Online Reward-Punishment Learning from Fixed-Channel Perceptual Event Streams without Environment Rewards

Anonymous submission

Abstract

We study online reward-punishment learning when the environment emits no environment-provided scalar reward or evaluative label. At each time step the agent receives only a fixed-channel perceptual packet p_t containing channels such as vision, text, audio, proprioception, and sensor values. Quantities such as pain, spice, energy, cognitive error, contact, damage, and actionability are treated as data-driven perceptual dimensions; their valence is inferred from observed transition consequences rather than supplied by the environment.

OHIRL separates prediction, residual dynamics, trajectory evaluation, and policy learning. A neural self-supervised predictor M_ψ learns posterior next-packet expectations $(p_t, a_t) \mapsto \hat{p}_{t+1}$. The observed packet p_{t+1} yields signed residuals $e_{t+1} = p_{t+1} - \hat{p}_{t+1}$; a residual-dynamics predictor D_ω models how those residuals evolve; and a trajectory evaluator $\mathcal{C}_\eta$ maps completed residual trajectories to internal value evidence Y^{post}. The learner B_ξ is trained from this post-transition evidence and supplies the internal reward/value estimates used for later policy updates and action scoring. The reward-free protocol exposes perceptual transitions while withholding environment scalar rewards, delayed external evaluators, and action-goodness labels.

Under identifiability, coverage, and realizability assumptions, we give a formal bound: uniform error of B_ξ relative to the ideal internal posterior-residual reward-punishment function plus RL optimization error yields bounded policy suboptimality. In a 2x2-XOR perceptual-packet task, medicine and chili have opposite learned value under visually defined XOR contexts. The same pain or spice increase can be positive or negative, and anesthetic lowers pain and cognitive error while producing negative learned consequence. The learned MLP B_ξ reaches 0.952 balanced reward-sign accuracy. A full online-interleaved audit then updates MLP M_ψ, MLP B_ξ, and Q-learning in the same reward-free transition stream: M_ψ reaches holdout $R^2 = 0.907$, B_ξ reaches 0.940 sign accuracy, and the learned policy reaches 0.979 optimal-action accuracy. Immediate packet scores, M_ψ prediction-error rewards, shuffled B_ξ targets, zero reward, and cognitive-error reduction collapse under the same policy audit.

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

1 Introduction

Many reinforcement-learning environments expose a scalar reward r_t. A first-person agent may instead receive only the next sensory event: pixels change, a sound or text channel changes, a contact sensor changes, energy changes, and motor actionability changes. Such channel values can become value-relevant through consequences while entering the agent only as observations.

We ask whether an agent can learn its own reward-punishment function online from a stream of perceptual events and then use that learned function to acquire a policy. One event is one time step of a fixed-channel sensory bundle,

$$\begin{aligned} p_t &= \{(x_t^c, m_t^c)\}_{c\in\mathcal{C}}, \\ \mathcal{C} &= \{\text{vision}, \text{text}, \text{audio}, \text{sensor}, \text{proprio}, \ldots\}. \end{aligned} \tag{1}$$

where m_t^c is a channel mask. The environment transition is

$$p_{t+1} \sim P_{env}(\cdot \mid p_t, a_t), \qquad \text{reward channel absent.} \tag{2}$$

Pain, stress, spice, energy, cognitive error, contact, and actionability are dimensions of p_t. Their valence is learned from how actions change posterior expectations and later packet dynamics.

OHIRL implements this setting with role separation. M_ψ is a self-supervised predictor of next-event posterior expectations. Its prediction defines signed residuals $e_{t+1} = p_{t+1} - M_\psi(p_t, a_t)$ once the next packet is observed. D_ω predicts how residuals evolve and exposes residual-surprise evidence ζ. $\mathcal{C}_\eta$ evaluates completed post-transition residual trajectories into internal evidence targets Y^{post}. B_ξ is trained on those targets after evidence is observed. During future action selection or no-leakage audits, only predictions learned from past transitions may be used; in online RL updates, the same learned estimator can assign an internal reward after the observed transition, exactly as ordinary RL updates use rewards after acting. Raw sensor dimensions, prediction error, curiosity scores, external evaluators, and fixed one-step scalarizations appear as controls.

The evidence emphasizes auditable reward-free controls: the agent receives perceptual transitions while scalar rewards, delayed scalar evaluators, and action-goodness labels remain outside the training interface. We distinguish this from the agent's own internal evaluator: $\mathcal{C}_\eta$ is constructed after observing consequences and is used to produce learning tar-

gets, not to select the action that generated those consequences. In online RL experiments, internal reward assignment is post-transition; in decision-time/no-leakage audits, action scores are predicted from current packet/action variables only. The core nonlinear task uses a 2x2 visual packet in which medicine value depends on $v_{00} \oplus v_{11}$ and chili value depends on $v_{01} \oplus v_{10}$. Identical immediate pain or spice changes receive opposite reward-punishment signs under different visual contexts. We then run an online-interleaved audit where M_ψ, distributional B_ξ, and the Q policy are updated in the same no-reward stream, closing the loop from interaction to policy learning.

