{Z})]$. By Jensen's inequality, $h(\sigma) \ge g(\mathbf{0}) = 0$ for all σ. Since g is convex, h is convex in σ. A convex function h with $h(0) = 0$ and $h(\sigma) \ge 0$ is non-decreasing for $\sigma \ge 0$: for $0 < \sigma_1 < \sigma_2$, convexity gives $h(\sigma_1) \le (\sigma_1/\sigma_2)h(\sigma_2)$, which implies $h(\sigma_2) \ge (\sigma_2/\sigma_1)h(\sigma_1) \ge h(\sigma_1)$. □

D Proxy Pitfall: Extended Analysis

The improvement proxy defines quality as $q_{\text{proxy}}(s, a) = \|s - \bar{s}\| - \|s' - \bar{s}\|$, where $\bar{s}$ is the population mean state and s' is the next state. This proxy measures "movement toward the centroid" and is high when states are far from the mean and actions bring them closer.

On HH-RLHF, this proxy yields $\text{SL}_{\text{proxy}} = 17.2$ because: (1) conversational states have high autocorrelation ($r = 0.27$ between consecutive states), so states far from the mean tend to stay far; (2) the centroid-movement proxy captures this persistence as "state-dependent quality"; (3) actual preference labels show $\text{SL}_{\text{real}} = -0.005$, confirming that preferences do not depend on conversational state.

The root cause is a conflation of two distinct properties: *state persistence* (how long states stay in a region) and *state-dependent quality* (whether the quality ranking of actions changes with state). A domain can have high persistence but low quality state-dependence (e.g., multi-turn chat where helpful responses are helpful regardless of context) or high quality state-dependence but low persistence (e.g., a single-step decision that depends on a rapidly changing state).

We tested four alternative proxy formulations—world-model residual, turn position, action diversity, and next-state predictability—and found that all exhibit similar failure modes to varying degrees. The only reliable approach is to use quality labels derived from the environment.

E Artifact Checks

Shuffled-state controls confirm that state-lift captures genuine state-dependence rather than statistical artifacts. We randomize state assignments while keeping actions and labels fixed; if state-lift reflects real structure, shuffling should destroy it.

Table 6: Shuffled-state controls confirm state-lift reflects genuine sequential structure.

Domain	Real	Shuffled	Drop
Math (GSM8K)	0.584	−0.013	100%
Tutoring (MathDial)	0.114	0.003	97%
Code (CodeContests)	0.010	0.002	79%
Emot. support (ESConv)	0.028	0.006	79%
Web nav. (Mind2Web)	Permutation $z = 11.3$		$p < 0.0001$

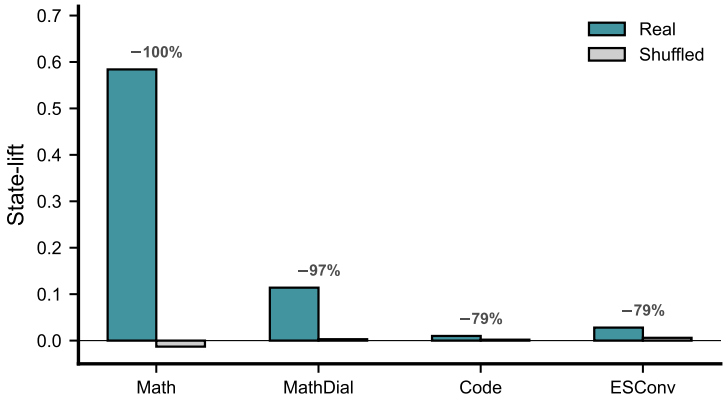

Figure 5: Shuffled-state controls. Randomizing state assignments while keeping actions and labels fixed destroys state-lift in all tested domains (79–100% drop), confirming the signal is genuine.

F Downstream Step-Level Beam Search

To test whether state-lift predicts *downstream task performance* (not just classification AUC), we conduct a step-level beam search on GSM8K. We generate 6,400 solutions from Llama-3.1-8B-

Instruct on 400 training problems, auto-label steps by outcome correctness, and train matched-distribution reward models (blind AUC $= 0.648$, PRM AUC $= 0.801$, lift $= +0.153$). We then solve 100 test problems step-by-step: at each step, generate 4 candidates and select the best according to each method.

Table 7: Downstream step-level beam search on GSM8K (matched-distribution reward models).

Method	Accuracy	Selection strategy
Greedy (no selection)	27.0%	Argmax decoding, single sample
Random (from 4 cands)	13.0%	Uniform random from 4 samples
ORM-guided (blind)	15.0%	Best-of-4 scored by step text only
PRM-guided (conditioned)	**26.0%**	Best-of-4 scored by state + step

PRM-guided selection (26%) dramatically outperforms ORM-guided selection (15%), a $+11$ percentage point gap. The ORM is barely better than random (15% vs. 13%), confirming that step quality in math reasoning is largely invisible from step text alone. PRM nearly matches greedy decoding (26% vs. 27%) despite stochastic sampling.

