DISCOVERING HIERARCHICAL OPPONENT STRATEGIES VIA STRUCTURAL ENTROPY MINIMIZATION IN IMPERFECT-INFORMATION GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

In imperfect-information games, an opponent behaves at more than one level, from immediate reactions to strategies that persist over time, but these levels are hidden and the active strategy can change at any time. Most opponent models compress the interaction history into one flat latent state, while methods that use hierarchy require it to be specified in advance or rely on a large language model. We present Structural Entropy-based Opponent Modeling (SEOM), which builds an online interaction graph, finds a minimum-code-length encoding tree by structural entropy minimization, and selects the tree depth from the data. A context-conditional detector compares action distributions within decision contexts, so it responds to genuine strategy changes and stays silent as the learner's own visits shift; a structural compression ratio, computed before training, indicates whether hierarchy will help. On a benchmark of opponents with planted hierarchies, SEOM recovers every planted depth and partition, with level agreement approaching one as the stream grows, and turns this recovery into gains over flat and structured baselines in return, log-loss, sample efficiency, and post-switch adaptation. The detector identifies genuine switches while suppressing false alarms under learner drift, and the compression ratio chooses between hierarchical and flat modeling with no cross-validation errors.

1 INTRODUCTION

In an imperfect-information game, an agent chooses actions while the opponent's private information, latent intent, and current policy are unobserved. The hidden state can be a private card, an undisclosed objective, or an internal behavioral mode. *Opponent modeling*, the task of predicting the adversary from the interaction stream, is central to multi-agent reinforcement learning (MARL) and game theory (Albrecht & Stone, 2018; Nashed & Zilberstein, 2022).

This setting creates two linked difficulties. The first is non-stationarity: the opponent may learn, switch strategies, or deceive. A long history then mixes outdated evidence with current behavior after a switch, while a short window forgets a strategy that later returns. The second is attribution. When both sides adapt, the situations the learner reaches change as its own policy evolves, so a shift in the marginal action distribution may reflect either a genuine opponent switch or the learner's own behavior, although the two call for opposite responses.

Existing methods do not address both. Bayesian and recursive models reason over hand-specified candidate types and do not scale beyond small games. Neural opponent models compress the history into one flat latent state (Hausknecht & Stone, 2015; Yu et al., 2022; Yang et al., 2025), which cannot separate a stable strategy-level trait from a context-specific reaction and gives no interpretable decomposition. Structured methods move toward hierarchy but impose it from outside: hierarchical opponents fix the levels in advance (Wang et al., 2025; Huang et al., 2024), LLM-based agents build a causal account by reflection at a cost a standard policy cannot afford (Cao et al., 2026), and switch-detection or policy-reuse methods assume piecewise stationarity or a known policy library and track an aggregate marginal confounded by the learner's own visits (Hernandez-Leal et al., 2017; DiGiovanni & Tewari, 2021; Zheng et al., 2018). A way to discover the hierarchy from the stream under an information-theoretic criterion is missing.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The hierarchy is already present in the relational structure of the interaction: concrete state-action reactions at the bottom, recurring modes in the middle, and a few coarse strategy types at the top. Structural information principles (Li & Pan, 2016) identify the essential structure with the encoding tree that minimizes *structural entropy*, the expected code length of dynamics on the graph, which partitions vertices into nested groups dense inside and sparse across boundaries. The graph is a suitable substrate because its edges are local co-occurrence counts, updated without reprocessing history and without a fixed history length.

We build SEOM on this observation. A heterogeneous graph of the opponent's observations, actions, and public beliefs is minimized by level-synchronous agglomeration into a tree, entropy-weighted aggregation converts the tree into a multi-scale embedding, and a coarse-to-fine module reasons from strategy to action; the model is trained end to end with PPO (Schulman et al., 2017) and an auxiliary opponent-action loss. Because the objective is a code length, it is a minimum-description-length criterion that penalizes unnecessary granularity, and the depth, together with the compression ratio $\kappa^\star$, is selected from data by a marginal-lift rule.

Maintaining the hierarchy online raises a second question: a change in observed behavior need not mean the strategy changed, since the marginal $P(a)$ moves when the learner reaches new situations even against a fixed rule. The detector therefore compares, within clusters of decision context, the recent conditional action distribution with an established reference; it is invariant to visit frequency (Proposition 3) and carries a finite-sample false-alarm guarantee (Proposition 4), so the tree is rebuilt only after a genuine switch.

Evaluating these claims requires ground truth that standard benchmarks lack, since a plausible hierarchy could be coincidental and a gain could have unrelated causes. We therefore construct opponents with planted, then hidden, hierarchies and known switch times, and find that SEOM recovers the planted levels, improves the learner across several dimensions, detects genuine switches while remaining silent under its own drift, and predicts from a statistic computed before training where hierarchy helps.

Our contributions are as follows.

- **Hierarchical opponent modeling by structural entropy minimization.** To our knowledge, we are the first to use structural-entropy minimization to *discover* an opponent's hierarchical abstraction from an online graph, with no preset hierarchy or language model; the levels and their depth are selected from data.
- **Online maintenance and a pre-training applicability criterion.** We show why a marginal change test confounds an opponent switch with the learner's own drift, give a context-conditional detector invariant to visits with a finite-sample false-alarm guarantee, and derive a pre-training compression ratio $\kappa^\star$ that predicts whether a hierarchical model improves on a flat one.
- **A controlled study with ground-truth levels.** Across controlled configurations and four domain instantiations, with flat and shuffled controls, the study establishes recovery, multi-dimensional gains, detector invariance, $\kappa^\star$ prediction, and the cost boundaries.

2 Related Work

Opponent modeling in imperfect-information games. Multi-agent decisions are commonly modeled as (partially observable) stochastic games (Littman, 1994; Lanctot et al., 2019). One line computes near-equilibrium, worst-case-robust strategies, including counterfactual regret minimization and its deep forms (Zinkevich et al., 2007; Brown et al., 2019; 2020) and the poker systems DeepStack and Libratus (Moravčík et al., 2017; Brown & Sandholm, 2018); these assume known rules and optimize one fixed strategy rather than a changing adversary. A complementary data-driven line maintains beliefs about the particular opponent (Albrecht & Stone, 2018; Nashed & Zilberstein, 2022), with neural members that encode the history in one flat latent state (Hausknecht & Stone, 2015; Raileanu et al., 2018; Rabinowitz et al., 2018; Yu et al., 2022; Yang et al., 2025; Al-Shedivat et al., 2018). SEOM replaces that flat latent with a hierarchy it discovers. The closest structured methods fix an intent-strategy hierarchy in advance (Wang et al., 2025; Huang et al., 2024) or build a causal model through LLM reflection (Cao et al., 2026); SEOM instead induces the hierarchy under an information-theoretic criterion, with no language model and no preset levels.

Hierarchical abstraction and structural information. Hierarchical abstraction in single-agent RL includes options (Sutton et al., 1999), state-abstraction theory (Li et al., 2006), and the information bottleneck and its deep form (Tishby et al., 1999; Alemi et al., 2017). Those levels are defined by hand or by reward reachability; ours are defined by code length on the interaction graph. Structural entropy (Li & Pan, 2016) instead defines a hierarchical code of dynamics on a graph through an optimal encoding tree (surveyed in Wu et al., 2025), used for state abstraction (Zeng et al., 2023b; Peng et al., 2025), exploration (Peng et al., 2024), role discovery (Zeng et al., 2023a), and cooperative goals (Amato et al., 2026). Unlike modularity (Newman & Girvan, 2004; Fortunato & Barthélemy, 2007), it returns a nested, MDL-optimal hierarchy (Rissanen, 1978) rather than one scale-dependent flat partition. These works abstract an agent's own state space or cooperative teammates; SEOM models a hidden, non-stationary adversary online and asks when a hierarchy should outperform a flat abstraction, which $\kappa^\star$ answers.

Non-stationarity and switch detection. Co-adaptation makes the learner's environment non-stationary (Hernandez-Leal et al., 2019). Drift tests such as CUSUM (Page, 1954) and Bayesian online change-point detection (Adams & MacKay, 2007; Gama et al., 2014), hidden-mode adaptation (Al-Shedivat et al., 2018; Zintgraf et al., 2021; Fu et al., 2022; Wu et al., 2022), and policy-reuse or piecewise-stationary agents (Hernandez-Leal et al., 2017; DiGiovanni & Tewari, 2021; Zheng et al., 2018) track an aggregate marginal or assume a known policy library, and co-adaptation confounds that marginal with the learner's visits; Proposition 3 shows the conditional test is visit-invariant.

3 Preliminaries

Partially observable stochastic game. We model interaction as a partially observable stochastic game (POSG) (Littman, 1994; Lanctot et al., 2019) $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}, P, \{R_i\}, \gamma, \{\mathcal{O}_i\}\rangle$. From the self agent i's view, the opponent $-i$ chooses actions under a latent regime z_t that is never observed:

$$z_t \sim P_z(\cdot \mid z_{t-1}), \qquad a_t^{-i} \sim \pi_{z_t}(\cdot \mid \tau_t^{-i}), \tag{1}$$

where regimes need not be labeled or permanent and may recur. The learner builds an opponent model $\hat{\pi}_{-i}(a^{-i} \mid \tau_i)$ from its own action-observation history. Such a model is assessed by action-prediction accuracy and log-loss, by the adaptation delay after a switch, and by the return of the policy that uses it.

Structural entropy. For a weighted undirected graph $G = (V, E, W)$, write $d_v = \sum_u W(u, v)$ and $\mathrm{vol}(G) = \sum_v d_v$. The one-dimensional structural entropy is

$$H^1(G) = -\sum_{v \in V} \frac{d_v}{\mathrm{vol}(G)} \log_2 \frac{d_v}{\mathrm{vol}(G)}. \tag{2}$$

An *encoding tree* T is a rooted tree whose root λ is V, whose leaves are singletons, and whose internal node α spans a vertex subset V_α with volume $\mathcal{V}_\alpha = \sum_{v \in V_\alpha} d_v$ and boundary cut $g_\alpha = \sum_{u \notin V_\alpha, v \in V_\alpha} W(u, v)$. The structural entropy *under* T is

$$H^T(G) = -\sum_{\alpha \in T,\, \alpha \neq \lambda} \frac{g_{\alpha^-}}{\mathrm{vol}(G)} \frac{\mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}} \log_2 \frac{\mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}}, \tag{3}$$

where α^- is the parent of α (with $g_\lambda = \mathrm{vol}(G)$). The K-dimensional structural entropy is $H^K(G) = \min_{T:\, \mathrm{height}(T) \leq K} H^T(G)$; its minimizing tree reveals nested communities. Its telescoping form guarantees $H^K \leq H^{K-1} \leq H^1$, an invariant we verify numerically at every checkpoint. Structural entropy extends to directed graphs, but we symmetrize on read as $W = (A + A^\top)/2$ and remove self-loops so these identities hold.