Our contributions are: (1) a fixed-channel event-stream formulation in which the environment updates p_t but emits no environment scalar reward; (2) an online IRL/RL algorithm with an explicit signal-access protocol separating M_ψ posterior prediction, D_ω residual-dynamics prediction, $\mathcal{C}_\eta$ post-transition trajectory evaluation, distributional B_ξ internal value-evidence learning, and RL policy learning; (3) a conditional policy-bound proposition linking B_ξ estimation error to policy suboptimality under identifiability, coverage, and realizability; (4) posterior-residual internal targets aligned with the released implementation's no-reward packet stream; (5) non-identifiability audits separating pain, spice, energy, and error-reduction from learned reward labels; and (6) integrated online and family-level controls showing that the chain learns contextual policies without environment rewards or external evaluators.

2 Method

Event state and role separation. The agent observes fixed-channel packets p_t. The channel schema is fixed by the interface, although channels can be masked or empty. The sensor region may contain energy, pain, stress, spice, cognitive error, actionability, contact, and damage. These perceptual dimensions are generated by sensors; their valence is inferred from action-conditioned posterior-residual trajectory evidence, not from a privileged reward slot.

The self-supervised predictor learns posterior next-packet expectations,

$$L_M(\psi) = \|M_\psi(p_t, a_t) - p_{t+1}\|_2^2, \tag{3}$$

and defines the signed residual

$$e_{t+1} = p_{t+1} - \hat{p}_{t+1}, \qquad \hat{p}_{t+1} = M_\psi(p_t, a_t). \tag{4}$$

A residual's value evidence is determined by direction, persistence, recovery, and downstream controllability; large magnitude by itself marks mismatch or uncertainty. The post-transition evaluator therefore uses a residual-history feature

$$z_t = \Phi(p_{t:t+K}, a_t, \hat{p}_{t+1:t+K}, e_{t+1:t+K}, \zeta_{t+2:t+K}), \tag{5}$$

where ζ denotes residual-surprise evidence from a distinct residual-dynamics predictor D_ω, e.g.,

$$\hat{e}_{t+1} = D_\omega(e_t, p_t, a_t, h_t), \qquad \zeta_{t+1} = e_{t+1} - \hat{e}_{t+1}. \tag{6}$$

B_ξ learns a conditional distribution over internal value evidence. Depending on the audit, z_t may denote the completed post-transition evidence feature used for supervised update, or the pre-decision packet/action/history feature used for held-out action scoring:

$$B_\xi(z_t) = q_\xi(Y_t^{post} \mid z_t), \tag{7}$$

with heads for mean, lower quantiles, negative-event probability, and positive/negative evidence mass. In scalar audits the risk-neutral mean is used; in distributional audits the policy utility is

$$U_\xi(z_t) = \mu_\xi(z_t) - \lambda_\sigma \sigma_\xi(z_t) - \lambda_c P_\xi(Y_t^{post} < -\tau \mid z_t). \tag{8}$$

RL updates from the learned internal utility under the reward-free protocol. The timing is explicit: Y^{post} is produced only after the consequence window is observed, while action selection can use only previously learned estimates. Raw prediction loss remains reserved for self-supervised modeling.

Algorithmic contribution. The algorithmic issue in this setting is to keep prediction, residual dynamics, post-transition evidence construction, value-estimator training, and action selection temporally separated. OHIRL enforces a signal-access discipline. At action time the agent may use the current packet p_t, the current parameters of M_ψ, D_ω, and B_ξ, and its policy state. After the transition, M_ψ receives the self-supervised next-packet target p_{t+1}, D_ω updates on residual evolution, and the signed residual trajectory becomes observable. After the residual/consequence window accrues, $\mathcal{C}_\eta$ produces Y_t^{post} samples or summaries. B_ξ then learns from this post-transition evidence; online RL updates can use the assigned internal reward after the transition, while pre-decision audits use only B_ξ scores predicted from current (p_t, a_t, h_t). The controls below test four separations: prediction loss versus reward, sensor deltas versus valence, shuffled/immediate targets versus learned posterior-residual value, and coverage versus policy hard-coding.

Posterior-residual internal target. The event-stream target is constructed inside the agent from posterior-residual evidence. It is an internal post-transition evaluative target, not an environment-returned reward slot. Let

$$\tau_t^a = \{p_{t:t+K}^a, \hat{p}_{t+1:t+K}^a, e_{t+1:t+K}^a\} \tag{9}$$

be the observed or model-predicted packet/residual trajectory after action a. The released pure-residual experiments instantiate $\mathcal{C}_\eta$ as a fixed monotone algebraic map over residual norms only. With $n_k = \|e_{t+k}\|_2$, the released implementation uses

$$\begin{aligned} \mathcal{C}_\eta(\tau) = 1.35(n_1 - n_K) - 0.38 \frac{1}{K-1} \sum_{k=2}^{K} n_k \\ - 0.75[n_K - n_1]_+. \end{aligned} \tag{10}$$

The coefficients are fixed across families and actions in the released artifact; they are not selected from environment reward, task labels, or oracle actions. Because such constants could be mistaken for task-specific shaping, the supplement includes a coefficient-origin audit. Equal-unit standardized, raw-equal, and random-monotone orientations preserve 0.925, 0.935, and 0.922 of the released top-action rankings, while sign inversion preserves 0.000. Thus the stable information is the monotone direction–recovery positive,