G Resolution Experiment

This experiment characterizes the diagnostic's behavior on *fine-grained math step grading*—tasks whose labels grade the symbolic correctness of individual reasoning steps. Practitioners triaging a new domain typically do not have such labels; they have the label types covered in the main experiments (context-correctness, outcome-based, preference, satisfaction, proxy-quality). We include this experiment to characterize what happens when state-lift is applied to math step grading, and to test whether the failure is an architectural limitation of bi-encoder probes or a more fundamental property of the task.

Four independent boundary datasets. We test state-lift on four math step-grading datasets with three independent label sources: PRM800K [17] provides 78,865 steps with *human* $+1/-1/0$ ratings; MathShepherd [27] provides steps labeled by *Monte-Carlo Tree Search* rollouts (tested in two different data formats); and RLHFlow-PRM provides steps with automated $+/-$ verification labels. All yield near-zero state-lift:

Table 8: State-lift on four math step-grading boundary datasets, with MiniLM bi-encoder + linear Ridge probe.

Dataset	Label source	State-lift	95% CI	LLM training lift
PRM800K	Human $+1/0/-1$	0.015	$[-0.010, 0.017]$	$+0.238$ AUC
MathShepherd	MCTS rollout	-0.002	$[-0.004, 0.034]$	$\sim$0.35 AUC (published)
trl-MathShepherd	MCTS (TRL format)	0.068	$[0.035, 0.081]$	—
RLHFlow-PRM	Automated $+/-$	0.010	$[-0.011, 0.015]$	—

The result is robust across label sources and data formats. The trl-MathShepherd variant shows slightly higher SL (0.068), likely reflecting cleaner step segmentation in TRL's pre-tokenized format, but remains 5–7$\times$ below the empirical PRM gap ($\sim$0.35). Three of the four datasets have CIs that include zero; all four are well below the 0.10 PRM-investment threshold.

Eight encoding architectures. To rule out the hypothesis that the failure is specific to bi-encoders that encode states and actions independently, we apply eight different encoding architectures to MathShepherd and compute state-lift with each. Architectures span similarity-enriched bi-encoders, cross-encoders with full cross-attention, NLI entailment models, and joint MPNet/DeBERTa encoders.

The persistent gap across architectures—including those with full cross-attention—indicates that the symbolic-correctness signal recovered by end-to-end LLM training (PRM800K $+0.238$ AUC) is not a structure that any probe-style measurement can recover, regardless of how the probe combines state and action representations. The earlier hypothesis that the boundary was caused by encode-separately architecture is ruled out by rows 3–4. Figure 6 visualizes this gap.

Table 9: Eight encoding architectures applied to MathShepherd. All yield SL in $[-0.27, 0.07]$. Cross-encoders with full cross-attention (rows 3–4) do not close the gap, ruling out the hypothesis that the boundary is an artifact of encode-separately probes.

Method	SL (R^2)	SL **(AUC)**	**Notes**
Bi-encoder (MiniLM)	-0.002	-0.003	Baseline
Similarity-enriched	0.006	-0.010	$\cos(s, a) \leftrightarrow$ correct: ρ=0.31
Cross-encoder (ms-marco)	0.000	-0.004	Full cross-attention
Cross-encoder + embeddings	0.003	-0.042	—
NLI (entailment)	-0.000	-0.012	Entailment $\neq$ correctness
NLI-only (no action emb)	0.038	0.124	Best, still $10\times$ below gap
Joint MPNet	-0.267	-0.172	Joint encoding destroys signal
Joint DeBERTa	-0.174	-0.154	Same problem

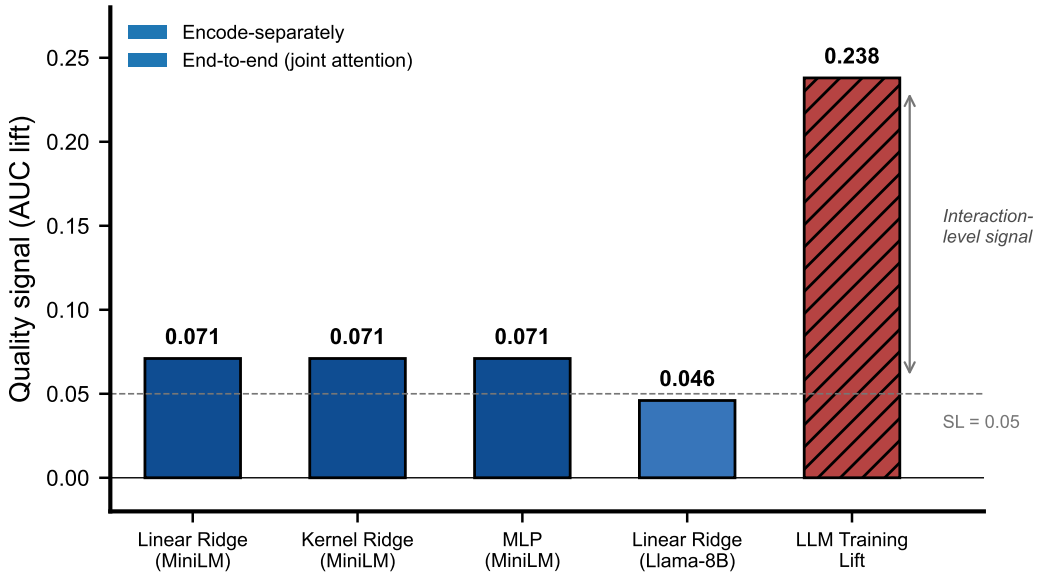

Figure 6: The resolution experiment: probe-style measurements of state-lift (any encoder, any estimator) yield near-zero values on PRM800K, while end-to-end LLM training achieves $+0.238$ AUC lift. The gap is robust across architectures and across four independent datasets with three label sources (PRM800K human ratings; MathShepherd MCTS labels in two formats; RLHFlow automated verification).