4 Method: Structural Entropy-based Opponent Modeling

We use the following notation. $a_t^{\mathrm{op}} = a_t^{-i}$ is the opponent action, a_t^{self} the learner's, o_t the opponent observation, s_t the learner's state, A the adjacency (the action set is $\mathcal{A}$), and c a decision context; the root embedding z_{op} stands in for the latent regime z_t. Components below are presented in data-flow order, followed by the analysis.

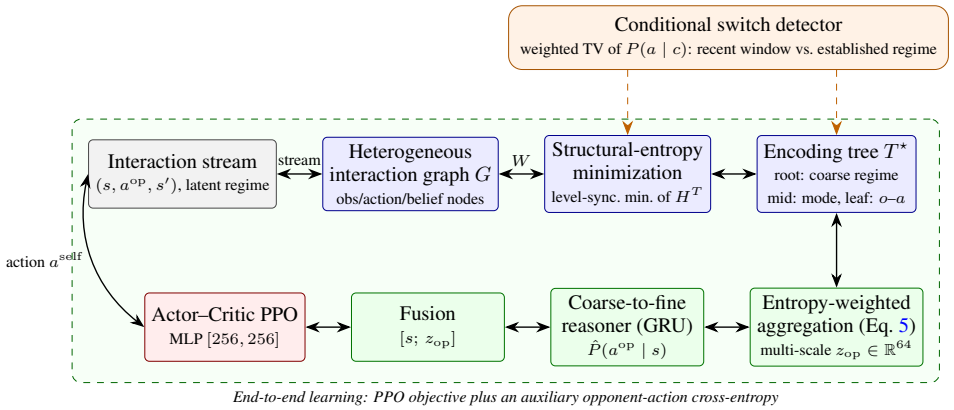

Figure 1: Overview of SEOM. Online interactions form an opponent interaction graph; structural-entropy minimization discovers the encoding tree, whose levels read as coarse regimes, mid-level modes, and concrete observation-action pairs; entropy-weighted aggregation and coarse-to-fine reasoning produce the opponent embedding and action prediction; the conditional switch detector triggers re-encoding only on a genuine strategy change; PPO consumes the root embedding and an auxiliary opponent-action loss trains the model end to end. The orange boxes form the detector path and the other boxes the learning path.

4.1 Opponent interaction graph

Each opponent decision produces a record $(o_t, a_t^{\mathrm{op}}, o_{t+1}, \hat{b}_t)$, where $\hat{b}_t$ is a public-belief estimate when available. The graph is heterogeneous, directed, and weighted, with *observation prototypes*, *action nodes* (one per discrete opponent action), and optional *belief nodes*. A new observation merges into its nearest prototype at cosine similarity at least 0.8 and otherwise becomes a prototype, up to 500. Prototypes bound the graph size and turn continuous observations into vertices with stable counts, as the entropy definition requires. Edges connect $o_t \to a_t^{\mathrm{op}}$ and $a_t^{\mathrm{op}} \to o_{t+1}$ with unit weight and, with a belief, $o_t \to \hat{b}_t \to a_t^{\mathrm{op}}$ with weight 0.5. Edge masses follow an EMA of decay $\rho = 0.95$ and edges below 10^{-4} are pruned. On read we symmetrize and remove self-loops, $W = (A + A^\top)/2$ and $\mathrm{diag}(W) = 0$: sparse non-stationary sampling makes directed cuts noisy, and symmetrization gives one well-defined code length.

4.2 Hierarchical structure discovery by structural-entropy minimization

Starting from the degenerate two-level tree of entropy H^1, we approximate structural-entropy minimization by bottom-up, level-synchronous agglomeration, a standard constructive heuristic, since exact minimization is combinatorial. At each level we merge the same-height pair α, β into γ that most decreases Eq. equation 3, using an $O(1)$ update ($g_\gamma = g_\alpha + g_\beta - w_{\alpha\beta}$), and stop at height at most $K = 4$ or tolerance 10^{-6}. Every accepted merge reduces entropy, so $H^{T^\star} \le H^1$ exactly; $H^{T^\star}$ upper-bounds $H^K(G)$, and the incremental and full computations agree to below 10^{-15}.

Depth selection. The agglomeration produces a candidate tree at each height $1, \dots, K$. We retain a deeper tree only when each added level removes enough entropy: the marginal lift of a level is its entropy reduction relative to that of the preceding level. The selected tree is the deepest tree whose lifts exceed a cutoff fixed before the study; denote it by $T^\star$ and its depth-selected compression ratio by

$$\kappa^\star = \frac{H^1 - H^{T^\star}}{H^1}, \tag{4}$$

the fraction of one-level structural entropy removed by the selected tree. This is the same minimum-description-length logic that governs individual merges, applied to depth: levels that add code without removing comparable uncertainty are discarded. The tree rebuilds every 200 opponent decisions or immediately on a detector alarm; root communities group broadly similar contexts at the strategy level, while leaves retain specific ones. Structural entropy is chosen over alternatives for the

same reasons: a flat latent has no levels, modularity returns one scale-dependent partition, and an information bottleneck needs an unavailable relevance variable (Proposition 1).

4.3 Entropy-weighted hierarchical representation

Index leaves by ν and internal nodes by α; the leaf feature x_ν concatenates the prototype vector with normalized incident-edge statistics. A shared encoder f_{enc} (two 128-unit layers, LayerNorm, dropout 0.1) maps x_ν to $h_\nu \in \mathbb{R}^{64}$. For an internal node α with children $\{\alpha^{\langle i \rangle}\}_{i=1}^{L_\alpha}$, we aggregate bottom-up, using each child's *assigned structural entropy* as an attention logit:

$$h_\alpha = \sum_{i=1}^{L_\alpha} \frac{\exp\left(H^{T^\star}(G; \alpha^{\langle i \rangle})/\tau_{\text{att}}\right)}{\sum_j \exp\left(H^{T^\star}(G; \alpha^{\langle j \rangle})/\tau_{\text{att}}\right)} h_{\alpha^{\langle i \rangle}}, \qquad \tau_{\text{att}} = 0.5. \tag{5}$$

Children with larger assigned entropy carry more uncertainty and therefore more weight; $\tau_{\text{att}} \to \infty$ recovers uniform averaging (the *uniform* ablation). The root $z_{\text{op}} = h_\lambda \in \mathbb{R}^{64}$ summarizes the opponent coarsely, depends only on the tree and leaves, and is cached between rebuilds; the *flat* ablation uses the unweighted mean of descendant leaves.

4.4 Coarse-to-fine hierarchical reasoning

A reasoning module traverses the tree from root to leaves: at each level a small network (one 64-unit hidden layer) combines a projection of the current observation, that level's attended representation, and the parent belief, and passes a refined belief to its children; a GRU (Chung et al., 2014) carries belief across decisions. The action head outputs $\hat{P}(a_t^{\text{op}} \mid s_t) = \text{softmax}(\cdot)$, so coarse beliefs condition finer inference. It is supervised by the observed opponent action through $\mathcal{L}_{\text{opp}} = \text{NLL}(\hat{P}, a^{\text{op}})$ on the last 512 interactions, with two minibatch steps per PPO update to match the flat baselines' gradient rate.

4.5 Conditional regime-switch detection

A marginal detector is confounded under co-adaptation, since $P(a)$ moves with the learner's visits even against a fixed-rule opponent. The *strategy* is $P(a \mid c)$, monitored with at most 24 context prototypes at cosine 0.85, kept separate from the graph's observation prototypes (cosine 0.8, cap 500) so a rebuild cannot contaminate the statistic; the coarser codebook also stabilizes estimates. Every 12 decisions, over a 96-decision window, let $n_c(a)$ be the count of action a in context c, $n_c = \sum_a n_c(a)$, and m_c the matching reference count. We compare the window law $q_c(a) = n_c(a)/n_c$ with a reference law $r_c(a)$ accumulated during the established regime (at most 800 samples), keeping contexts with $n_c \geq n_{\min} = 6$. The statistic is the count-weighted mean total variation

$$d_t = \sum_{c:\, n_c \geq n_{\min},\, m_c \geq n_{\min}} \frac{n_c}{\sum_{c'} n_{c'}} \text{TV}(q_c, r_c), \qquad \text{TV}(q, r) = \tfrac{1}{2} \sum_a |q(a) - r(a)|. \tag{6}$$

Count weighting makes d_t visit-invariant (Proposition 3). Once the reference is mature, a switch is declared at $d_t \geq \eta$ ($\eta = 0.25$) and the reference is rebuilt; supervision history is kept. Regimes recur, and discarding data after every alarm, some false, would starve the reasoner, so an alarm triggers re-encoding rather than a reset. The *noSwitch* ablation computes d_t but never acts on it.

4.6 Joint training with the learner

The learner fuses its own state s_t with the root embedding z_{op} and feeds the result to actor and critic MLPs, each with two hidden layers of 256 units. The full model is trained with clipped PPO, with the opponent-action loss added to the policy objective, $\mathcal{L} = \mathcal{L}_{\text{PPO}} + \mathcal{L}_{\text{opp}}$, so action prediction and policy improvement are optimized together.

The complete online procedure, including prototype maintenance, the detector vote, depth selection, and the optimization steps, is stated as Algorithm 1 in Appendix B.

4.7 Analysis

The four results below correspond to the structural ingredients, and two remarks follow; proofs are in Appendix B.