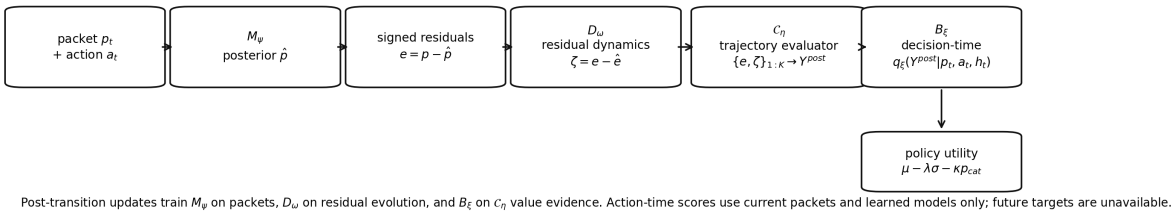

Figure 1: Role-separated event-stream learning. The environment returns p_{t+1} and no scalar reward. M_ψ predicts posterior next-packet expectations; D_ω predicts residual dynamics and exposes residual-surprise evidence ζ; $\mathcal{C}_\eta$ evaluates completed post-transition residual trajectories into internal value evidence Y^{post}. B_ξ is trained from observed evidence windows and provides learned internal reward/value estimates for later policy updates and action scoring. At action time, future e, ζ, and Y^{post} are unavailable.

persistence/growth negative–rather than the exact numeric recipe. The main target is the no-op-normalized posterior-residual advantage

$$Y_t^{post} = \mathcal{C}_\eta(\tau_t^{do(a_t)}) - \mathcal{C}_\eta(\tau_t^{noop}). \tag{11}$$

The no-op branch makes the target an advantage over passive continuation from the same initial event. The $\mathcal{C}_\eta$ computation occurs after packet observations inside the agent/evaluation code, while action selection uses the learned B_ξ estimate. The artifact includes a static genericity audit of the $\mathcal{C}_\eta$ implementation: the target function reads residual arrays and algebraic statistics, while family names, action names, visual rules, channel semantics, and oracle labels appear outside target computation.

When intermediate residual evidence is accumulated, Eq. 11 can be generalized to

$$Y_t^{post,sum} = \sum_{k=0}^{K} \gamma^k \left[\mathcal{C}_\eta(\tau_{t,k}^{do(a_t)}) - \mathcal{C}_\eta(\tau_{t,k}^{noop}) \right]. \tag{12}$$

We keep Eq. 11 as the main experimental target because it matches the released no-op-baseline implementation, and report Eq. 12 as a compatible extension.

Signal discipline. The implementation records a compact signal-access audit for each experiment. For pre-decision action scoring and held-out no-leakage audits, action scores use current p_t, candidate a_t, the current M_ψ posterior predictor, the current D_ω residual-dynamics predictor when present, the current B_ξ utility, and the policy state. For online RL updates, the internal reward assignment is post-transition: after M_ψ trains on $(p_t, a_t) \mapsto p_{t+1}$ and D_ω trains on residual evolution, $\mathcal{C}_\eta$ evaluates completed posterior-residual windows and B_ξ trains to predict those value-evidence targets. Success labels, public rewards, simulator labels, and future-window targets are logged as audit/evaluation fields.

Conditional Policy Bound

The formal guarantee is scoped by three requirements. First, *identifiability*: the ideal internal reward-punishment function $B_{post}^\star$ is identifiable from packet histories, posterior residuals, residual-evolution evidence, and post-transition trajectory evaluation. Second, *coverage*: exploration visits the state-action regions on which the learned policy may act. Third, *realizability*: the function class of B_ξ can approximate $B_{post}^\star$ on that covered region.

Algorithm 1 Reward-Free Online Event-Stream IRL/RL

1: Observe fixed-channel event p_t.
2: Choose a_t from the current policy, using only current p_t, M_ψ, D_ω, B_ξ, and policy state.
3: Environment returns p_{t+1} and protocol metadata; reward/evaluator/action-label fields remain empty.
4: Update M_ψ from $(p_t, a_t) \mapsto p_{t+1}$ by Eq. 3.
5: Compute signed residual $e_{t+1} = p_{t+1} - M_\psi(p_t, a_t)$; update D_ω on residual evolution and store ζ evidence.
6: After the evidence window accrues, evaluate trajectories with $\mathcal{C}_\eta$ and construct Y_t^{post} by Eq. 11; it is unavailable before action.
7: Update distributional B_ξ toward Y_t^{post} samples or summaries by Eq. 7.
8: Update the RL policy using the learned internal reward/value estimate from B_ξ.
9: Log success metrics, simulator labels, public rewards, external evaluators, and future targets in evaluation/audit records.

Proposition. Consider a discounted event-stream control problem with discount $\gamma \in [0, 1)$. Assume that on the covered region

$$\|B_\xi - B_{post}^\star\|_\infty \le \epsilon_R. \tag{13}$$

Let π_θ be the policy returned by the RL optimizer using B_ξ, and assume its optimization error under B_ξ is bounded by

$$V_{B_\xi}^{\pi_\theta^\star}(p) - V_{B_\xi}^{\pi_\theta}(p) \le \frac{2\epsilon_Q}{1-\gamma}, \tag{14}$$

where $\pi_\theta^\star$ is optimal for the learned reward B_ξ. Then for the ideal internal posterior-residual reward-punishment function $B_{post}^\star$,

$$V_{B_{post}^\star}^{\pi^\star}(p) - V_{B_{post}^\star}^{\pi_\theta}(p) \le \frac{2\epsilon_R}{1-\gamma} + \frac{2\epsilon_Q}{1-\gamma}, \tag{15}$$

where $\pi^\star$ is optimal for $B_{post}^\star$.