Consistency with state-lift's success on math under different labels. This boundary does not contradict the diagnostic's success on the same domain family under *different* label types: GSM8K with context-correctness labels yields SL $= 0.584$ and a $+0.474$ AUC lift (Tables 2, 3; downstream beam search in Appendix F). The boundary is a property of what the label requires the scorer to compute, not of the domain.

H Effective Gap Regimes

Figure 7 visualizes the two regimes predicted by Theorem 1 as a function of the effective gap $\textsc{Eff-Gap} = \sqrt{(1 - \text{SL})/\text{SL}}$. When $\textsc{Eff-Gap} < 1$ (state noise dominates the action-mean gap), PRM is essential; when $\textsc{Eff-Gap} > 3$ (action gaps dominate), ORM is sufficient. The intermediate range is a hybrid regime where PRM provides moderate benefit. Under context-correctness labels, math and tool-use fall in the noise-dominant regime and code in the gap-dominant regime.

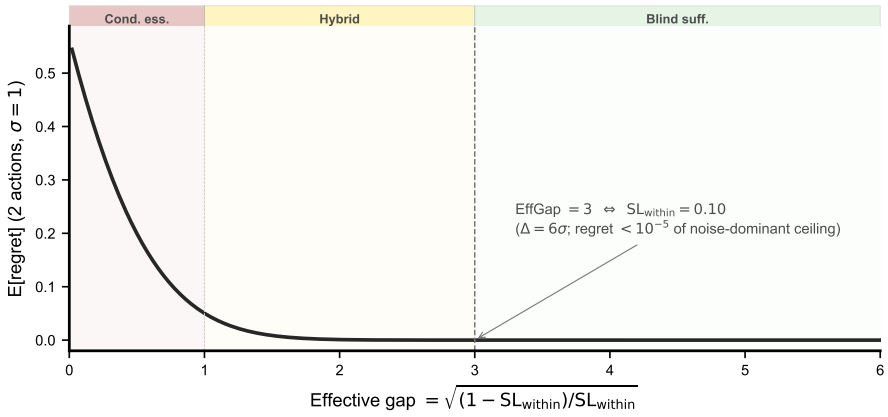

Figure 7: Two regimes determined by the effective gap. When EFF-GAP < 1 (state noise dominates action gaps), PRM is essential. When EFF-GAP > 3 (action gaps dominate), ORM is sufficient. The dashed line marks EFF-GAP $= 3$, i.e. SL $= 0.10$.

I Interpretation of K in Text Generation

In the regret bound, $K = |\mathcal{A}|$ is the number of candidate actions at *selection* time (e.g., N in best-of-N, beam width, or tree search branching). The $\sqrt{2 \ln K}$ factor grows slowly: $K = 8$ gives 2.04, $K = 64$ gives 2.88, a $1.4\times$ range that does not change the PRM/ORM recommendation for any domain. The regime boundary (SL $= 0.10$, EFF-GAP $= 3$) is invariant across this practical range.

J Extended Experimental Results

J.1 LLM Training Details

All LLM experiments use Llama-3.1-8B-Instruct with LoRA (rank 8, alpha 16, targeting query and value projections). Training hyperparameters are held constant across domains: 3 epochs, batch size 4 with gradient accumulation 4 (effective batch size 16), learning rate 2×10^{-5} with AdamW, maximum sequence length 512 tokens. Models are trained on NVIDIA L40S GPUs (48 GB VRAM).

J.2 Per-Domain Training Curves

Table 10 shows per-epoch training and evaluation metrics for all five LLM experiments.

Table 10: Per-epoch metrics for blind (ORM) and conditioned (PRM) Llama-8B reward models.

Domain	Model	Ep.	Train Acc	Eval AUC	Loss	Time (s)
Math	Blind	1	0.497	0.526	0.891	108
		2	0.506	0.502	0.738	107
		3	0.527	0.500	0.704	108
	PRM	1	0.893	1.000	0.275	261
		2	0.999	1.000	0.004	256
		3	1.000	1.000	<0.001	256
Tool-Use	Blind	1	—	0.486	—	—
		3	—	0.486	—	—
	PRM	1	—	0.989	—	—
		3	0.975	0.989	0.117	269
HH-RLHF	Blind	best	0.637	—	0.651	275
	PRM	best	0.709	—	0.564	759

The most striking pattern is the contrast between blind and conditioned models on math: the blind model never exceeds chance (AUC=0.526) across all three epochs, while the conditioned model reaches near-perfect accuracy (AUC=1.000) within the first epoch and converges to loss <0.001 by epoch 3. This demonstrates that the quality signal is completely invisible from action text alone but trivially recoverable with state information.