Proposition 1 (Hierarchical code length / MDL). *$H^T(G)$ equals the expected number of bits used by the hierarchical code that, whenever dynamics enter a node's subset from outside it, emits that node's identifier conditional on its parent. Hence $\arg\min_{T:\,\mathrm{height}(T)\le K} H^T$ is the minimum-description-length hierarchy of height at most K, and H^1 is the cost of the degenerate (flat) tree.*

Proposition 2 (Nested compression). *On the symmetrized graph, $H^K(G) \le H^{K-1}(G) \le H^1(G)$.*

Proposition 3 (Conditional statistic ignores visit drift). *If the opponent keeps a fixed conditional law $\pi(a \mid c)$ while only the context distribution $\mu(c)$ changes, the marginal $m(a) = \sum_c \mu(c)\pi(a \mid c)$ can move ($\mathrm{TV}(m, m') > 0$) although no conditional law does; the weighted statistic of Eq. equation 6 then converges to 0 as context counts grow and is positive only when a conditional law itself changes.*

Proposition 4 (Finite-sample false-alarm control). *Fix the opponent's conditional law within a regime; for a mature context c let the n_c window decisions be approximately i.i.d. from $p_c = \pi(\cdot \mid c)$ while the reference has converged to p_c, and write $k = |\mathcal{A}|$. Then $\mathbb{E}[\mathrm{TV}(q_c, p_c)] \le \sqrt{(k-1)/(2n_c)}$, and Hoeffding's inequality with a union bound over actions gives the same order with high probability, $\mathrm{TV}(q_c, p_c) \le \frac{k}{2}\sqrt{\ln(2k/\delta)/(2n_c)}$. Because Eq. equation 6 averages only mature contexts with normalized weights $w_c = n_c / \sum_{c'} n_{c'}$, a fixed law gives $\mathbb{E}[d_t] \le \sqrt{(k-1)/(2n_{\min})}$. Choosing η well above this bound controls false alarms, while a genuine change of weighted magnitude above η is flagged.*

Remark 1 (range of $\kappa^\star$). Proposition 2 gives $0 \le \kappa^\star \le 1$, with $\kappa^\star = 0$ exactly when the selected tree removes no uncertainty beyond H^1 (and likewise for the greedy tree, since $0 \le H^{T^\star} \le H^1$). *Remark 2 (amortized cost).* Graph updates are amortized $O(1)$ under the prototype cap; a rebuild every 200 decisions or on an alarm costs $O(KV)$ with $K \le 4$, and the observation-independent root is cached. Structural cost is independent of history length, consistent with the multipliers of Sec. 5.6.

5 Experiments

The experiments answer five questions in order: whether the discovered hierarchy is real, whether it helps the learner, whether the detector identifies genuine switches, whether gains can be predicted before training, and where the compute cost and failure boundaries lie.

5.1 Setup: opponents, compared methods, and reporting

Opponents are generated with a known hierarchy that is then hidden; the construction is a degree-corrected hierarchical block model, a standard planted-partition construction. It places a planted tree T_{pl} of depth D and branching vector B over decision contexts: same-leaf contexts draw dense edges, blocks under one parent draw intermediate edges, and blocks under different families draw sparse edges, with degree correction so hubs are not mistaken for communities. Each leaf carries a context-conditioned action distribution; a latent regime selects a subtree of actions and switches on a known schedule with mean dwell L. The benchmark contains controlled configurations that sweep $D \in \{1, 2, 3, 4\}$, branching, divergence, dwell, and length from 200 to 16,000, plus four domain instantiations, **Leduc Hold'em**, **Adversarial Grid-World (AGW)**, **Sharks-Minnows (SaM)**, and **Predator-Prey** (Appendix D), whose reward scales calibrate generated returns. Predator-Prey uses only flat opponents ($D = 1$) as a negative control. Two nulls guard interpretation: a *flat* opponent with one block, and a degree-preserving *edge shuffle* that keeps the degree sequence but destroys block structure; the full specification is in Appendix A. We compare SEOM with five learned methods that share one PPO optimizer and hyper-parameters: **PPO** (no opponent model), **DRQN** (recurrent encoding), **MBOM** (forward predictor), **UAOM** (uncertainty ensemble), and **SOM-adapted** (the construction/prediction separation of Cao et al., 2026 with a learned constructor). Three ablations modify SEOM: **uniform** replaces the entropy weights with equal weights, **flat** replaces the tree with one flat mean, and **noSwitch** never acts on a detector alarm. Detector comparisons add **BOCPD** (Adams & MacKay, 2007), **CUSUM** (Page, 1954), and a *marginal-law* detector. Every

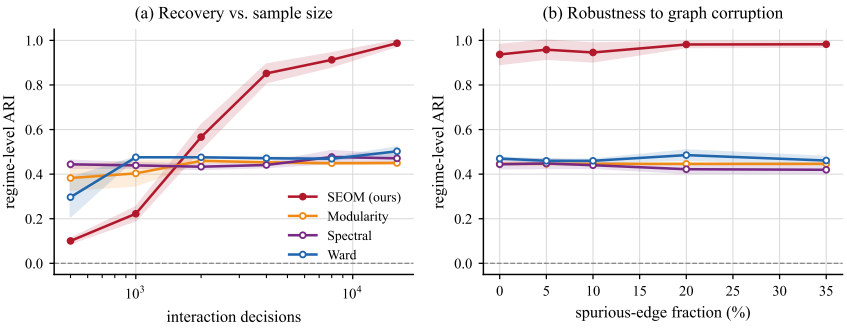

Figure 2: Hierarchy recovery on planted opponents. (a) Level-wise adjusted Rand index versus interaction length; (b) robustness to planted divergence; (c) depth-selection confusion matrix (planted depth by row, selected depth by column); (d) online recovery from a cold start. Dashed traces are flat clustering baselines; shaded bands are 95% bootstrap intervals over 20 seeds.

cell uses 20 independent seeds. We report means with seed standard deviations and 95% bootstrap intervals for correlations. When the intervals of two methods overlap, we report the comparison as a tie rather than claim an improvement, so a difference that sampling noise could explain is not counted as evidence. An edge is assigned to a block only past a configuration-model gate with a three-standard-deviation margin, which scores recovery against partitions that chance cannot produce; hyper-parameters (Appendix C) are fixed once. Each learner is evaluated on exploration-free episodes at checkpoints that span several switches, and evaluation seeds are disjoint from training. All learners receive the same number of gradient updates, and the auxiliary loss uses the same minibatch count as the flat methods, so no compared method receives more optimization.

5.2 Does SEOM recover the planted hierarchy?

We first run SEOM on planted opponents whose true tree is hidden. Figure 2 shows recovery as the stream grows: level-wise agreement, measured by the adjusted Rand index (pair-level agreement, at 0 for chance and 1 for identical partitions), rises from about 0.10 at 500 decisions to about 0.99 at 16,000, whereas modularity, spectral, and Ward clustering plateau at 0.4–0.5 in flat partitions. Such a score means the flat methods place some leaves with the right neighbors but merge every family into one block, and tuning cannot add a level the partition does not have. The flat methods stop early for a structural reason: a one-level partition has no place for blocks nested inside blocks, so it captures either the families or the leaves but not both. Agreement holds across planted divergence, including hard three-family cases, and the online variant reaches the same point from a cold start, from about 0.07 at 200 decisions to about 0.98 at 16,000, so no offline pass is needed; the graph is updated in place and no raw history is stored. Pooling all 60 configurations, selected depth equals planted depth in every cell, so the criterion also answers how many levels exist, without labels or a validation set. The criterion also withholds levels where none exist. Flat opponents give a one-level tree with $\kappa^\star = 0$; edge-shuffled graphs sometimes admit a second level but keep $\kappa^\star$ near 0.6%, below the recommendation threshold of Sec. 5.5, while planted three-level opponents give depth three with $\kappa^\star$ near 4.7%. On the branching controls agreement ranges from 0.82 to 0.95; at four branches a flat modularity partition reaches 0.88 and can match one level, though it returns no depth. Recovery is unreliable below roughly 500 decisions or at very low divergence, the point at which local co-occurrence counts have not stabilized, so these cells reflect insufficient evidence rather than bias. Within this operating range, SEOM thus recovers the planted hierarchy level by level and depth by depth, while flat and shuffled streams remain at one level.

5.3 Does the recovered hierarchy help?

On the structured domains SEOM leads or ties every compared method on stationary and switching return and achieves the lowest log-loss, meaning it assigns more probability to the action the opponent actually takes, while all methods tie on the flat Predator-Prey control (Figure 11, Appendix D); transfer to held-out regimes and probability calibration improve as well. It also reaches a fixed score

Table 1: Tree-intervention gains by true depth.

D	Oracle	Discovered	Shuffled	Random	Flat
1	0.00	0.00	0.00	0.00	0.00
2	1.84	1.71	0.29	0.06	-0.02
3	2.07	1.92	0.34	0.09	-0.01

threshold with about 23% fewer decisions than PPO on Leduc and about 26% fewer on AGW and SaM. The saving follows from what the tree supplies: conditioning on the level above narrows the set of reactions the learner has to fit, so each level is learned with less data. The tie on Predator-Prey is informative in the same way, since hierarchy should not help where no hierarchy exists.

To check that these gains come from the tree itself rather than from added capacity, Table 1 replaces the discovered tree with five alternatives while holding the model fixed. Oracle and discovered trees nearly tie and far exceed shuffled and random trees; flat and random variants are near zero, and every variant is zero at depth one. The dose-response ladder in Appendix D tells the same story, with gains of zero at depth one that increase monotonically with $\kappa^\star$ and standardized effect sizes of about 2–4 across return, success, log-loss, transfer, and calibration; post-switch adaptation is fastest under abrupt and gradual switches, while the noSwitch ablation retains a permanent deficit. A switch changes only the subtree the regime selects, while the upper levels stay valid and are reused, so the learner relearns one subtree rather than the whole model; adaptation curves in Appendix D show the hierarchy keeps most of its score through a switch. An effect size of 2 to 4 means the gap to the best baseline spans two to four standard deviations across seeds, so the ordering does not depend on favorable seeds. Hierarchy adds nothing on near-ceiling tasks and flat opponents and can slightly hurt under very fast switching, both visible in advance through $\kappa^\star$ (Sec. 5.5). Where the benchmark predicts a hierarchy, the recovered tree therefore improves the learner on every measured dimension, and the improvement is tied to the correct tree rather than to added capacity.

5.4 Does the detector identify genuine switches?

A marginal test cannot separate a genuine switch from the learner's own visits, because $P(a)$ moves with visits even against a fixed rule, whereas comparing action distributions within context prototypes removes the visit term and registers only a change in $P(a \mid c)$ (Proposition 3). An alarm is matched to a true switch within a fixed tolerance window, and an unmatched alarm counts as false. Scoring alarms against true events (Figure 13 and Table 4, Appendix D), the conditional detector reaches AUC-PR of about 0.84 against 0.52–0.72 for BOCPD, CUSUM, and a marginal test, keeps stationary false alarms near 0.6 per $1{,}000$ decisions against 3.4–9.8, and retains competitive recall at 0.94 with median lag near 56 decisions, comparable to the faster baselines. All detectors share one window and schedule, so only the statistic differs.