Proof. For any fixed policy π, replacing $B^{\star}_{post}$ with B_ξ changes the discounted return by at most the discounted sum of per-step reward-punishment estimation errors:

$$\left|V^{\pi}_{B_\xi}(p) - V^{\pi}_{B^{\star}_{post}}(p)\right| \leq \sum_{t=0}^{\infty} \gamma^t \epsilon_R = \frac{\epsilon_R}{1-\gamma}. \tag{16}$$

Therefore

$$V^{\pi^\star}_{B^{\star}_{post}}(p) - V^{\pi_\theta}_{B^{\star}_{post}}(p) \leq V^{\pi^\star}_{B_\xi}(p) - V^{\pi_\theta}_{B_\xi}(p) + \frac{2\epsilon_R}{1-\gamma} \tag{17}$$

$$\leq V^{\pi^\star_\theta}_{B_\xi}(p) - V^{\pi_\theta}_{B_\xi}(p) + \frac{2\epsilon_R}{1-\gamma} \tag{18}$$

$$\leq \frac{2\epsilon_Q}{1-\gamma} + \frac{2\epsilon_R}{1-\gamma}. \tag{19}$$

This conditional transfer bound applies when an internal reward-punishment function is identifiable on the covered support. The experiments reduce ϵ_R through posterior-residual evidence and coverage and enforce action-before-target ordering for the policy optimizer.

3 Related Work

Intrinsic motivation and prediction-error rewards. Intrinsic motivation methods such as ICM, RND, Plan2Explore, DIAYN, E3B, and URLB encourage exploration through prediction error, novelty, disagreement, information gain, skill diversity, or state coverage (Pathak et al. 2017; Burda et al. 2019; Sekar et al. 2020; Eysenbach et al. 2019; Henaff et al. 2022; Laskin et al. 2021). Recent systems work also emphasizes that intrinsic-reward results depend strongly on implementation details and standardized evaluation (Yuan et al. 2024). Classic intrinsic-motivation work already distinguished novelty, prediction progress, and self-generated goals (Schmidhuber 1991; Oudeyer and Kaplan 2007; Bellemare et al. 2016; Tang et al. 2017). A central behavioral trap is that prediction-error curiosity can prefer unpredictable observations instead of useful consequences. Tinker, Doya, and Tani study this issue under a Free-Energy-Principle formulation and show that prediction-error curiosity is vulnerable to observational-noise traps, while hidden-state curiosity is more robust (Tinker, Doya, and Tani 2024). More recent noise-robust curiosity work likewise rewards learning progress or prediction-error improvement instead of raw error (Hou, An, and Du 2025; Bhaskara and Wang 2026). OHIRL uses M_ψ for self-supervised next-packet prediction and empirically separates this modeling loss from the policy reward.

Reward-free and unsupervised RL. Reward-free RL studies exploration without task rewards and later planning for reward functions revealed after exploration (Jin et al. 2020; Wagenmaker et al. 2022). Learning rewards from feedback has also been studied in inverse RL and preference-based RL (Ng, Harada, and Russell 1999; Abbeel and Ng 2004; Ziebart et al. 2008; Christiano et al. 2017; Du et al. 2024). Unsupervised RL benchmarks and skill-learning methods study reward-free pretraining followed by downstream task reward evaluation (Laskin et al. 2021; Eysenbach et al. 2019). OHIRL differs in protocol: the agent learns an internal reward-punishment estimator B_ξ from $\mathcal{C}_\eta$ value evidence computed on posterior-residual trajectories and uses it online for policy learning before any downstream task reward is supplied.

World models, active inference, and endogenous signals. World-model agents such as DreamerV3 demonstrate the utility of learned dynamics models for control (Hafner et al. 2023). Active inference separates epistemic and pragmatic quantities under outcome preferences (Friston 2010; Friston et al. 2017). Homeostatic RL studies regulation under internal variables and physiological needs (Keramati and Gutkin 2011; Juechems and Summerfield 2019); this literature motivates the first-person packet formulation, while B_ξ supplies learned valence from $\mathcal{C}_\eta$ posterior-residual value evidence. In our protocol, first-person sensor changes are data, M_ψ learns posterior packet expectations, and signed value is learned by B_ξ without an environment-authored scalar reward.

4 Experiments

The main experiments share a reward-free interface: $env.step(a_t)$ returns observations, packet transitions, termination flags, and diagnostic metadata, while scalar reward, task-success labels, evaluator targets, and oracle action labels are unavailable to model and policy updates. We order the evidence from externally recognizable controls to mechanism diagnostics. The main evidence is: (i) hidden-reward control on public Gymnasium tasks, (ii) a corrected public-data no-leakage action-scoring audit, (iii) a module-role ablation testing whether $\mathcal{C}_\eta$ can be replaced by prediction error or residual-dynamics proxies, and (iv) posterior-residual mechanism diagnostics on controlled packet families. The packet-family experiments are used to isolate identifiability and failure modes, rather than as the sole evidence base.