J.3 Truncated-Training Escalation

Both arms are trained with evaluation every 10 optimizer steps; Δ_B is conditioned minus blind accuracy at step B, mean over seeds, and B^* the first step after which $\Delta_B > 0$ in every seed. The permutation control pairs each action with a random state, leaving prompt format, sequence length and action text untouched.

Table 11: Truncated-training read. The escalation is cheap when the effect is large and expensive when it is small.

Domain	**Seeds**	**Full lift**	$\Delta_{10\%}$	$\Delta_{10\%}$ **perm.**	B^*	Δ_{B^*}
Arithmetic (constructed)	2	$+0.292$	$+0.100$	-0.025	9%	$+0.100$
Constraints (constructed)	2	$+0.345$	$+0.131$	-0.009	6%	$+0.058$
PersuasionForGood	2	$+0.084$	$+0.022$	-0.014	9%	$+0.021$
DealOrNoDeal	2	$+0.059$	$+0.014$	-0.018	17%	$+0.024$
HH-RLHF	4	$+0.033$	-0.025	—	46%	$+0.022$
PRM800K (paired)	4	-0.000	-0.045	-0.118	never	—

J.4 Per-Domain Hyperparameter Check for the Low-SL Rows

Both arms were re-trained on the three genuine-label domains with the smallest state-lift under three alternative configurations: learning rate 5×10^{-5} (vs. 2×10^{-5}) at 3 epochs, 2×10^{-5} at 5 epochs (vs. 3), and 5×10^{-5} at 5 epochs; everything else as in Table 3. The metric is the best held-out accuracy over epochs, so the 5-epoch value is at least the 3-epoch value by construction; identical entries mean the best epoch was ≤ 3 in both arms.

Table 12: Hyperparameter check, two seeds per cell (lift = conditioned − blind accuracy). Baseline is the reported configuration over all its seeds. No cell moves a domain's mean lift by more than 0.005; every individual lift lies within one baseline SD.

Domain	**LR**	**Epochs**	**Seed 42**	**Seed 43**	**Mean**	**Baseline**
ESConv	5×10^{-5}	3	$+0.008$	-0.003	$+0.003$	
	2×10^{-5}	5	$+0.012$	$+0.010$	$+0.011$	$+0.008 \pm 0.009$ (n=8)
	5×10^{-5}	5	$+0.008$	-0.003	$+0.003$	
HH-RLHF	5×10^{-5}	3	$+0.030$	$+0.035$	$+0.033$	
	2×10^{-5}	5	$+0.035$	$+0.028$	$+0.031$	$+0.030 \pm 0.012$ (n=8)
	5×10^{-5}	5	$+0.030$	$+0.035$	$+0.033$	
CaSiNo	5×10^{-5}	3	$+0.009$	$+0.035$	$+0.022$	
	2×10^{-5}	5	$+0.020$	$+0.035$	$+0.028$	$+0.027 \pm 0.022$ (n=13)
	5×10^{-5}	5	$+0.019$	$+0.040$	$+0.030$	

J.5 State-Stratified Evaluation (HH-RLHF)

To test whether the small PRM lift on HH-RLHF concentrates in specific state regions, we split the test set by state-vector distance from the mean into quartiles.

The PRM advantage on HH-RLHF is concentrated in extreme states (Q4: +7.5%) and absent for near-mean states (Q1: -1.3%). This suggests that even in a nominally state-independent domain, unusual conversational contexts create pockets of state-dependent quality that a PRM can exploit. However, the overall effect is small (+3.4%), consistent with near-zero state-lift.

Table 13: HH-RLHF accuracy by state distance quartile.

State Region	Blind Acc	PRM Acc	Lift
Q1 (near mean, $n = 375$)	0.637	0.624	−0.013
Q2–Q3 (moderate, $n = 750$)	0.621	0.659	+0.038
Q4 (extreme, $n = 374$)	0.604	0.679	+0.075

J.6 Published PRM–ORM Comparisons

Table 14 lists the published comparisons referred to in Section 5. They are context rather than validation: the state-lift values were computed on different datasets, label types and metrics from the published gaps.

Table 14: Published PRM vs. ORM performance comparisons used for calibration.

Paper	Domain	Benchmark	Metric	PRM Gap
Lightman et al. 2023	Math	MATH-500	Best-of-N acc	+5.8%
Lightman et al. 2023	Math	AP Calculus	Best-of-N acc	+17.8%
Math-Shepherd 2024	Math	MATH	Verification acc	+4.1%
Math-Shepherd 2024	Math	GSM8K	Verification acc	+1.4%
Setlur et al. 2025	Math	MATH	RL acc	+8.0%
AgentPRM 2024	Agents	WebShop	Best-of-64	+19.0%
AgentPRM 2024	Agents	BabyAI	Best-of-64	+6.1%
AgentPRM 2024	Agents	TextCraft	Best-of-64	+13.4%
CodePRM 2025	Code	LiveCodeBench	Pass@1	+1.5%
CodePRM 2025	Code	LCB Medium	Pass@1	+2.6%
Du et al. 2024	Dialogue	MultiWOZ	Combined	+10.4
Lee et al. 2025	General	MMLU-Pro	Best-of-N acc	0%

K Robustness Analyses

We conduct six robustness analyses addressing reviewer-relevant concerns: bootstrap confidence intervals, encoder sensitivity, persistence controls, mixed-effects comparison, sample size sensitivity, and nonlinear estimators.