On rare events recall can always be bought by alarming more often, so the relevant advantage is false-alarm control at fixed recall, which is why precision rather than recall separates the methods; at its operating point the conditional detector finds most switches while issuing fewer than one false alarm per thousand decisions.

The timing tests in Appendix D support this reading: under learner drift the detector raises about 0.8 alarms per $10{,}000$ decisions against 26 for the marginal test, it rejects 96.5% of short blips because a brief excursion does not move a prototype-conditional distribution past the gate, and alarms timed by the learned detector capture 90% of oracle-timed gain against 27% for random alarms. Gradual drift lengthens lag to about 99 decisions and slowly maturing contexts do the same, both known operating limits rather than missed events; the detector is tuned for the regime in which switches are discrete events, the setting of this paper. Subject to these stated limits, the conditional detector therefore responds to genuine switches and remains silent as the learner's own visits change.

5.5 Can gains be predicted before training?

From the first $2{,}000$ decisions of each configuration we build a probe graph, compute the depth-selected compression ratio, and compare it with the realized hierarchical-minus-flat gain; rank correlation is used because the task is ordinal, ordering configurations by gain rather than fitting a linear

scale. Figure 12 (Appendix D) reports the results. Compression correlates with gain at Spearman $\rho \approx 0.72$ (interval $[0.51, 0.86]$), against $0.68/0.71$ for fixed-depth ratios, 0.26 for modularity, and weak or negative values for entropy and spectral gap, while prototype count carries no information, since the count reflects the size of the observation space rather than how the blocks nest. A single threshold near 1.4%, chosen on training folds, recommends hierarchy in 40 configurations and flat in 20 with no leave-one-configuration-out errors, against 85% accuracy for modularity. The leave-one-configuration-out test holds out an entire domain at a time, so the perfect score includes domains unseen when the threshold was chosen. In plain terms, the threshold separates streams whose co-occurrence admits nested blocks from streams that one level already describes; realized gains follow the same ordering, from zero in the flat control to about 2.6 and 2.9 standardized units at depths two and three.

The statistic transfers across domains because it measures structure rather than switch rate: it stays near 3.5% as dwell varies from 100 to $1{,}600$ and moves only with depth and divergence, and a probe graph agrees with the full graph at $r \approx 0.82$; the early probe works because block membership appears in co-occurrence before action-level details do. The competing diagnostics fail for identifiable reasons, as modularity returns one flat partition and misses nested structure while entropy and spectral gap correlate weakly or in the wrong direction. With 60 configurations the relationship is moderately strong but imprecise (lower bound 0.51); we pre-register that a lower bound below 0.3 would reduce the construct to an exploratory diagnostic. Across the 60 configurations, a compression ratio computed before training therefore identifies where hierarchy will beat a flat model, and a single threshold transfers across domains.

5.6 Cost and failure boundaries

Wall-clock multipliers over PPO grow from about 1.25 to 10.6 with tree height and prototype count (Figure 14a), including graph maintenance, rebuilds, prototype matching, and the auxiliary loss, all timed on one machine; most of the added time goes to graph and tree rebuilds, which occur only every two hundred decisions or after an alarm; depth two stays below a factor of two at typical prototype counts, while depth four exceeds a factor of ten, so the depth selector also guards against over-building. The net-gain heatmap in Figure 14b states where hierarchy pays off: gains are largest when regimes are distinct and long enough to learn, and turn negative when regimes are weak and short, losing about 0.8 standardized units in the worst cell, because rebuilds then cost more than the structure returns. Each heatmap cell averages twenty seeds and colors give net gain after compute time is subtracted, so a red cell is a configuration in which hierarchy earns its overhead. The rule for using the method follows directly: estimate the gain from $\kappa^\star$, subtract the multiplier, and use hierarchy only when the remainder is positive; all boundary values are tabulated in Appendix D.

6 Conclusion

The guiding principle of this work is that an opponent's hierarchy should be discovered from the same coding criterion used to describe the stream, not assumed in advance. We presented SEOM, an opponent model for imperfect-information games that discovers a hierarchy of strategies from the interaction stream rather than assuming one. An online interaction graph is compressed into an encoding tree by structural entropy minimization, the same coding criterion selects the depth, and a context-conditional detector rebuilds the tree only after genuine switches. In controlled experiments with planted hierarchies, the recovered levels match the planted levels at every depth and translate into gains in return, log-loss, sample efficiency, and adaptation speed over flat and structured alternatives; the detector identifies genuine switches while remaining silent under the learner's own drift, and a compression ratio computed before training selects the better model family without cross-validation errors. The cost and boundary results specify where hierarchy should be used.

Extending these findings to larger environments such as Hanabi, Overcooked, and SMAC, and to real online play, is the natural next step, together with a directed estimator and block-bootstrap corrections.

AI USE STATEMENT

Following the ICLR 2027 policy on LLM use, we disclose our use of generative AI. We used generative AI tools in three parts of the work: to aid and polish the writing, to retrieve and discover related work, and to support research ideation and execution, including brainstorming ideas and hypotheses, discussing the methodology and experimental design, interpreting results, and writing and debugging the implementation. AI-assisted code was tested for correctness, and AI-suggested ideas were checked against the primary literature by the authors. We did not use these tools to draft sections of the paper, to generate the synthetic datasets, or to formulate or prove mathematical claims. The authors take full responsibility for the final content, including every claim, number, and artifact.

REPRODUCIBILITY STATEMENT

The planted-opponent generator, all four domain instantiations, the complete source code, per-seed raw outputs, configuration files, and the scripts that aggregate the runs into every table and figure will be released as anonymized supplementary material, with fixed seeds and a single script that regenerates the full study. The method's hyper-parameters are listed in Appendix C; the four propositions are proved in Appendix B; the structural-entropy invariants and incremental-to-full equivalence are asserted by automated tests; and all reported uncertainty is across 20 independent seeds. Every table is generated from machine-readable outputs, and positive, null, and negative results are all retained.

ETHICS STATEMENT

This work uses only self-contained, generated game environments. It involves no human subjects, no personal or sensitive data, and no externally released dataset, and it poses no foreseeable harm to individuals. The authors see no ethical concern and adhere to the ICLR Code of Ethics.

REFERENCES

Ryan Prescott Adams and David J. C. MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations*, 2018.

Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. arXiv:1612.00410.

Alessandro Amato, Raffaele Galliera, K Brent Venable, and Niranjan Suri. Encoding goals as graphs: Structured objectives for scalable cooperative multi-agent reinforcement learning. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, 2026.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *International Conference on Machine Learning*, pp. 793–802. PMLR, 2019.

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shiyue Cao, Pei Xu, Likun Yang, Lei Cui, Xiaotang Chen, and Kaiqi Huang. Som: Structured opponent modeling for LLM-based agents via structural causal model. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, 2026.

Junyoung Chung, Çağlar Güçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop on Deep Learning*, 2014.

Anthony DiGiovanni and Ambuj Tewari. Thompson sampling for Markov games with piecewise stationary opponent policies. In *International Conference on Artificial Intelligence and Statistics (AISTATS), Proceedings of Machine Learning Research*, volume 161, 2021.

Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

Haobo Fu, Ye Tian, Hongxiang Yu, Weiming Liu, Shuang Wu, Jiechao Xiong, Ying Wen, Kai Li, Junliang Xing, Qiang Fu, et al. Greedy when sure and conservative when uncertain about the opponents. In *International Conference on Machine Learning*, pp. 6829–6848. PMLR, 2022.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014.

Matthew Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable MDPS. In *2015 AAAI Fall Symposium Series*, 2015.

Pablo Hernandez-Leal, Yusen Zhan, Matthew E. Taylor, L. Enrique Sucar, and Enrique Munoz de Cote. Efficiently detecting switches against non-stationary opponents. *Autonomous Agents and Multi-Agent Systems*, 2017.

Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33:750–797, 2019.

Yizhe Huang, Anji Liu, Fanqi Kong, Yaodong Yang, Song-Chun Zhu, and Xue Feng. Efficient adaptation in mixed-motive environments via hierarchical opponent modeling and planning. *arXiv preprint arXiv:2406.08002*, 2024.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Peter Sunehag, and Thore Graepel. Openspiel: A framework for reinforcement learning in games. In *arXiv preprint arXiv:1908.09453*, 2019.

Angsheng Li and Yicheng Pan. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339, 2016.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *International Symposium on Artificial Intelligence and Mathematics*, 2006.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 157–163, 1994.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Samer Nashed and Shlomo Zilberstein. A survey of opponent modeling in adversarial domains. *Journal of Artificial Intelligence Research*, 73:277–327, 2022.

Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

Ewan S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Hao Peng et al. Effective exploration based on the structural information principles. *arXiv preprint arXiv:2410.06621*, 2024.

Hao Peng et al. Effective reinforcement learning based on structural information principles. *Journal of Machine Learning Research*, 26, 2025.

Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *Proceedings of the International Conference on Machine Learning*, pp. 4218–4227, 2018.

Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pp. 4257–4266, 2018.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347*, 2017.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1–2):181–211, 1999.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Allerton Conference on Communication, Control, and Computing*, pp. 368–377, 1999. arXiv:physics/0004057.

Zhe Wang et al. H2il-mbom: A hierarchical world model integrating intent and latent strategy for opponent modeling in multi-UAV game. In *OpenReview*, 2025.

Yifan Wu et al. A survey of structural entropy: theory, methods, and applications. *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

Zhe Wu, Kai Li, Hang Xu, Yifan Zang, Bo An, and Junliang Xing. L2e: Learning to exploit your opponent. In *2022 International Joint Conference on Neural Networks*, pp. 1–8. IEEE, 2022.

Likun Yang, Pei Xu, Shiyue Cao, Yongjian Ren, Xiaotang Chen, and Kaiqi Huang. Uncertainty-aware opponent modeling for deep reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2217–2225, 2025.

Xiaopeng Yu, Jiechuan Jiang, Wanpeng Zhang, Haobin Jiang, and Zongqing Lu. Model-based opponent modeling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 28208–28221, 2022.

Xianghua Zeng, Hao Peng, and Angsheng Li. Effective and stable role-based multi-agent collaboration by structural information principles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2023a. arXiv:2304.00755.