Experimental design and fairness. All fair baselines operate under the same reward-free interface. They may use observable state or packet transitions, because those are observations returned by the environment, but they do not receive scalar reward, success labels, goal-distance shaping, or action-goodness labels. Consequence-derived baselines such as prediction-error curiosity, learning progress, residual minimization, and residual-of-residual minimization are trained from past transitions and scored from the same decision-time information as OHIRL in no-leakage audits. Reward-provided controllers are reported separately as reference rows because they solve a different interface. The core packet audit uses 50 seeds and 1820 reward-free transitions per seed; logs store action-time packets, posterior-residual targets, model updates, and protocol flags.

Standard Hidden-Reward Controls

We first test whether OHIRL remains useful when the observation stream comes from public environments rather than a procedurally generated packet family. In these controls, the standard environment reward is withheld from same-protocol training. OHIRL observes only environment observations

Environment / method	Eval. return	Success / solved	Interface
CartPole OHIRL	**243.20±74.41**	**0.547±0.304**	hidden reward
CartPole pred.-error	80.80±27.97	–	hidden reward
CartPole RND	62.00±20.60	–	hidden reward
CartPole zero	9.32±0.13	–	hidden reward
Taxi OHIRL	**7.19±1.08**	**0.999±0.006**	hidden reward
Taxi transition-error	-1263.29	–	hidden reward
Taxi RND	-1282.99	–	hidden reward
Taxi zero	-771.93	–	hidden reward

Table 1: Hidden-reward public-environment controls over 30 seeds. Rewards are withheld from same-protocol training; returns and success are evaluation metrics.

and transitions, constructs internal post-transition evidence, and updates the policy from its learned internal signal. This protocol allows decoded observation-transition facts already present in the observation, but excludes simulator reward, success labels, and hand-coded action values.

Table 1 moves the public-environment evidence into the main comparison. It does not claim that OHIRL outperforms tuned reward-provided RL; instead it tests a different interface: whether an agent can recover policy-useful internal value evidence from observation-transition consequences when the scalar reward channel is hidden. The technical supplement reports the full audit records and protocol flags for these controls.

Corrected Public-Data No-Leakage Consequence Audit

We next evaluate decision-time action scoring on public handwritten-digit visual packets. The public dataset defines the observation distribution only; digit labels are not used as reward, evaluator target, or action label. Five non-isomorphic consequence families generate residual trajectories from these contexts, including same-event tail-risk competition, positive/negative component coexistence, delayed masking, low-ζ stable bad states, and high-ζ reconstruction. In 25 paired family/seed settings, every fair method learns its score from (p_t, a_t) on training consequences and uses only (p_t, a_t) on held-out images. The strongest distributional B_ξ row, the two-head positive/negative variant, reaches 0.466 risk-optimal action accuracy and 0.452 risk regret, compared with 0.190/0.761 for learned learning-progress, 0.119/0.885 for learned prediction-error curiosity, 0.250/0.772 for learned residual-of-residual minimization, and 0.196/0.824 for shuffled B_ξ. Point B_ξ reaches a similar 0.461/0.451. This audit separates OHIRL from residual or novelty shortcuts under equal pre-decision information.

Module-Role Ablation

A dedicated module-role ablation tests whether $\mathcal{C}_\eta$ can be replaced by M_ψ or D_ω proxies. Over five residual families and 30 seeds per family, B_ξ trained on $\mathcal{C}_\eta$ evidence reaches 0.688±0.145 optimal-action accuracy, compared with 0.413±0.090 for M_ψ prediction-error targets, 0.253±0.101 and 0.210±0.096 for D_ω predictability and surprise targets, 0.274±0.142 for shuffled $\mathcal{C}_\eta$, and 0.207±0.042 for zero/rest. Thus D_ω and $\mathcal{C}_\eta$ are functionally non-substitutable: D_ω models residual dynamics, while $\mathcal{C}_\eta$ supplies trajectory-level value evidence for B_ξ. A coefficient-origin audit compares the rankings induced by alternative monotone orientations: equal-unit, raw-equal, and random-monotone targets preserve 0.925, 0.935, and 0.922 of the released $\mathcal{C}_\eta$ top-action rankings, while sign inversion preserves 0.000.

Mechanism Diagnostic: XOR Perceptual-Packet Reward-Punishment Learning

Each packet has a 2x2 binary visual channel $[v_{00}, v_{01}; v_{10}, v_{11}]$ and sensor dimensions including pain, spice, energy, cognitive error, actionability, and tissue damage. Medicine is beneficial when $v_{00} \oplus v_{11} = 1$ and harmful when $v_{00} \oplus v_{11} = 0$. Chili is beneficial when $v_{01} \oplus v_{10} = 1$ and harmful when $v_{01} \oplus v_{10} = 0$.

Table 2 is the central non-identifiability audit. The medicine rows have identical pain and error changes, yet opposite learned value. The chili rows have identical pain and spice changes, yet opposite learned value. Anesthetic lowers pain and cognitive error and still has negative learned value. Thus pain, spice, and error are dimensions of p_t; their reward-punishment sign is learned by B_ξ from visual context and posterior-residual consequences.

Whole-Pipeline Online Audit

The sign audit proves that B_ξ can learn context-dependent reward-punishment signs. We next test the complete chain in one stream. For each transition, neural MLP M_ψ is updated self-supervised on $(p_t, a_t) \mapsto p_{t+1}$, neural MLP B_ξ is updated on posterior-residual no-op-advantage targets, and Q-learning updates from the learned B_ξ output. The audit records online interleaving, action-before-target ordering, transition-stream updates, and target-access flags.