K.1 Sample Size Calibration Guide

Table 15 provides practical guidance on minimum sample sizes for reliable state-lift estimation, based on our bootstrap analysis across three domains.

Table 15: Recommended minimum n for reliable state-lift estimation. "Reliable" means 95% CI width < 0.05 for the given SL regime. Based on bootstrap resampling (200 draws).

Expected SL regime	Min n (directional)	Min n (CI excludes 0)	Min n ($\pm$0.02)
High (SL > 0.1)	200	500	1,000
Moderate (SL 0.01–0.1)	500	2,000	5,000
Low (SL ≈ 0)	500	N/A (signal is null)	2,000

For practical use: if $n = 500$ gives SL > 0.10 with the lower CI bound above zero over at least three seeds, PRM investment is justified. A near-zero value at any n is uninformative on its own and routes to Step 2 of the protocol. With the corrected registry, every threshold in $[0.06, 0.39]$ produces the identical classification of the ten domains, so moderate estimation error in SL does not change the recommendation.

K.2 Bootstrap Confidence Intervals

We compute state-lift on 200 bootstrap resamples at various sample sizes to characterize estimation uncertainty.

Table 16: Bootstrap 95% CIs for state-lift at different sample sizes. $n \geq 500$ produces stable estimates for high-SL domains; $n \geq 2000$ is needed for low-SL domains.

Domain	n	Mean SL	SD	95% CI
Math	100	-0.055	0.098	$[-0.27, 0.10]$
	500	0.257	0.027	[0.20, 0.31]
	2000	0.480	0.012	[0.46, 0.50]
ESConv	500	-0.016	0.059	$[-0.06, 0.04]$
	2000	0.020	0.010	[0.00, 0.04]
	5000	0.027	0.006	[0.02, 0.04]
HH-RLHF	500	-0.054	0.026	$[-0.10, -0.01]$
	2000	-0.010	0.007	$[-0.02, 0.00]$

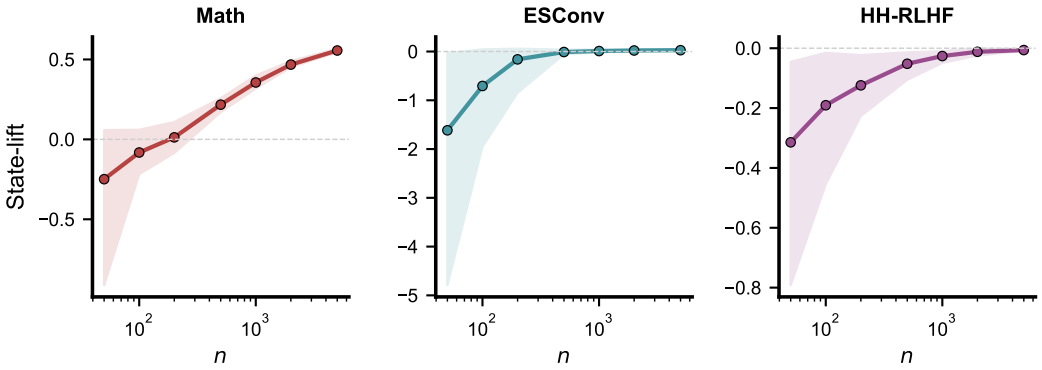

Figure 8: State-lift stabilizes with increasing sample size. Shaded regions show 95% CIs from 50 random draws at each n. For Math (high SL), estimates are reliable at $n \geq 500$. For ESConv and HH-RLHF (low SL), $n \geq 2000$ is needed for the CI to exclude zero.

K.3 Encoder Sensitivity

We test three sentence encoders with different architectures and training objectives.

The qualitative ordering is invariant to encoder choice (Math $\gg$ ESConv $>$ HH-RLHF for all three encoders). Absolute state-lift values range from 0.13 to 0.53 for Math across encoders, suggesting that practitioners should compare relative values within an encoder rather than absolute thresholds across encoders.

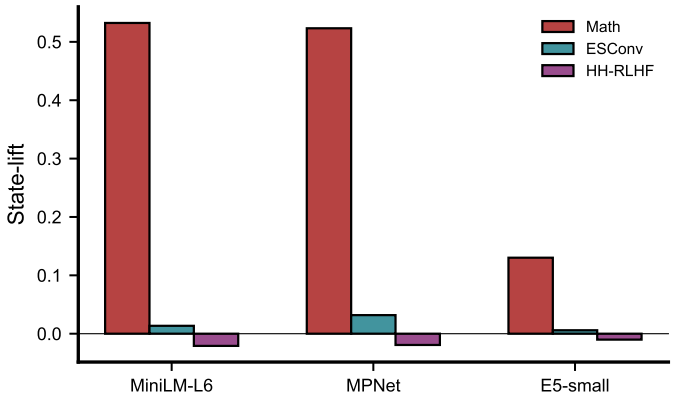

Figure 9: State-lift computed with three different sentence encoders. The ordering Math $\gg$ ESConv $>$ HH-RLHF is preserved across all encoders, though absolute magnitudes vary. E5-small produces lower values but the same ranking.