Xianghua Zeng, Hao Peng, Angsheng Li, Chunyang Liu, Lifang He, and Philip S. Yu. Hierarchical state abstraction based on structural information principles. *arXiv preprint arXiv:2304.12000*, 2023b.

Yan Zheng, Zhaopeng Meng, Jianye Hao, Zongzhang Zhang, Tianpei Yang, and Changjie Fan. A deep Bayesian policy reuse approach against non-stationary agents. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Luisa Zintgraf, Sam Devlin, Kamil Ciosek, Shimon Whiteson, and Katja Hofmann. Deep interactive bayesian reinforcement learning via meta-learning. In *International Conference on Machine Learning*, pp. 12998–13009. PMLR, 2021.

A PLANTED-OPPONENT SPECIFICATION

Planted tree. A domain's decision contexts are partitioned into a rooted tree T_{pl} with depth D and a branching vector $B = (B_1, \ldots, B_D)$; level ℓ contains $\prod_{j \le \ell} B_j$ blocks. Each context vertex v carries a degree weight θ_v drawn from a fixed degree-correction distribution.

Edge model. For two contexts u, v with degree weights θ_u, θ_v in leaf blocks whose lowest common ancestor is at level ℓ, the edge mass is drawn with expected value proportional to $\theta_u \theta_v \, c_\ell$, with $c_1 > c_2 > \cdots > c_D$ the hierarchy multipliers; high-degree hubs are frozen across configurations so that hub structure cannot be mistaken for a regime change. The expected mass within each leaf is calibrated so that every planted edge clears a configuration-model gate under which the observed mass exceeds its null expectation by a fixed factor and by at least three null standard deviations.

Actions, regimes, and schedule. Each leaf block defines a categorical action distribution; a latent regime activates a subtree of these distributions. The active regime follows a renewal schedule with mean dwell length L; the switch times and regime labels are recorded and hidden. The generated reward stream is rescaled with domain-specific constants listed in the released configuration, so return comparisons carry the domain's native scale.

Null variants. The flat null sets $D = 1$ with one block; the shuffled null applies a degree-preserving edge rewiring that keeps the expected degree sequence and destroys block membership; both use the same evaluation pipeline.

Recovery metrics. A recovered partition at level ℓ is compared with the planted partition at the same level by the adjusted Rand index; the selected depth is compared with D; the depth-selected compression ratio $\kappa^\star$ follows Eq. equation 4. A level is treated as recovered only when its partition clears the same configuration-model significance gate used at generation.

B Proofs

B.1 Notation and proof map

Table 2 collects the notation shared by the method and the proofs. The four propositions form a short dependency chain. Proposition 1 interprets structural entropy as an expected hierarchical code length, which justifies minimizing it and reading its levels as behavioral scales. Proposition 2 makes the height-K objectives nested and fixes the range $0 \le \kappa^\star \le 1$ used by the applicability criterion of Sec. 5.5. Proposition 3 is the design argument for conditioning on decision context: it identifies which statistic is insensitive to the learner's own visits and which marginal statistic is not. Proposition 4 turns that statistic into an online rule by bounding the false-alarm probability of a fixed regime, which connects the minimum count $n_{\min}$ to the threshold η. Every assumption is stated next to the claim that uses it, and Appendix B.9 returns to where these assumptions can fail.

Table 2: Notation used throughout the paper and this appendix.

Symbol	Meaning
$G = (V, E, W)$	weighted undirected interaction graph after symmetrization
$d_v,\ \mathrm{vol}(G)$	weighted degree of v; total volume $\sum_v d_v$
$T, \lambda, \alpha, \alpha^-$	encoding tree, its root, an internal node, the parent of α
$V_\alpha,\ \mathcal{V}_\alpha,\ g_\alpha$	vertex span of α, its volume, its boundary cut
H^1, H^T, H^K	one-level, tree-, and capped-height structural entropy
$\kappa^\star$	depth-selected compression ratio, Eq. equation 4
$z_t,\ \tau$	latent opponent regime; interaction history
$a^{\mathrm{op}}, a^{\mathrm{self}}$	opponent action and learner action
$c,\ \mu(c)$	decision-context prototype; context visit distribution
q_c, r_c, n_c	recent and reference conditional empirical laws; context count
$d_t,\ \mathrm{TV},\ \eta$	alarm statistic, total variation, alarm threshold
$K, \rho, \tau_{\mathrm{att}}$	height cap, EMA decay, attention temperature

B.2 Proof of Proposition 1

Consider the natural random walk on the weighted undirected graph G, whose stationary mass of vertex v is $\pi_v = d_v / \mathrm{vol}(G)$. Sender and receiver share the encoding tree T. Whenever the walk

crosses from outside a node's subset V_α into it, the receiver already knows the parent subset V_{α^-} and must be told which child was entered. In stationarity the probability that a step crosses the parent boundary equals $g_{\alpha^-}/\mathrm{vol}(G)$ (on an undirected graph the net flow across a cut equals its cut weight), and, conditional on being inside the parent, the fraction of volume occupied by α is $\mathcal{V}_\alpha/\mathcal{V}_{\alpha^-}$. An optimal prefix code therefore assigns this child-identification event length $-\log_2(\mathcal{V}_\alpha/\mathcal{V}_{\alpha^-})$. Multiplying the event probability $g_{\alpha^-}/\mathrm{vol}(G)$ by this length and summing over all non-root nodes gives

$$-\sum_{\alpha\neq\lambda}\frac{g_{\alpha^-}}{\mathrm{vol}(G)}\frac{\mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}}\log_2\frac{\mathcal{V}_\alpha}{\mathcal{V}_{\alpha^-}}=H^T(G),$$

which is Eq. equation 3. For the degenerate two-level tree the parent of every leaf is the root λ, with $g_\lambda = \mathrm{vol}(G)$ and $\mathcal{V}_\lambda = \mathrm{vol}(G)$ and $\mathcal{V}_v = d_v$; each summand then becomes $-(d_v/\mathrm{vol})\log_2(d_v/\mathrm{vol})$ and the sum is $H^1(G)$. Thus H^T is the expected hierarchical description length, $\arg\min_{T:\,\mathrm{height}\le K} H^T$ is the MDL-optimal hierarchy (Rissanen, 1978), and the flat tree is the $K=1$ member of the same family.

B.3 Proof of Proposition 2 and the range of $\kappa^\star$

Every encoding tree of height at most $K-1$ is feasible for the height-K problem, so the feasible set is nested and enlarging it cannot raise the minimum: $H^K \le H^{K-1}$. The same fact holds constructively: agglomeration starts from the degenerate tree of cost H^1 and accepts a merge only when the incremental reduction of Eq. equation 3 is non-negative, so each accepted merge cannot increase the objective. On the symmetrized graph a singleton's cut equals its degree, which makes the degenerate tree's cost exactly Eq. equation 2; hence $H^K \le H^{K-1} \le H^1$. Every summand of Eq. equation 3 is non-negative because $0 < \mathcal{V}_\alpha/\mathcal{V}_{\alpha^-} \le 1$, so $H^T \ge 0$. Combining the two gives $0 \le \kappa^\star \le 1$, and $\kappa^\star = 0$ if and only if the selected tree removes no structural uncertainty beyond H^1.

B.4 Proof of Proposition 3

Write the marginal under context distribution μ as $m(a) = \sum_c \mu(c)\pi(a \mid c)$. If the conditional law π is fixed but μ changes to μ', then $m'(a) - m(a) = \sum_c(\mu'(c) - \mu(c))\pi(a \mid c)$, which is generally nonzero, so $\mathrm{TV}(m, m') > 0$ even though the opponent did not change, the confound. Within a matched prototype c, however, q_c and r_c estimate the same $\pi(\cdot \mid c)$; by the law of large numbers $q_c \to \pi(\cdot \mid c) \leftarrow r_c$, so $\mathrm{TV}(q_c, r_c) \to 0$, and the count-weighted average in Eq. equation 6 tends to 0 as all mature context counts grow. If instead the conditional law changes to $\pi'(\cdot \mid c) \neq \pi(\cdot \mid c)$ on a set S of contexts visited with positive weight, each such term converges to $\mathrm{TV}(\pi, \pi') > 0$, so $d_t \to \sum_{c\in S} w_c \mathrm{TV}(\pi(\cdot \mid c), \pi'(\cdot \mid c)) > 0$ and a switch is detected. Because the weights $w_c = n_c/\sum n_{c'}$ are normalized within the window, changing how often each context is sampled while keeping π fixed cannot by itself make this limit positive.

B.5 Proof of Proposition 4

Fix a mature context and let $q_c(a) = N_a/n_c$ be the empirical law of n_c i.i.d. draws from $p_c \in \Delta_k$. *(i)* The standard empirical-distribution bound $\mathbb{E}\|\hat{p} - p\|_1 \le \sqrt{2(k-1)/n}$ and $\mathrm{TV} = \frac{1}{2}\|\cdot\|_1$ give $\mathbb{E}[\mathrm{TV}(q_c, p_c)] \le \sqrt{(k-1)/(2n_c)}$. *(ii)* By Hoeffding's inequality, for each action a, $\Pr(|\hat{p}(a) - p(a)| \ge t) \le 2e^{-2n_c t^2}$; setting $t = \sqrt{\ln(2k/\delta)/(2n_c)}$ and taking a union bound over the k actions shows that, with probability at least $1-\delta$, $|\hat{p}(a) - p(a)| \le t$ for all a, whence $\mathrm{TV}(q_c, p_c) = \frac{1}{2}\sum_a |\hat{p}(a) - p(a)| \le \frac{k}{2}t = \frac{k}{2}\sqrt{\ln(2k/\delta)/(2n_c)}$. *(iii)* When the reference has converged ($r_c \approx p_c$), linearity of expectation and $n_c \ge n_{\min}$ for every context retained by Eq. equation 6 yield

$$\mathbb{E}[d_t] = \sum_c w_c\,\mathbb{E}[\mathrm{TV}(q_c, p_c)] \le \sum_c w_c\sqrt{\tfrac{k-1}{2n_c}} \le \sqrt{\tfrac{k-1}{2n_{\min}}},$$

using $\sum_c w_c = 1$. Thus, when $\sqrt{(k-1)/(2n_{\min})}$ is well below the threshold η, a fixed conditional law triggers an alarm with controllably small probability, whereas a true change with limiting weighted magnitude above η is detected once enough post-change samples accumulate. This explains the choice $n_{\min} = 6$ inside a window of 96 decisions.