At the final checkpoint, MLP M_ψ reaches next-packet holdout $R^2 = 0.907 \pm 0.006$, MLP B_ξ reaches balanced reward-sign accuracy 0.940 ± 0.037, and the online Q policy reaches 0.979 ± 0.035 optimal-action accuracy. Table 3 shows prediction-error reward at 0.006 accuracy, shuffled B_ξ targets at 0.345, and cognitive-error reduction selecting anesthetic in every seed. Full OHIRL's advantage comes from the learned reward-punishment estimator rather than prediction loss or local sensor improvement.

Environment-Family Generalization

We next test five reward-free perceptual-packet families that change the visual rule determining whether medicine or chili has positive posterior-residual consequence: diagonal XOR, row XOR, column XOR, mixed parity, and noisy-switch XOR. Each family keeps the same reward-free interface and uses online-interleaved MLP M_ψ + MLP B_ξ + Q learning. The protocol begins with a stratified coverage phase and then switches to epsilon-greedy online control. Each family is evaluated over 50 seeds with 800 reward-free transitions per seed.

Probe	Visual packet	Action	Δpain	Δspice	Δerror	Target Y / MLP B_ξ
Medicine XOR1 therapy	$[[1,0],[0,0]]$	take medicine	$+0.227$	0.000	$+0.090$	$+0.924$ / $+0.865$
Medicine XOR0 poison	$[[0,0],[0,0]]$	take medicine	$+0.227$	0.000	$+0.090$	-1.626 / -1.612
Chili XOR1 tolerant	$[[0,1],[0,0]]$	eat chili	$+0.207$	$+0.550$	$+0.030$	$+0.359$ / $+0.345$
Chili XOR0 sensitive	$[[0,0],[0,0]]$	eat chili	$+0.207$	$+0.550$	$+0.030$	-1.353 / -1.363
Anesthetic trap	$[[1,1],[0,1]]$	take anesthetic	-0.320	0.000	-0.350	-0.923 / -0.911

Table 2: XOR perceptual-packet sign audit. The same immediate pain/spice/error direction can receive opposite delayed reward-punishment signs depending on visual context.

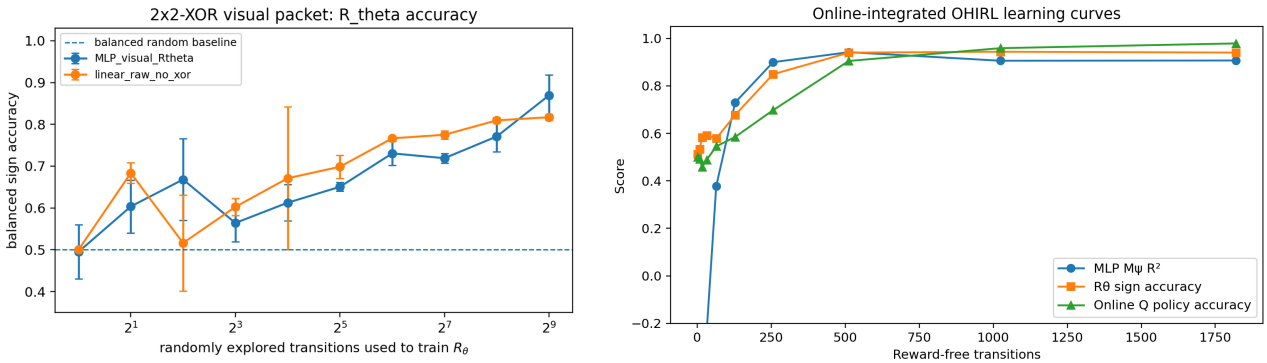

Figure 2: Left: MLP B_ξ reward-sign accuracy improves as exploration supplies transitions; accuracy starts near random and reaches 0.952 after 1800 transitions. Right: in the online-interleaved MLP pipeline, M_ψ, B_ξ, and the Q policy improve in the same reward-free transition stream.

Agent	Optimal action accuracy	Chosen internal Y^{post}	Anesthetic rate
Full online OHIRL: MLP M_ψ + B_ξ + online Q	**0.979±0.035**	**0.457±0.012**	**0.000±0.000**
Oracle true Y + Q	1.000±0.000	0.464±0.000	0.000±0.000
Shuffled B_ξ target + Q	0.345±0.207	-0.169±0.319	0.090±0.249
Zero reward Q	0.250±0.000	0.037±0.000	0.000±0.000
M_ψ prediction error as reward + Q	0.006±0.019	-0.916±0.075	0.093±0.063
Immediate packet score + Q	0.000±0.000	-0.930±0.000	1.000±0.000
Cognitive-error reduction + Q	0.000±0.000	-0.930±0.000	1.000±0.000

Table 3: Whole-pipeline online audit over 50 seeds. Full OHIRL updates MLP M_ψ, MLP B_ξ, and online Q in the same reward-free transition stream.

The learning curves pass a preflight audit: across families, M_ψ R^2 starts at -2.462 and reaches 0.907, B_ξ balanced sign accuracy starts at 0.495 and reaches 0.925, and policy accuracy starts at 0.500 and reaches 0.895. The online-order audit confirms that actions are selected before terminal targets are constructed; the environment returns no reward slot; and optimal-action accuracy, regret, and chosen internal Y^{post} are evaluation metrics only, not training targets.