K.4 Persistence Controls

A key concern is whether state-lift measures genuine quality state-dependence or merely state persistence (autocorrelation). We test this with three controls: (1) *lagged state*: use z_{t-1} instead of z_t—if SL is driven by persistence, the lagged state should be equally predictive; (2) *block shuffle*: shuffle states within blocks of 50—destroys local temporal structure while preserving marginal distribution; (3) *residualization*: zero out the first principal component (typically the most autocorrelated dimension).

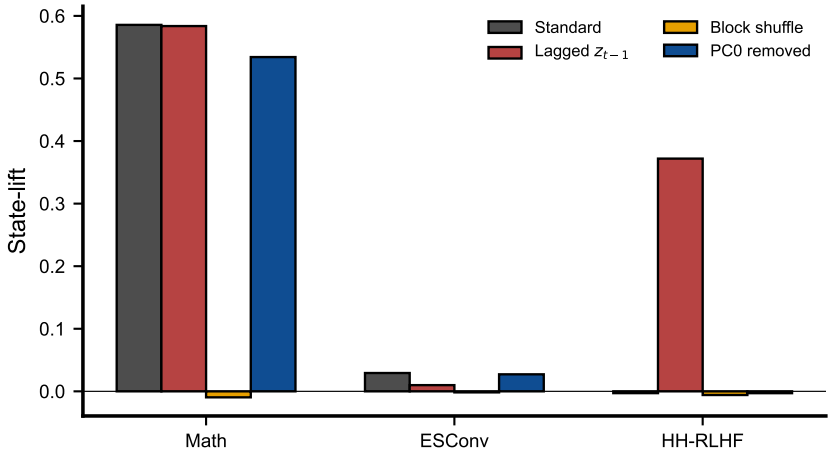

Figure 10: Persistence controls. Math: all controls give similar SL, confirming SL is genuine (not persistence-driven, since autocorrelation = 0.002). ESConv: lagged SL drops to 0.34$\times$ standard SL, confirming two-thirds of SL is genuine. HH-RLHF: lagged state *increases* SL dramatically (persistence masquerades as quality state-dependence), confirming the proxy pitfall.

The most striking finding is HH-RLHF: using the lagged state z_{t-1} produces SL $= 0.37$, far higher than the current-state SL $= -0.003$. This means earlier conversational turns are predictive of response quality—not because quality depends on state, but because conversations with better early turns tend to have better responses throughout. The current state z_t adds no information beyond this baseline persistence. This confirms and strengthens the proxy pitfall analysis.

K.5 Mixed-Effects vs. R^2-Difference Estimator

We compare our R^2-difference estimator with a variance-decomposition approach that directly estimates $\mathrm{Var}_a(r_{\text{blind}})$ and $\mathbb{E}_a[\mathrm{Var}_s(R|a)]$.

Table 17: Comparison of state-lift estimators.

Domain	R^2-diff	Var-decomp	Clipped [0,1]
Math	0.586	0.464	0.586
ESConv	0.029	0.031	0.029
HH-RLHF	−0.003	0.002	0.000

Both estimators agree on the ordering. The R^2-difference can produce values outside $[0, 1]$ (negative for HH-RLHF), which we clip in practice. The variance-decomposition estimator naturally stays in $[0, 1]$ and could be preferred when calibrated estimates are needed.

K.6 Nonlinear Estimators

We test whether the linear Ridge estimator underestimates state-dependence by comparing with kernel Ridge (RBF) and MLP (64, 32 hidden units) estimators.

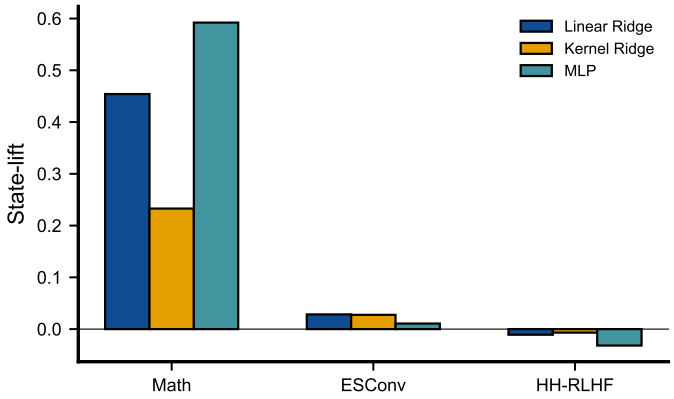

Figure 11: Nonlinear estimators: MLP finds higher state-lift than linear Ridge for Math (0.59 vs. 0.45), confirming that linear SL is a lower bound. Ordering is preserved across all estimators.

The MLP estimator finds SL $= 0.59$ for Math versus 0.45 for linear Ridge, confirming that our linear estimate is conservative. The ordering (Math $\gg$ ESConv $\approx$ HH-RLHF) is preserved across all estimators. This means that when linear state-lift is high, the true state-dependence is at least as high; when linear state-lift is near zero, nonlinear estimators also find no signal.