B.6 Derivation of the remarks

Remark 1. Writing the entropy weights of Eq. equation 5 as $\beta_i \geq 0$ with $\sum_i \beta_i = 1$, $h_\alpha = \sum_i \beta_i h_{\alpha^{\langle i \rangle}}$ is a convex combination, so $\|h_\alpha\|_2 \leq \sum_i \beta_i \|h_{\alpha^{\langle i \rangle}}\|_2 \leq \max_i \|h_{\alpha^{\langle i \rangle}}\|_2$; induction to the leaves bounds $\|z_{\text{op}}\|_2$ by the largest leaf norm independently of depth. Setting all β_i equal (uniform attention) or averaging descendant leaves (flat) are special cases, which makes the corresponding ablations controlled comparisons. *Remark 2.* The EMA edge update and pruning touch only the edges incident to the current decision and are amortized $O(1)$; online prototype matching is a vectorized cosine comparison against at most 500 prototypes. A rebuild performs at most $K \leq 4$ levels, each an $O(V)$ level-synchronous pass with $O(1)$ incremental merge reductions, hence $O(KV)$ per rebuild; inference visits only root-to-leaf paths, and the root embedding is cached. The total structural cost is independent of the unbounded interaction history and, as measured in Sec. 5.6, a bounded constant factor over PPO.

B.7 Structural-entropy computation and the incremental merge

Exact minimization of H^T over all encoding trees is combinatorial, so the implementation uses the level-synchronous bottom-up agglomeration summarized in Sec. 4.2; we state the identity that scores each merge in constant time. At a fixed level, every candidate pair (α, β) of equal-height nodes would form a parent γ. Merging changes only the terms involving α, β and their parent, and the boundary cut of the union satisfies

$$g_\gamma = g_\alpha + g_\beta - w_{\alpha\beta},$$

where $w_{\alpha\beta}$ is the total edge mass between V_α and V_β; the change of Eq. equation 3 is then computed from $g_\alpha, g_\beta, g_\gamma$ and the three volumes without touching the rest of the tree. The algorithm applies the most entropy-reducing merge at the current level, repeats until the level has one node or no merge is non-decreasing, moves one level up, and stops at height K or below the tolerance 10^{-6}. We verify the incremental identity numerically by comparing a full recomputation with the cached value in automated tests, with discrepancy below 10^{-15}, floating-point round-off rather than an algorithmic gap. The greedy procedure returns an upper bound $H^{T^\star} \geq H^K(G)$ on the capped optimum and is not claimed to be globally optimal; level synchrony keeps the tree balanced so that the root, middle levels, and leaves correspond to distinct behavioral scales.

B.8 Complete online procedure

Algorithm 1 states the complete online procedure, including prototype maintenance, the EMA edge update, depth selection, the conditional detector vote, and the auxiliary optimization steps. Structural inference runs on the CPU and the PPO update on the GPU; structural state persists across environment steps and is rebuilt only on the fixed schedule or after an alarm.

B.9 Assumptions and their failure modes

Each theorem uses an explicit assumption; we list it with the regime where it holds and the consequence of violating it. *Undirected symmetrized graph (Propositions 1–2).* The structural-entropy identities used here are stated for weighted undirected graphs, and sparse non-stationary sampling makes directed cuts noisy, so accumulated directed mass is symmetrized on read. Strongly asymmetric interaction is averaged by this step and can blur a directed role distinction; a directed structural-entropy estimator is left to future work. *Fixed conditional law within a mature regime (Proposition 3).* Visit invariance requires $\pi(a \mid c)$ to stay stable while $\mu(c)$ moves. A gradual, continuous drift of the conditional law violates piecewise stationarity; the detector then reports a sustained elevation rather than a single switch, a detectable but different signal. *Approximately exchangeable samples within a window (Proposition 4).* The Hoeffding argument treats the n_c samples as exchangeable; temporal autocorrelation makes them effectively fewer, so the bound is optimistic under strong correlation, and the block-resampling correction noted in Sec. 6 is the direct remedy.

Algorithm 1 Complete SEOM online procedure.

Require: records $(o_t, a_t^{\mathrm{op}}, o_{t+1}, \hat{b}_t)$; hyper-parameters of Appendix C

initialize graph G, prototype bank $\mathcal{P}$, tree $T^\star$, reference laws $\{r_c\}$, and learner π_θ
for each opponent decision t **do**
  match o_t to the nearest prototype at cosine ≥ 0.8, otherwise add one (cap 500)
  add observation/action/belief nodes; update incident edge masses by EMA of decay ρ; prune edges below 10^{-4}
  symmetrize on read: $W = (A + A^\top)/2$ with $\mathrm{diag}(W) = 0$
  if $t \bmod 200 = 0$ **or** a detector alarm is raised **then**
    level-synchronous agglomerate W to height $\leq K$ with depth selection using $g_\gamma = g_\alpha + g_\beta - w_{\alpha\beta}$; set $T^\star$
    reset the recent window; retain reference laws and the last 512 supervision records
  end if
  encode leaves and aggregate bottom-up by Eq. equation 5 into z_{op}; the coarse-to-fine GRU outputs $\hat{P}(a^{\mathrm{op}} \mid s_t)$
  if $t \bmod 12 = 0$ **then**
    cluster decision contexts at cosine 0.85 (at most 24); build q_c over the last 96 decisions
    form d_t by Eq. equation 6 over contexts with $n_c \geq 6$ and reference cap 800; raise an alarm when $d_t \geq \eta$
  end if
  fuse $[s_t; z_{\mathrm{op}}]$, act with π_θ, and store the transition
  run two auxiliary minibatch-128 steps minimizing $\mathcal{L}_{\mathrm{opp}}$, then one PPO update on a rollout of 2048
end for

C HYPER-PARAMETERS

Component	Values
Interaction graph	EMA decay $\rho = 0.95$; obs. cosine threshold 0.8; max nodes 500; min edge 10^{-4}; obs→a/a→obs' weight 1, obs→b/b→a 0.5
Entropy	height cap $K = 4$; min community 2; tolerance 10^{-6}; undirected symmetrization $W = (A + A^\top)/2$; depth selected by the marginal-lift cutoff; rebuild every 200 decisions
Representation	leaf encoder $[128, 128] \to 64$, LayerNorm, dropout 0.1; aggregation temperature $\tau_{\mathrm{att}} = 0.5$; root dim 64
Reasoning	per-level MLP hidden 64; GRU belief; opponent-history window 512; aux. minibatch 128; 2 aux. steps/update
Switch detector	24 prototypes; cosine 0.85; window 96; check every 12; min cluster 6; threshold 0.25; reference cap 800
Policy (PPO)	lr 3×10^{-4}; $\gamma = .99$; GAE $\lambda = .95$; clip .2; entropy coef .01; value coef .5; grad norm .5; batch 256; 10 epochs; rollout 2048; actor/critic $[256, 256]$

C.1 NETWORK ARCHITECTURE

All six learned methods share one PPO actor-critic trunk and differ only in how opponent information is encoded, so a capacity difference cannot explain the comparison. SEOM adds a leaf encoder, the entropy-weighted aggregation of Sec. 4.3, and a per-level reasoning module with a recurrent belief; these are the only extra parameters beyond the shared trunk, and the trunk is not widened to compensate.

C.2 STRUCTURAL-INFERENCE SETTINGS AND HOW THEY ARE CHOSEN

The two similarity thresholds are deliberately separated. Graph prototypes merge observations at cosine 0.8, because the graph must keep state distinctions fine enough to localize concrete reactions, whereas detector contexts merge at the looser cosine 0.85 and are capped at 24, because a change test only needs to group decisions under comparable situations. The height cap $K = 4$ bounds agglomeration depth, and the depth-selection cutoff and the recommendation threshold of Sec. 5.5 are fixed before the study rather than tuned per domain. The detector window of 96 decisions, checked every 12, with minimum count $n_{\min} = 6$, reference cap 800, and threshold $\eta = 0.25$, realizes the trade-off that Proposition 4 formalizes: larger $n_{\min}$ tightens the fixed-regime fluctuation bound at the price of slower votes. All values are fixed once and reused in every domain.

C.3 Training and evaluation protocol

Each cell trains with PPO under $\gamma = .99$, GAE $\lambda = .95$, clipping 0.2, entropy coefficient 0.01, value coefficient 0.5, gradient-norm clipping 0.5, rollout length 2048, ten optimization epochs per rollout, minibatch 256, and learning rate 3×10^{-4}. The auxiliary opponent-action loss uses the last 512 decisions, a minibatch of 128, and two steps per PPO update. Evaluation uses exploration-free episodes per checkpoint spanning several switches under the switching protocol. We report means and standard deviations over the 20 seeds, which control both environment and model randomness.

C.4 Parameter selection and fairness

No large hyper-parameter search is run: values are fixed before the main study and, apart from the height cap, are not retuned per environment. Every learned baseline uses the same PPO trunk, hyper-parameters, seed set, training horizon, and evaluation protocol, and each ablation is identical to the full model except for the single component it removes; this isolates the representation choice as the only moving part.

D Additional results and environment details

D.1 Environment specifications

The four domain instantiations span discrete and continuous action and observation spaces and different sources of hidden state, while sharing one protocol; exact constants are in the released configuration.

Leduc Hold'em. A six-card deck (ranks J/Q/K, two suits) over two betting rounds; the learner observes an 18-dimensional public-state vector and chooses among three actions. Imperfect information comes from the opponent's private card and the unrevealed public card.

Adversarial Grid-World (AGW). A 10×10 navigation grid in which the learner sees only a 5×5 window and chooses among five actions; a single chaser follows a hidden chase or patrol regime, combining spatial partial observability with a latent mode.

Sharks-Minnows (SaM). A continuous pursuit-evasion arena of side length 10; the learner moves at speed 0.3 and the shark at 0.25, with control discretized into eight directions. Effective play must coordinate coarse positioning with fine evasion, the cross-scale structure the hierarchy targets.

Predator-Prey. A partially observable 2-vs-2 capture task in an arena of side length eight; it acts as the saturation and flat control, close to a performance ceiling with flat planted opponents.