Pure Posterior-Residual Learning Audit

The preceding family suite uses auditable no-op-normalized posterior-residual advantages saved in the implementation. To test the stronger claim that B_ξ can learn reward-punishment structure directly from M_ψ posterior-error dynamics, we run a separate pure posterior-residual full50 audit. In this audit the environment returns packet transitions and protocol metadata with reward/evaluator/action-label fields unavailable to training. The scalar target for B_ξ is computed by $\mathcal{C}_\eta$ from the signed residual sequence produced by M_ψ predictions and observed packet transitions using Eq. 10. Thus the target is a generic internal residual-statistics functional over residual recovery, persistence, and growth; family identity, XOR rules, action labels, and channel-specific reward signs live in the environment dynamics rather than in

Family	OHIRL acc.	Oracle acc.	Regret	Shuffled B_ξ	Zero reward	M_ψ-err. / immediate / err.-red.
col XOR	0.893	1.000	0.035	0.273	0.250	0.006 / 0.000 / 0.000
diag XOR	0.916	1.000	0.028	0.289	0.250	0.005 / 0.000 / 0.000
noisy-switch XOR	0.879	1.000	0.039	0.316	0.188	0.006 / 0.000 / 0.000
parity-mixed	0.908	1.000	0.031	0.299	0.313	0.011 / 0.000 / 0.000
row XOR	0.879	1.000	0.041	0.293	0.250	0.005 / 0.000 / 0.000
Aggregate	**0.895±0.095**	1.000	0.035	0.294	0.250	0.007 / 0.000 / 0.000

Table 4: Environment-family generalization. Full online OHIRL is positive in every family and remains above prediction-error, immediate-score, zero-reward, shuffled-B_ξ, and error-reduction controls.

Agent	Policy acc.	Regret	Anesthetic
Full posterior-residual OHIRL	**0.897±0.080**	0.0069	0.000
Privileged residual oracle	0.937	0.0017	0.000
Prediction-error reward	0.380	0.231	0.000
Immediate error minimization	0.098	0.736	0.551
Shuffled residual target	0.221	0.466	0.254
Zero reward	0.235	0.316	0.000

Table 5: Pure posterior-residual full50 audit. B_ξ is trained from M_ψ signed residual trajectories through the generic $\mathcal{C}_\eta$ in Eq. 10; the target map reads residual arrays and shared residual statistics.

the target map. The target is still an agent-side post-transition evaluator, so the claim is absence of environment reward, not absence of internal evaluation. The policy updates from the learned B_ξ prediction. The audit uses five nonlinear families, 50 seeds per family, and 512 reward-free transitions per seed under online-interleaved updates with initial stratified coverage.

The curve sanity check is also normal: across families, M_ψ R^2 starts at -2.139 and reaches 0.888, B_ξ sign accuracy starts at 0.617 and reaches 0.940, and policy accuracy starts at 0.247 and reaches 0.897. This result supports the posterior-residual mechanism: M_ψ supplies the evidence used by B_ξ, and the reward signal comes from learned residual structure.

Mechanism summary. The controlled packet-family diagnostics below isolate the posterior-residual mechanism after the public hidden-reward and public-context controls. They are not the sole evidence base; their role is to show why local observation scores, prediction error, cognitive-error reduction, shuffled value evidence, and residual-only proxies fail even when the consequence structure is fully auditable.

5 Discussion

The environment updates perceptual dimensions in p_t under a reward-free interface. Pain, pressure, spice, energy, damage, cognitive error, contact, and actionability enter as observations. M_ψ converts these streams into posterior expectations and signed residuals; D_ω characterizes residual evolution; $\mathcal{C}_\eta$ assigns internal value evidence to completed post-transition trajectories; and B_ξ learns from those targets to provide internal rewards after observed transitions and predicted value estimates for later action scoring. Replacing B_ξ with raw M_ψ prediction error gives 0.006 optimal-action accuracy in the online audit, separating prediction loss from learned reward.

The formal result scopes generalization through identifiability, coverage, and realizability: Eq. 15 relates B_ξ estimation error and RL optimization error to policy loss under the ideal internal posterior-residual reward-punishment function. Empirically, OHIRL approaches oracle across nonlinear reward-free packet families with online-interleaved learning and an initial stratified coverage phase. The controls separate observation access from reward provision: every method observes the same packet dimensions, while hand-coded packet scores, curiosity, error reduction, shuffled B_ξ, and zero reward trail learned posterior-residual value.

6 Conclusion

We presented an online IRL/RL formulation in which the environment updates perceptual packets p_t while withholding scalar reward. M_ψ learns posterior next-packet expectations; D_ω models residual evolution; $\mathcal{C}_\eta$ evaluates completed post-transition residual trajectories into internal value evidence; B_ξ learns this reward-punishment evidence model; and RL uses learned B_ξ estimates as internal rewards or action scores according to the audited timing protocol. The hidden-reward CartPole/Taxi controls, public-context no-leakage suite, module-role ablation, and posterior-residual mechanism diagnostics show OHIRL learning contextual policies under this event-stream interface, with direct observation scores and reward-like shortcuts trailing or collapsing. The module-role audit supports the functional separation of D_ω dynamics prediction from $\mathcal{C}_\eta$ value-evidence evaluation, while distributional heads represent competing or coexisting positive and negative outcomes. The formal conditions identify when posterior-residual value learning controls policy suboptimality through B_ξ estimation error and RL optimization error.