K.7 Robustness of Regret Bound to Distributional Assumptions

We test whether the upper bound (Theorem 1(ii)) holds under violations of the Gaussian i.i.d. assumption, sweeping state-lift from 0.01 to 0.9 ($K = 5$, 20,000 trials per point, 90 points per condition).

The bound is *more* conservative under correlated and heavy-tailed noise (lower ratios). Intuitively, correlation reduces the effective number of independent "flippable" actions, and heavy-tailed distributions have lower expected maxima than Gaussians of the same variance. The Gaussian i.i.d. assumption in Theorem 1 is therefore sufficient for the upper bound: real deviations from this assumption make the bound *looser*, not tighter.

Table 18: Regret bound robustness. Max ratio = max(simulated regret / bound) across all state-lift values. The bound holds (ratio < 1) under all tested conditions.

Condition	**Max ratio**	**Violations**	**Bound holds?**
Gaussian i.i.d. (baseline)	0.413	0 / 90	Yes
Correlated $\rho = 0.3$	0.424	0 / 90	Yes
Correlated $\rho = 0.6$	0.411	0 / 90	Yes
Correlated $\rho = 0.9$	0.404	0 / 90	Yes
Heavy-tail $t(\text{df} = 3)$	0.310	0 / 90	Yes
Heavy-tail $t(\text{df} = 5)$	0.371	0 / 90	Yes

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the paper's contributions: (1) state-lift as a minutes-scale, one-directional screen for the PRM–ORM decision; (2) a regret bound (Theorem 1) with a derived threshold; (3) an admissibility precondition checked by a model-free audit (Section 4.3); (4) matched multi-seed reward-model training on seven domains (Table 3) with a fixed-state selection evaluation; (5) a protocol that escalates near-zero readings to a truncated-training check. Scope limits are stated where they apply: readings above threshold occur only on context-correctness labels, and probes cannot read computation-dependent state-dependence (Section 7).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses limitations including: state-lift does not detect symbolic-reasoning correctness in math; context-correctness labels inflate SL by construction; sentence embeddings may fail for structurally-differentiated actions; the Gaussian regret bound is approximate; and downstream evaluation is limited to GSM8K. Section 4.5 additionally reports a negative result (HH-RLHF improvement-proxy false positive).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 1 (State-Blind Regret Bound) is stated in Section 3 with all assumptions made explicit: a Gaussian noise model $R(s,a) = \mu_a + \sigma \cdot Z_a(s)$ with Z_a independent standard normals and $\sigma^2 = \mathrm{SL} \cdot \mathrm{Var}(R)$, K actions, and gap $\Delta = \mu_{(1)} - \mu_{(2)}$. A proof sketch appears immediately after the statement; full proofs of all four parts (degeneracy at $\mathrm{SL} = 0$, the $\sqrt{2 \ln K}$ upper bound, the exact two-action lower bound, and the regime asymptotics) are provided in Appendix C, along with a monotonicity proposition. The Gaussian i.i.d. assumption and its robustness to violations (correlated noise with $\rho \in \{0.3, 0.6, 0.9\}$ and heavy-tailed t distributions with df $\in \{3, 5\}$) are tested numerically in Appendix K (Table 18). All theorems and equations are numbered and cross-referenced.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 fully specifies the state-lift computation pipeline (sentence-encoder embedding with all-MiniLM-L6-v2, PCA to $D = 16$, two Ridge regressions, $\mathrm{SL} = R^2_{\text{conditioned}} - R^2_{\text{action}}$). Section 5 and Appendix J fully specify the LLM reward model training (Llama-3.1-8B-Instruct with LoRA rank 8, alpha 16, query and value projections; 3 epochs; effective batch size 16; learning rate 2×10^{-5} with AdamW; max sequence length 512). The downstream beam-search protocol (Appendix F, Table 7) and the seven environment-derived quality signals (Table 4) are described in full. The frozen Llama-8B hidden-state probing pipeline used to test the interaction-level boundary is described in Section 5 and visualized in Appendix G. Appendices D, E, and K provide proxy-pitfall details, shuffled-state controls (Table 6), bootstrap CIs (Table 16), encoder sensitivity, persistence controls, and nonlinear estimator comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets are publicly available and cited in Appendix B. Source code for the state-lift estimator, data preprocessing, LLM reward model training (LoRA), multi-seed evaluation, SDI-style computation, and figure generation is included as supplementary material (`state_lift_code.zip`). The base model (Llama-3.1-8B-Instruct) is publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies all details for state-lift computation (encoder, PCA dimension $D = 16$, Ridge probe choice, R^2-difference vs. variance-decomposition estimators in Table 17). Section 5 specifies LLM training: Llama-3.1-8B-Instruct, LoRA rank 8 (alpha 16, query/value projections), 3 epochs, batch size 4 with gradient accumulation 4 (effective 16), learning rate 2×10^{-5} with AdamW, max sequence length 512. Appendix J adds per-epoch metrics (Table 10), state-stratified evaluation (Table 13), and the literature-calibration table (Table 14). Beam-search hyperparameters (4 candidates per step, 100 GSM8K test problems) are specified in Appendix F. Appendix K provides sensitivity analyses for sample size (Table 15, Figure 8), encoder choice (Figure 9), persistence controls (Figure 10), and probe nonlinearity (Figure 11), justifying the headline choices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Bootstrap 95% confidence intervals (200 resamples) for state-lift across sample sizes ($n \in \{100, 500, 2000, 5000\}$) are reported in Appendix K (Table 16, Figure 8) for Math, ESConv, and HH-RLHF, with shaded-region plots showing CI shrinkage with increasing n. Permutation tests are used for the shuffled-state artifact checks (Appendix E, Table 6), including a permutation $z = 11.3$ ($p < 0.0001$) for Mind2Web. The synthetic regret-bound validation reports a tight correlation between $\sqrt{\mathrm{SL} \cdot \mathrm{Var}(R)}$ and simulated regret (Figure 2). Robustness of the regret bound under correlated ($\rho \in \{0.3, 0.6, 0.9\}$) and heavy-tailed ($t$ with df $\in \{3, 5\}$) noise is reported as max-ratio statistics over 90 sweep points with 20,000 trials each (Table 18). All confidence interval methods (bootstrap, permutation) and assumptions are stated explicitly.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).

- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix J states that all LLM experiments use NVIDIA L40S GPUs (48 GB VRAM) with Llama-3.1-8B-Instruct + LoRA. Per-domain wall-clock times are reported per epoch in Table 10: blind models train in roughly 100–300 s/epoch, conditioned models in 250–800 s/epoch, depending on the domain. State-lift computation itself is described as $\sim$5 minutes on CPU with no GPU required (Section 6). The full practitioner protocol is dominated by an optional pilot experiment estimated at $\sim$1 GPU-hour, $\sim 100\times$ cheaper than full PRM training. The paper's headline cost-efficiency claim (minutes-scale diagnostic vs. hours-scale training) is therefore quantified end-to-end.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work analyzes existing public datasets and trains lightweight LoRA adapters on a publicly available pretrained model (Llama-3.1-8B-Instruct). It involves no human subjects, no personally identifiable information, no scraped private data, and no deployed system. Standard research-use licenses for all assets are respected, and dialogue datasets such as ESConv and HH-RLHF are used in their original anonymized released forms.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is a foundational diagnostic and reward-design methodology paper. It introduces a variance-decomposition statistic and a regret bound that help practitioners decide between process and outcome reward models *before* committing to expensive training. It does not introduce new generative capabilities, deployed systems, or models with a direct path to societal impact. The downstream effect is to make reward-model design more compute-efficient, which is broadly positive for accessibility but does not raise specific positive or negative impacts that warrant standalone discussion.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new pretrained models, scraped datasets, or generative systems. The trained LoRA reward-model adapters are research artifacts of the experimental validation and are not the contribution; all base models (Llama-3.1-8B-Instruct) and datasets (GSM8K, Glaive, CaSiNo, ESConv, CodeContests, HH-RLHF, Mind2Web, MathDial, PRM800K) are pre-existing public releases.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (GSM8K [5], CaSiNo [3], ESConv [18], CodeContests [16], Anthropic HH-RLHF [2], Mind2Web [6], PRM800K [17], Glaive Function Calling v2, MathDial) and the base model Llama-3.1-8B-Instruct are cited at first use with the original papers. All are used under their respective research-use licenses (predominantly MIT,

Apache 2.0, and the Llama Community License). The sentence encoder all-MiniLM-L6-v2 is used under Apache 2.0. The LoRA implementation [9] and Hugging Face PEFT/transformers libraries are likewise used under Apache 2.0.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new datasets or pretrained model checkpoints are introduced or released at submission time. Code implementing the state-lift estimator, the LoRA reward-model training pipeline, the proxy-pitfall analysis, and the GSM8K beam-search experiment will be released publicly with full documentation after the review process.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or any research with human subjects. All human-labeled data (e.g., HH-RLHF preferences, PRM800K step-correctness ratings) is consumed from existing public datasets whose original collection procedures are documented in their respective publications.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research; no IRB approval is required. Pre-existing human-labeled datasets (HH-RLHF, PRM800K, ESConv, CaSiNo) are used under their original release terms.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are used as the empirical evaluation object of this work, not as a writing or ideation tool. Specifically, Llama-3.1-8B-Instruct is fine-tuned with LoRA as both a blind (ORM) and state-conditioned (PRM) reward model across six domains to validate that the cheap state-lift diagnostic predicts full reward-model training outcomes (Section 5, Appendix J). Llama-3.1-8B is also used to extract frozen hidden-state representations on PRM800K (Appendix G) to test whether richer representations close the diagnostic gap with end-to-end training; the resulting comparison between encode-separately methods (linear/kernel/MLP probes on MiniLM and Llama-8B hidden states) and end-to-end training is the central evidence for the interaction-level diagnostic boundary. Additionally, Llama-8B is used to generate solution candidates and step-level samples for the GSM8K beam-search experiment (Appendix F). All architectures, hyperparameters, and prompting choices are specified in Section 5 and Appendix J.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.