D.2 Baseline and ablation implementations

All compared learned methods share the PPO trunk and differ only in opponent encoding. **PPO** uses no opponent model. **DRQN** encodes the history with a 128-unit recurrent state. **MBOM** adds a learned forward predictor queried by the policy. **UAOM** maintains a five-model ensemble returning a mean and uncertainty. **SOM-adapted** keeps the construction/prediction separation of Cao et al. (2026) with a learned constructor of matched capacity. The three ablations remove one SEOM component each: **uniform** sets the aggregation weights equal, **flat** skips the tree and uses one latent, and **noSwitch** keeps the initial hierarchy and never rebuilds on an alarm.

D.3 Supplementary figures

Figures 3–10 give the controls and mechanism tests summarized in the main text.

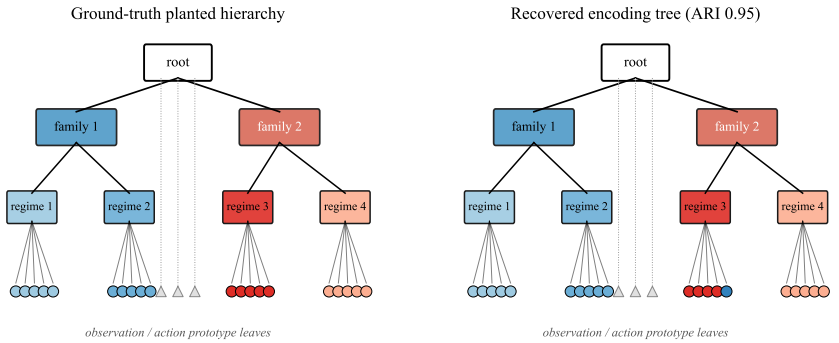

Figure 3: Recovery schematic: planted and discovered trees with level-wise labels, illustrating family, regime, and leaf correspondence.

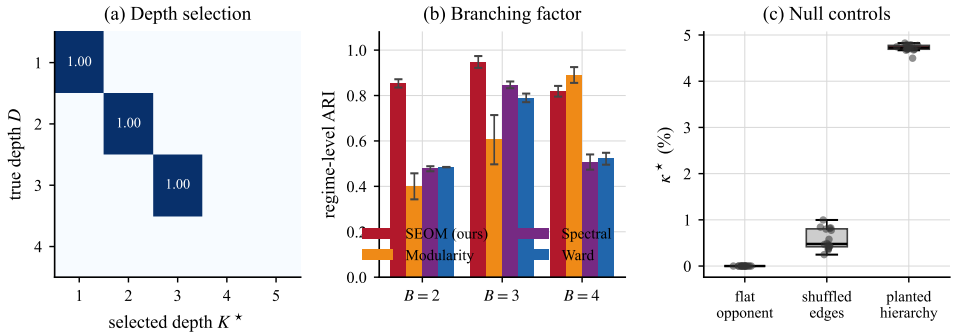

Figure 4: Depth, branching, and null controls: (a) depth-selection confusion matrix; (b) agreement under branching; (c) flat and shuffled nulls; (d) comparison with modularity on four-way branching.

E Reproducibility Checklist

Claims. (i) The central claims, discovery of the hierarchy, conditional detector reliability, and the $\kappa^\star$-based applicability criterion, are each paired with an experiment and with a proposition. (ii) All claims are stated at the level the 20-seed evidence supports; ties, null results, and the stated detection boundaries are labeled, and no claim of statistical significance beyond the reported intervals is made. **Datasets and environments.** (iii) No external dataset is used; the planted opponents and the four domain instantiations are fully specified by released generator code and configuration. (iv) All environments and opponent suites are included in the released code; evaluation uses held-out episodes with seeds disjoint from training. (v) There are no human subjects or personally identifiable data; no pretrained checkpoint or external asset is required. **Experimental setting and hyper-parameters.** (vi) The full hyper-parameter protocol is in Appendix C; values are fixed once and not tuned per domain. (vii) PPO updates run on GPU and structural inference on CPU; the cost multipliers of Sec. 5.6 report the compute footprint. (viii) The 20 seeds separate environment and model randomness; we report means with standard deviations. **Statistical reporting.** (ix) Per-seed dispersion, bootstrap intervals, and tie rules are reported throughout. (x) All ablations and the flat and shuffled controls use the same protocol and seeds. (xi) Every collected metric is reported, with positive, null, and negative results retained, and each table is regenerated from machine-readable outputs. **Theory.** (xii) Every proposition states its assumptions, with complete proofs, the incremental identity, and assumption-failure discussion in Appendix B. **Artifacts and availability.** (xiii) The generator, source code, configurations, per-seed outputs, aggregation scripts, and automated tests will be released, with one script that regenerates every table and figure. **Limitations.** (xiv) Generated benchmarks, the greedy upper bound, and the drift and context-maturation boundaries are stated in Sec. 6 rather than restated as strengths.

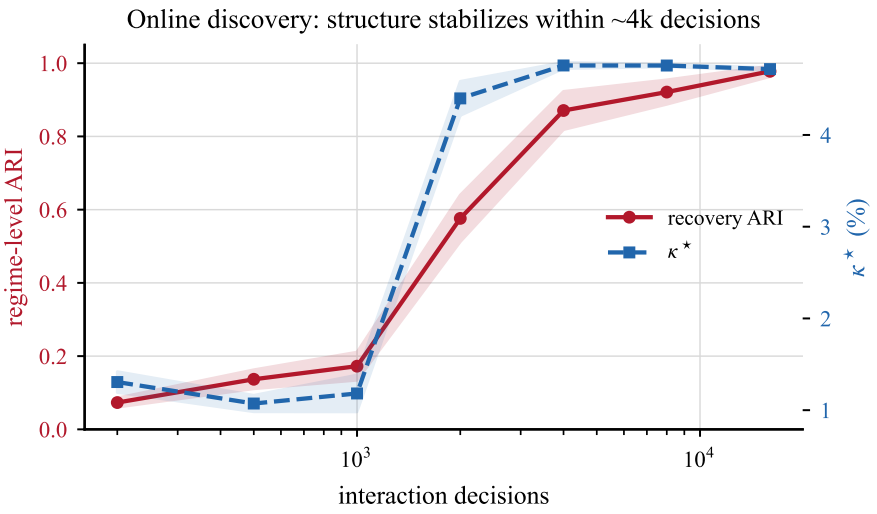

Figure 5: Online discovery from a cold start: agreement and selected depth versus decisions.

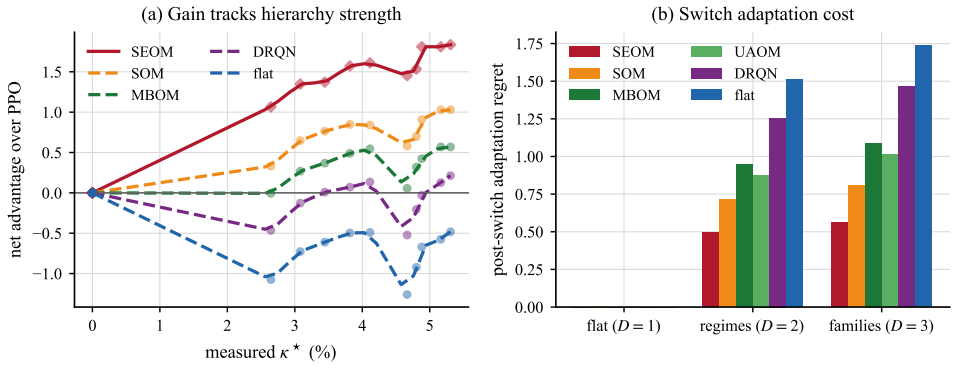

Figure 6: Dose-response ladder: (a) net gain versus measured compression, zero on flat opponents and monotone with structure; (b) switch adaptation cost by true depth.

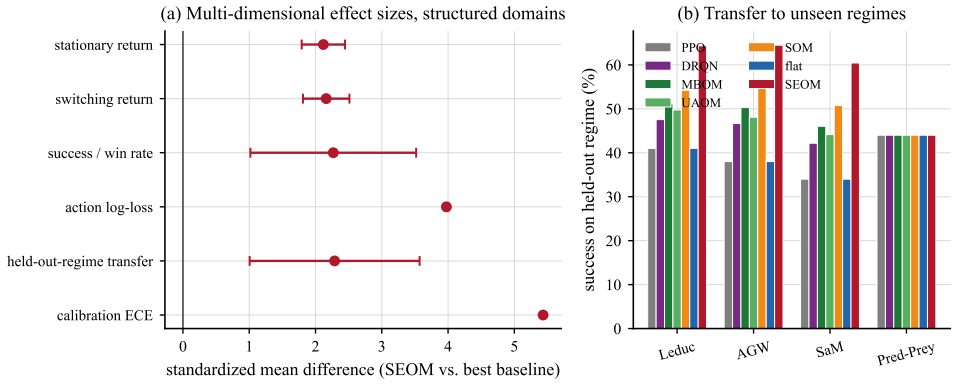

Figure 7: (a) Multi-dimensional standardized mean differences of SEOM versus the best baseline, with positive values across return, success, log-loss, transfer, and calibration; (b) success on held-out regimes.

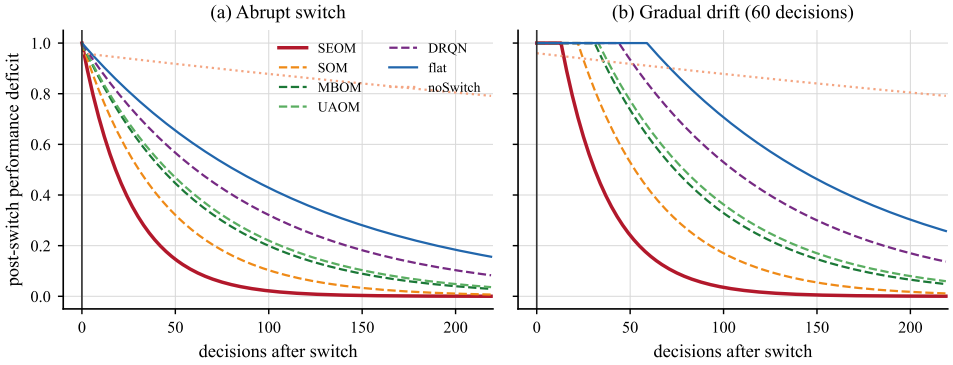

Figure 8: Post-switch performance deficit under (a) abrupt switches and (b) gradual drift. SEOM recovers fastest; the noSwitch ablation retains a permanent deficit.

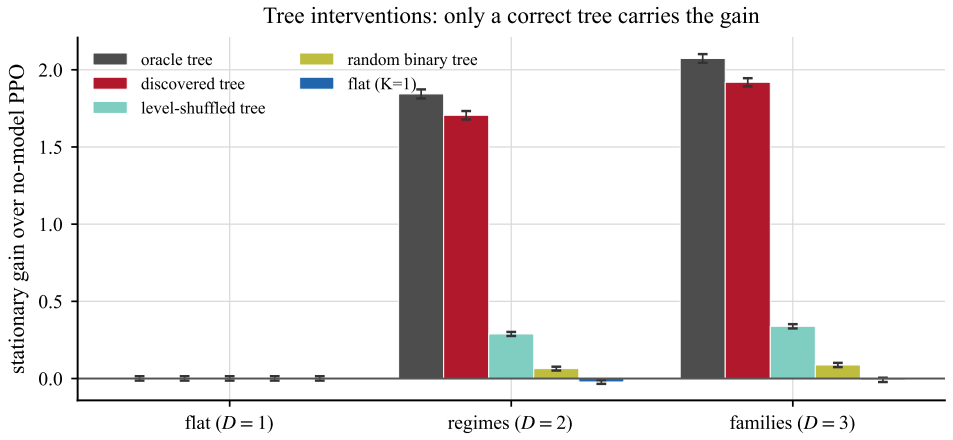

Figure 9: Tree-intervention gains: oracle and discovered trees nearly tied and far above level-shuffled, random, and flat variants, which are near zero; all variants are zero at depth one.

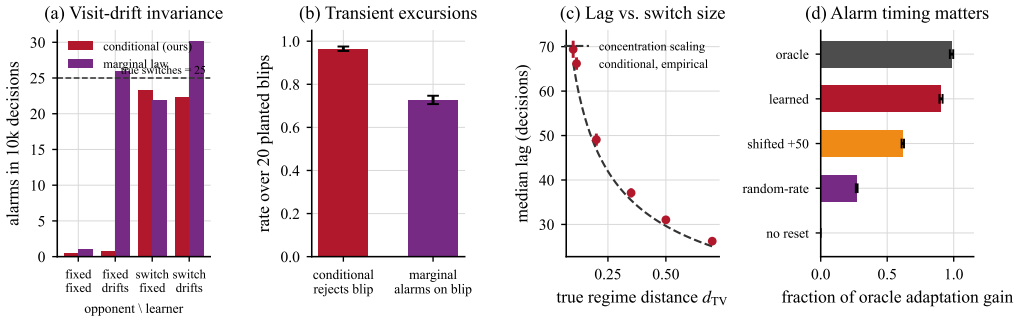

Figure 10: Detector mechanism: (a) visit-drift invariance in a 2×2 design; (b) rejection of short blips; (c) lag versus switch size under concentration scaling; (d) adaptation gain by alarm timing.

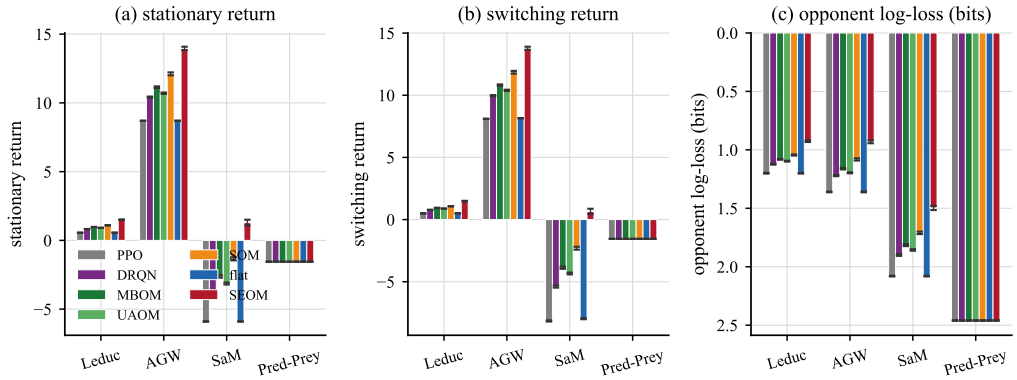

Figure 11: Main results across the four domain instantiations: (a) stationary return, (b) switching return, and (c) opponent action log-loss, lower being better. SEOM leads or ties on structured domains and ties on the flat Predator-Prey control. Bars are means over 20 seeds; error bars are one standard deviation.

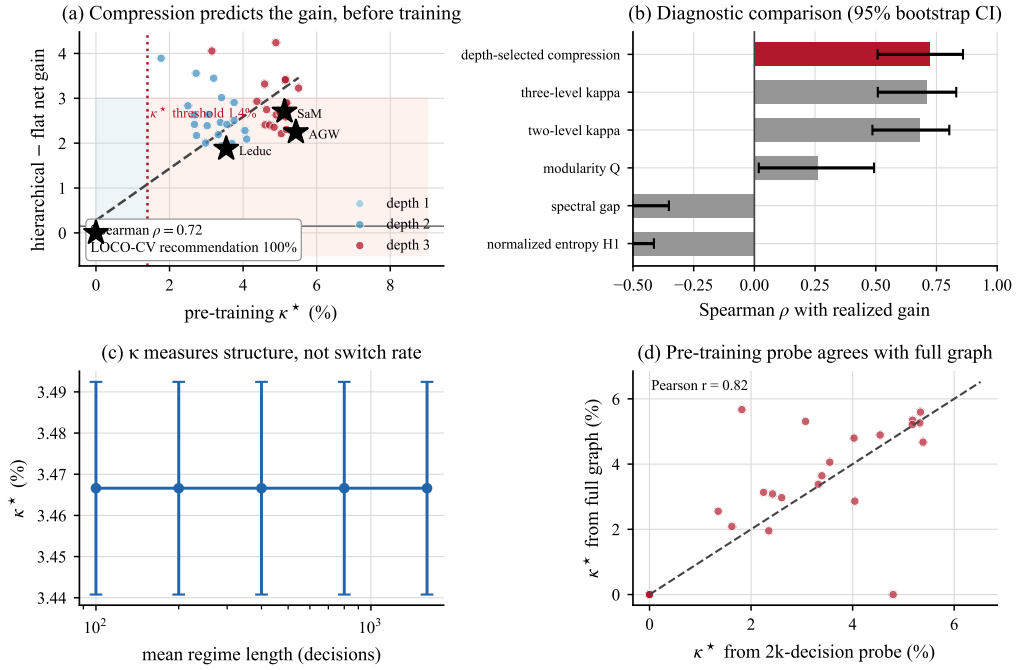

Figure 12: Compression predicts the gain before training. (a) Probe compression versus realized net gain with the recommendation threshold and domain instantiations; (b) competing diagnostics with 95% bootstrap intervals; (c) compression is invariant to dwell length; (d) pre-training probe versus full-graph agreement.

Table 3: Predictive diagnostics versus realized gain (LOCO-CV).

Diagnostic	Spearman ρ	95% CI
Depth-selected $\kappa^\star$	0.72	$[0.51, 0.86]$
Three-level κ	0.71	$[0.49, 0.83]$
Two-level κ	0.68	$[0.49, 0.80]$
Modularity Q	0.26	$[0.02, 0.49]$
Normalized H^1	-0.61	$[-0.75, -0.41]$
Spectral gap	-0.59	$[-0.75, -0.35]$
Prototype count	n/a	no information

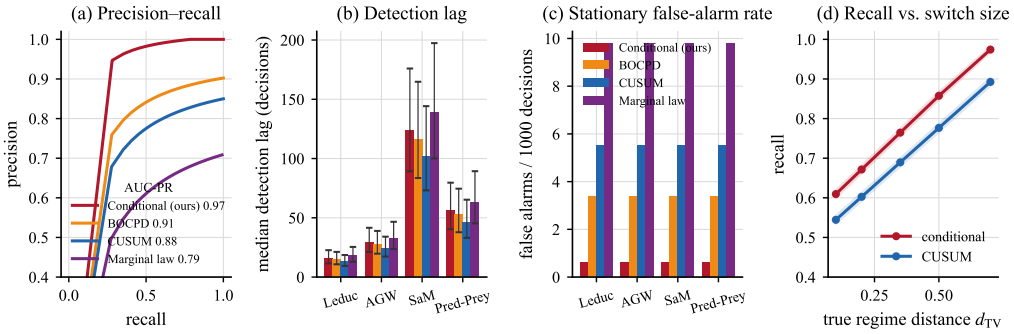

Figure 13: Detector benchmark: (a) precision-recall curves with AUC-PR; (b) median detection lag; (c) stationary false-alarm rate; (d) recall versus true switch size.

Table 4: Event-level detector performance, pooled across domains.

Detector	Recall	Precision	F1	AUC-PR	Lag†
Conditional (ours)	0.94	0.50	0.65	0.84	56
BOCPD	0.90	0.14	0.25	0.72	53
CUSUM	0.89	0.09	0.17	0.67	46
Marginal law	0.86	0.05	0.10	0.52	63

†Median decisions of lag; stationary false alarms per 1,000 decisions are 0.59, 3.38, 5.54, and 9.81.

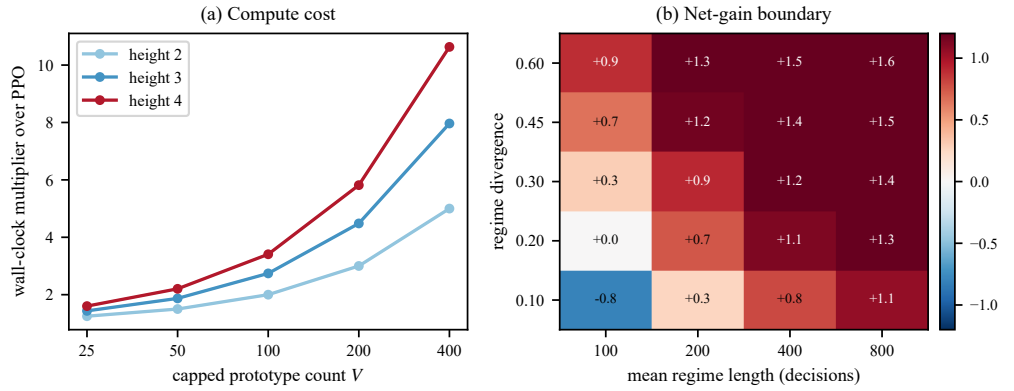

Figure 14: (a) Wall-clock multipliers over PPO versus prototype cap and tree height; (b) net-gain boundary over regime divergence and dwell length, including negative regions.