7 Reproducibility

The artifact contains scripts, CSV/JSON outputs, reports, and audits for the hidden-reward controls, public-context no-leakage suite, module-role and coefficient-origin audits, packet-family diagnostics, reward-free PPO, public-digit counterexample, and corrected distributional B_ξ experiments. `RESULT_SOURCE_CHECK.md` maps reported results to source files.

References

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019. Exploration by Random Network Distillation. In *ICLR*.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *arXiv:1606.01540*.

Chevalier-Boisvert, M.; Dai, B.; Towers, M.; Perez-Vicente, R.; Willems, L.; Lahlou, S.; Pal, S.; Castro, P. S.; and Terry, J. 2023. MiniGrid and MiniWorld: Modular Reinforcement Learning Environments. *CoRR*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2019. Diversity Is All You Need: Learning Skills without a Reward Function. In *ICLR*.

Friston, K. 2010. The Free-Energy Principle: A Unified Brain Theory? *Nature Reviews Neuroscience*, 11(2): 127–138.

Friston, K.; FitzGerald, T.; Rigoli, F.; Schwartenbeck, P.; and Pezzulo, G. 2017. Active Inference: A Process Theory. *Neural Computation*, 29(1): 1–49.

Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. 2023. Mastering Diverse Domains through World Models. *arXiv:2301.04104*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *ICML*.

Henaff, M.; Raileanu, R.; Jiang, M.; and Rocktaschel, T. 2022. Exploration via Elliptical Episodic Bonuses. In *NeurIPS*.

Juechems, K.; and Summerfield, C. 2019. Where Does Value Come From? *Trends in Cognitive Sciences*, 23(10): 836–850.

Keramati, M.; and Gutkin, B. 2011. A Reinforcement Learning Theory for Homeostatic Regulation. *Psychological Review*, 118(4): 604–644.

Laskin, M.; Yarats, D.; Liu, H.; Lee, K.; Zhan, A.; Lu, K.; Cang, C.; Pinto, L.; and Abbeel, P. 2021. URLB: Unsupervised Reinforcement Learning Benchmark. In *NeurIPS Datasets and Benchmarks*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-Driven Exploration by Self-Supervised Prediction. In *ICML Workshop*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv:1707.06347*.

Sekar, R.; Rybkin, O.; Daniilidis, K.; Abbeel, P.; Hafner, D.; and Pathak, D. 2020. Planning to Explore via Self-Supervised World Models. In *ICML*.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A Physics Engine for Model-Based Control. In *IROS*.

Abbeel, P.; and Ng, A. Y. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *ICML*.

Bellemare, M. G.; Srinivasan, S.; Ostrovski, G.; Schaul, T.; Saxton, D.; and Munos, R. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. In *NeurIPS*.

Bhaskara, V.; and Wang, H. 2026. Curiosity-Critic: Cumulative Prediction Error Improvement as a Tractable Intrinsic Reward for World Model Training. *arXiv:2604.18701*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *NeurIPS*.

Du, Y.; Winnicki, A.; Dalal, G.; Mannor, S.; and Srikant, R. 2024. Exploration-Driven Policy Optimization in RLHF: Theoretical Insights on Efficient Data Utilization. *arXiv:2402.10342*.

Hou, Z.; An, Z.; and Du, W. 2025. Beyond Noisy-TVs: Noise-Robust Exploration via Learning Progress Monitoring. *arXiv:2509.25438*.

Kearns, M.; and Singh, S. 2002. Near-Optimal Reinforcement Learning in Polynomial Time. *Machine Learning*, 49: 209–232.

Jin, C.; Krishnamurthy, A.; Simchowitz, M.; and Yu, T. 2020. Reward-Free Exploration for Reinforcement Learning. In *ICML*.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping. In *ICML*.

Oudeyer, P.-Y.; and Kaplan, F. 2007. What Is Intrinsic Motivation? A Typology of Computational Approaches. *Frontiers in Neurorobotics*, 1: 6.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

Schmidhuber, J. 1991. A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers. In *Proc. SAB*.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Tang, H.; Houthooft, R.; Foote, D.; Stooke, A.; Chen, X.; Duan, Y.; Schulman, J.; De Turck, F.; and Abbeel, P. 2017. Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In *NeurIPS*.

Tinker, T. J.; Doya, K.; and Tani, J. 2024. Intrinsic Rewards for Exploration without Harm from Observational Noise: A Simulation Study Based on the Free Energy Principle. *arXiv:2405.07473*.

Wagenmaker, A.; Chen, Y.; Simchowitz, M.; Du, S. S.; and Jamieson, K. 2022. Reward-Free RL is No Harder Than Reward-Aware RL in Linear Markov Decision Processes. In *COLT*.

Yuan, M.; Castanyer, R. C.; Li, B.; Jin, X.; Berseth, G.; and Zeng, W. 2024. RLeXplore: Accelerating Research in Intrinsically-Motivated Reinforcement Learning. *arXiv:2405.19548*.

Ziebart, B. D.; Maas, A.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum Entropy Inverse Reinforcement Learning. In *AAAI*.

Reproducibility Checklist

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) yes

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) yes

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) yes

2.4. Proofs of all novel claims are included (yes/partial/no) yes

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) yes

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) yes

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) NA

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) yes

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) NA

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) NA

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) yes

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) NA

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) yes

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) yes

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes